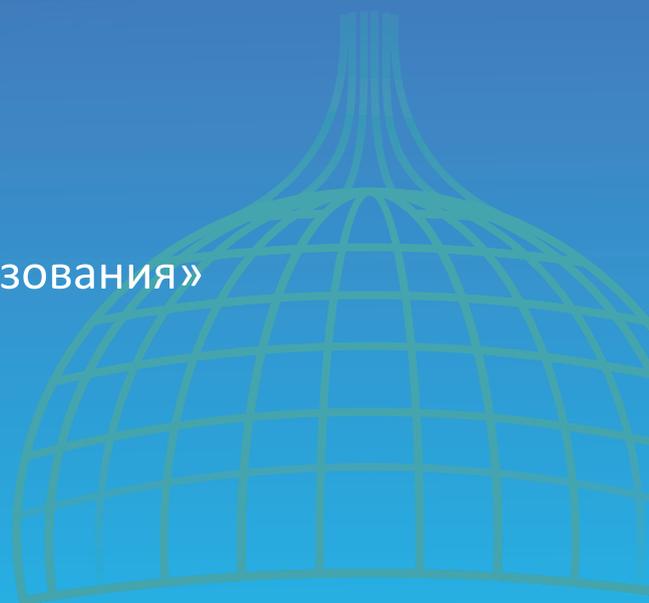


PSI 19

12-я Международная Ершовская
конференция по информатике
Труды семинара «Информатика образования»

2–3 июля 2019
Россия, Новосибирск



ИНСТИТУТ СИСТЕМ ИНФОРМАТИКИ ИМ. А. П. ЕРШОВА СО РАН
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПСИ'19

**12-я МЕЖДУНАРОДНАЯ ЕРШОВСКАЯ КОНФЕРЕНЦИЯ
ПО ИНФОРМАТИКЕ**

ИНФОРМАТИКА ОБРАЗОВАНИЯ

Труды семинара

2-3 июля 2019 г.

Новосибирск
2019

УДК 004.4
ББК В185я431
И 741

Программный комитет:

Председатель — д-р физ.-мат. наук, проф. *А. Г. Марчук*
Секретарь программного комитета — *Т. И. Тихонова*

д-р физ.-мат. наук, д-р пед. наук, проф. *А. Г. Гейн*, канд. физ.-мат. наук *Л. В. Городняя*,
канд. техн. наук, проф. *В. П. Котляров*, д-р физ.-мат. наук, проф. *М. М. Лаврентьев*,
канд. физ.-мат. наук *Ф. А. Мурзин*, д-р физ.-мат. наук, проф. *А. Н. Терехов*,
В. Г. Перкова, *А. В. Гиглавый*

И 741 Информатика образования : труды семинара 12-й Междунар. Ершовской конф. по информатике (ПСИ'19). 2–3 июля 2019 г. / Ин-т систем информатики им. А. П. Ершова СО РАН, Новосиб. гос. ун-т. — Новосибирск : ИПЦ НГУ, 2019. — 100 с.

ISBN 978-5-4437-0917-8

В сборнике содержатся материалы, представленные на рабочем семинаре «Информатика образования», который проходил в рамках 12-й Международной Ершовской конференции по информатике (ПСИ'19).

Семинар организован Институтом систем информатики им. А. П. Ершова СО РАН при поддержке Новосибирского государственного университета.

УДК 004.4
ББК В185я431

Печатается при поддержке гранта РФФИ № 19-07-20041.

ISBN 978-5-4437-0917-8

© Институт систем информатики
им. А. П. Ершова СО РАН, 2019
© Новосибирский государственный
университет, 2019

A. P. ERSHOV INSTITUTE OF INFORMATICS SYSTEMS SB RAS
MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
NOVOSIBIRSK STATE UNIVERSITY

PSI'19

**A. P. ERSHOV INFORMATICS CONFERENCE
(THE PSI CONFERENCE SERIES, 12TH EDITION)**

EDUCATIONAL INFORMATICS

Workshop proceedings

July 2-3, 2019

Novosibirsk
2019

УДК 004.4
ББК В185я431
И 741

Program Committee:
Chair — *Alexander Marchuk*
Secretary — *Tatiana Tikhonova*

Alexander Gein
Lidiya Gorodnyaya
Vsevolod Kotlyarov
Mikhail Lavrentiev
Feodor Murzin
Vera Perkova
Andrey Terekhov
Alexander Giglavy

И 741 Educational Informatics: Workshop proceedings of the 12 th A. P. Ershov Informatics Conference (PSI'19). July 2–3, 2019 / A. P. Ershov Institute of Informatics Systems SB RAS, Novosibirsk State University. — Novosibirsk : IPC NSU, 2019. — 100 p.

ISBN 978-5-4437-0917-8

This volume comprises the papers selected for presentation at the Workshop on Educational Informatics of the A.P. Ershov Informatics Conference PSI'19 (the PSI Conference Series, 12th edition)

The workshop is organized by the A. P. Ershov Institute of Informatics Systems SB RAS and Novosibirsk State University.

УДК 004.4
ББК В185я431

The reported study was funded by RFBR according to the research project № 19-07-20041.

ISBN 978-5-4437-0917-8

© A. P. Ershov Institute of Informatics Systems
SB RAS, 2019
© Novosibirsk State University, 2019

СОДЕРЖАНИЕ

Andreyeva T.A. Automation of Correctness Checking in Education	6
Воронцова О. В., Роженко Т. И. Использование электронных таблиц при решении задач на проценты, сплавы и смеси при подготовке к ОГЭ в рамках интегрированного курса по математике и информатике	16
Городняя Л. В., Демидов С. Е., Шаченко Н. В. Конструирование учебных игр	21
Григоренко М. В. Пропуск в будущее	28
Ильин В. П. О технологиях познания в интегрированных вычислительных окружениях математического моделирования	33
Иртегов Д. В., Нестеренко Т. В., Чурина Т. Г. Системы автоматизированной оценки заданий по программированию: использование в учебном процессе	35
Крайнева И. А., Ревич Ю. А. Образовательное значение истории информатики: состояние и перспективы	40
Лаврентьев М. М., [Васючкова Т. С.], Городняя Л. В., Держо М. А., Иванчева Н. А., Белаго И. В., Бартош В. С. Развитие профессиональной ориентации студентов IT-специальностей	47
Мартынова А. В., Дудышева Е. В. Организация критериального оценивания деятельности школьников на уроках информатики	51
Новожилова В. И. Использование компетентностно-ориентированных заданий разной сложности для обучения алгоритмизации и программированию в классах с разным уровнем готовности школьников к изучению информатики	58
Мелешко К. В., Перкова В. Г., Сапрыкин Э. Э., Сапрыкина Г. А. Электронные ресурсы для развития одаренности школьников	65
Скопин И. Н. Обучение параллельным вычислениям на ранней стадии изучения программирования	73
Тихонова Т. И. Традиции работы с обучающимися IT-технологиям	79
Тихонова Т. И. Научно-практическая конференция школьников как мотивация исследовательской деятельности	85
Файфель Б. Л. Особенности преподавания основ вычислений с плавающей точкой	92
Лаврищева Е. М. Отечественная теория программирования. Прошлое и настоящее	99

AUTOMATION OF CORRECTNESS CHECKING IN EDUCATION

T. A. Andreyeva

Ph.D., A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: ata@iis.nsk.su

Abstract. Having their origin in programming contests, now various systems for automated checking of solutions become useful not only in programming education. With their help, any teacher can organize a mini-contest or check own tests and students' works faster and easier. In order to attract a wider range of users, all processes of both checking and preparation should be automated.

The author extends the approach gained from own experience of using and creating the automated checking systems for the programming problems onto the automated checking of the non-programming problems.

The article discusses the structure of a common problem body, studies various approaches to the automated checking, introduces problem complexes, suggests methods for creating accurate and consistent problem statements and check sets, and touches the automation of the preparation and checking processes.

Keywords: Automated Correctness Checking, Test Case Generation, Complex Information Systems

Introduction

Systems for the automated checking of solutions originate from programming contests. But today they become the means of automation of the teacher's work. They can be useful not only at programming classes. With their help, any teacher can organize a mini-contest or check own tests and students' works faster and easier.

But systems for automated checking also make additional demands to the contest problems. This requires a higher discipline from authors of all parts of a contest problem. To eliminate difficulties, the processes of preparation to the automated check-up should also be automated. The special systems for the automated preparation of problem complexes can reduce the number of possible errors, ambiguities, and inconsistencies, especially if the problem's description, specifications, the check set, and the exemplar solution are created collegially.

To create full, explicit and consistent problems (not only for contests with the automated checking of solutions but for all quizzes as well) it is necessary to realize that a problem is a union of a statement, specifications and checking means. Thus we introduce the important notion of the *problem complex*.

To understand the structure of a proper system for the automated preparation of problem complexes, we thoroughly consider what a generalised problem, its solution and results are, how a gen-

eralised problem can be automatically checked, and how the number of check cases and the check approach depend on the amount and the type of the problem's variable and constant data, its conditions and restrictions.

Also, we touch some questions of automated generation of check cases.

1. Problem, Solution and Result

The common-sense point of view that *to solve a problem* means *to find a correct answer* is obviously unacceptable since a wrong method of solution can lead to a correct answer. Moreover, the method can be the main point of the solution not only in programming but in other fields too. For example, in IQ tests [8], an often task is “find out the rule and then apply it”.

Now, let a problem be represented by a triad $\langle \mathbf{D}, \mathbf{C}, \mathbf{R} \rangle$, where \mathbf{D} is known *data*, \mathbf{C} is *conditions* necessary for creating a correct solution and obtaining a correct *result*, and \mathbf{R} is unknown values to be found out during solving. Then the solving *method* \mathbf{M} is a function, which creates the result \mathbf{R} from the initial data \mathbf{D} and the conditions \mathbf{C} .

$$M: D \square C \rightarrow R$$

Solution method is a sequence of inter-connected and inter-dependable components ($\mathbf{R}_1, \dots, \mathbf{R}_N$); each *stage* \mathbf{R}_i is derived from stages $\mathbf{R}_{\{0..i-1\}}$ by a *step* of method \mathbf{M}_i . The whole method \mathbf{M} is thus a sequence of solution steps ($\mathbf{M}_1, \dots, \mathbf{M}_N$).

The sequence ($\mathbf{R}_1, \dots, \mathbf{R}_N$) is not bound to be strictly linear, on the contrary, in most cases it only can be depicted by a graph structure where vertices are *stages* $\mathbf{R}_i (i = 0 .. N)$ and oriented edges are steps $\mathbf{M}_i (i = 1 .. S)$.

Each shift \mathbf{M}_i from \mathbf{R}_i to \mathbf{R}_{i+1} can be converted into a subproblem and, thus, we can split the initial problem into several interconnected subproblems. The use of this point of view was studied in [4].

2. Correctness and Checking

Now that we agree that *to solve a problem* is *to find a correct solution method*, let us focus at the check-up of this method.

2.1. Equivalent methods

A problem may have several correct solution methods. For example, in order to find the greatest common divisor (G. C. D.) of two natural numbers, one can a) use the Euclidean algorithm or b) find all natural divisors for both numbers separately (by trying to divide them by each natural), compare these sets of divisors, and find the greatest common one. Both of these methods correctly solve the initial problem but differ in the efficiency.¹

Let us call two methods \mathbf{M}' and \mathbf{M}'' *equivalent* if being applied to the same input data \mathbf{D} they produce the same result \mathbf{R} .

¹ If it is important that the solution method be a particular one, the author of the problem can shift the focus from the result to the method itself: not “Find the G.C.D. of two naturals” but “Describe the Euclidean algorithm of finding the G.C.D.”. In this new problem, what earlier was a method (one of several possible ones) became the only correct result.

2.2. What is a correct method?

The most reliable way to check whether a method is correct is to prove that it solves the initial problem. Still, this way is too laborious to be exploited for mass checking. In education, we are in the situation of multiple methods to be checked in a short time and, thus, our aim is to diminish teacher's time and efforts. And the best way leading to automation of the checking process is to change the notion *correct*.

Suggesting students a problem in a test, teacher always has at least one "correct" solution method of this problem. This method we call *exemplar* and believe that it is actually correct. Here we should mention that proving of the correctness of the exemplar method left to teacher or contest organiser may cause difficulties; and we have multiple examples when a wrong or incomplete exemplar solution was declared correct and the really correct ones were erroneously declined.

Nonetheless, henceforth we call method M *correct* if it is equivalent to the exemplar method M_{ex} . Thus we shift from checking of the correctness to checking of the equivalence.

To make the matter more intricate, there is a situation with multiple correct answers (refer to Section 3.1). In this case, the exemplar method should provide all possible correct results while the method under examination may produce only one of them. So, the equivalence should be not between two methods M and M_{ex} but between the method M and only a sub-method of the method M_{ex} .

A good way to eliminate teacher's errors can be proposed by a profoundly developed information system intended for automating the preparation of problem cases, which will be discussed later in Section 4.

2.3. Checking

Let us discuss the equivalence check-up suitable for the educational purposes.

Since equivalent methods must produce the same outcomes (see sections 2.1 and 2.2), it is sufficient to check whether the result produced by the method under examination coincides with the *exemplar outcome* which is the result produced by the exemplar method.

This approach ascends to the theory of so called *Black-box testing* introduced by Ashby in 1956 [6] and well developed for programming (see, for example, [6, 7 or 13]); we shall try to adopt some of its methods for developing the theory of the automated checking in education.

3. Practicable Input and Output

The exemplar outcome is the result of applying the exemplar method to some exemplar initial data. Now let us discuss the number of these.

3.1. Power of the Output

The number of all possible correct answers $|R|$ is very important for our discussion of automation of checking. Here we only mention the possible variants. And the influence of the multiplicity of possible correct answers onto the checking process is discussed in Section 3.4.

$|R| = 0$ means that the problem is stated erroneously. No correct answer is possible. Such problems must not appear in any test, quiz or contest.

$|R| = I$ means that there exists the only solution. In this case, checking is obvious and easy: it is sufficient to ascertain that the retrieved outcome coincides with the given exemplar one.

$|R| > I$ means that the problem has several correct answers. There can be three cases:

- $|R|$ is finite. For example, “The anterior part of a shoulder is called a collar bone or a clavicle”;
- $|R|$ is infinite but denumerable. For example, “Any odd integer”;
- $|R|$ is infinite and non-denumerable. For example, “Any real value from the interval $[0..I]$ ” or “Any point on the plane within the circle with the centre in $(0, 0)$ and radius I ”.

Let us note that restrictions of the computer data representation obviously reduce the case of infinite (denumerable or non-denumerable) $|R|$ to the case of finite $|R| > I$. The only difference between them is in approaches to automated checking.

3.2. Power and Dimension of the Input

Variables and constants. Domain D of all known data can be divided into two parts: invariables D_{inv} and variables D_{var} . *Invariable data* are stated in the problem’s description and never change. On the contrary, *variable data* are provided during checking and can differ not only in value but in number too.

So, the power $|D_{var}|$ can vary, and the power $|D_{inv}|$ is constant. And the power of the whole domain D is their sum:

$$|D| = |D_{var}| + |D_{inv}|.$$

No Variables. Obviously, there exist problems with no variable input. Almost all non-programming problems are like this. In this case, the *power* $|D_{var}| = 0$.

If such a problem is correctly stated, it must have the unique correct answer. As a rule, if there are several correct answers, it means that the problem description is not full or consistent.²

Variables as the Means to Split a Problem. In programming, any “good” algorithm has to be *mass-oriented*, i.e., it must be potent to solve not a single problem but a whole class of similar problems. Therefore, presence of a variable input is characteristic for programming problems.

Still, problems with variable input can be met not only in programming but in other fields too. For example, several variants of a quiz may include the same task with different numeric values.³

If there are several variables V_1, \dots, V_N each of them having its own domain D_i ; then the whole variable domain D_{var} can be represented as a direct product of these domains:

$$D_{var} = [D_1 \times \dots \times D_N].$$

The dimension of the domain D_{var} is the sum of dimensions of D_1, \dots, D_N :

$$\dim D_{var} = \dim D_1 + \dots + \dim D_N.$$

² Various faults of problem statements were studied in [5].

³ In such a case, each variant can be checked as an independent one and, therefore, can represent the single-answer case.

We can make a *section* of the domain D_{var} by taking an actual value for each variable. With thus restricted input domain, the initial problem turns into a problem with no variables. Thus, we can reduce a problem with a variable input to a problem with the invariable input.

Constant and Known Number of Variables. Now let us consider one variable V_i . The dimension of its domain can vary in a wide range from 1 to any value having the practical sense. Multidimensional input is common in programming, where the *dimension* denotes the number of variable's components.

Constant bounds. As a rule, the lower and upper bounds for possible dimensions of a variable's domain are specified explicitly for all variables in the problem's description; they are known before the checking process has started. They can be considered as constants belonging to the invariable input D_{inv} .

Variable bounds. The bounds of a variable can also vary. Then they belong to D_{var} and must have bounds too. Let us call the changeable bound the *sub-bound*.

An example of such a situation is “ N integers $A_1 \dots A_N$ are given ($1 < N < 100$)...” Here variable A consists of N components $A_1 \dots A_N$ and, therefore, its dimension $dim A = N$. Variable N is the sub-bound of the current dimension, and 100 is the invariable upper bound common for all possible sub-bounds. Note that each A_i must have its own upper and lower limits too, but none are mentioned in this example.

So, the sub-bound is variable but becomes known when the check starts.

Indefinite Number of Variables. Now let us consider the case when the actual dimension of a variable's domain stays unknown until the end of checking.

Example is “No more than 100 integers are given...” Here we do not know how many components $A_1 \dots A_N$ the actual input has. We can preliminary write all of them down and count them; then we will know the current sub-bound N (not given but calculated). Thus, we return to the case of a known sub-bound. On the other hand, we can process these components not using the value of N at all. The difference can be illustrated by cycles

for 1 to N do... and do... until <the end is detected>

Another example of the situation when the number of variables must be retrieved from the input is “A graph is specified by the list of its edges, which are pairs of vertex numbers”.

Theoretically both the number of variants and their range can be infinite. Nonetheless, practically it is impossible due to restrictions of checking and computer representation.

3.3. Exemplar Input and Output

To check solutions automatically, we must have the exemplar input and the exemplar output.

Even in the case of finite domain D , it is too generous to check all of its members (which are the problem's possible valid inputs). It is sufficient to apply the exemplar method only to some characteristic representatives.

Partial cases. Domain D of all valid inputs can be split into equivalence classes.

Input data belong to the same equivalence class if being produced by the exemplar method they generate the same (or equivalent) outcomes. Each equivalence class is considered to be a *partial case*. Only one representative from each partial case is sufficient for the exemplar input.⁴

⁴ For more details, see, for example, [7, 11 or 12].

Unlike Beizer [7] or Myers [11, 12] who apply the equivalence partition to the domain of all possible inputs (and, thus, the affiliation with an equivalence class only show whether a variable belongs to the valid range), we only consider the domain of all valid inputs.

Restriction of the input. If domain D is infinite (refer to Section 3.2) then some (or even all) of the equivalence classes can be infinite too. If the number of classes is finite, getting one representative from each class forms a finite set of exemplar inputs.

Still, there can be infinite number of equivalence classes. To lessen this number, additional restrictions should be imposed on the domain of valid input data.

Automation. The partition of the domain D into equivalence classes can be done manually basing on the characteristics of the subject domain and the problem itself or automatically through the inner properties of the exemplar method. Still, the thorough discussion of this question is the topic of another article.

Check Cases. Having an exemplar method, one can trace all partial cases it processes. Since all inputs from an equivalence class are interchangeable, the representatives can be selected randomly.

Let a *check case* be a pair of some exemplar input and the corresponding exemplar output. We refer to a pack of check cases as a *check set*. Here we follow the analogy with *test cases* and *test sets* in programming.⁵

A check case is a set of points representing a section of domain D . Therefore, a check case is a sub-problem of the initial problem where all variables in the D , C and R have actual values.

3.4. Checking and Judging

The next step is to compare the acquired output with the corresponding exemplar output or the exemplar outcome.

$|R|$ is finite. If the problem admits only finite number of correct answers, the comparison can be easily performed by verifying the coincidence. In this case, the exemplar outcome should consist of one or several exemplar outputs. Let us also note that a poly-dimensional output brings almost no difference into the result-checking procedure.

$|R|$ is infinite. This case is more difficult. We cannot practically list all possible outputs; therefore, the “comparison” should mean performing a special checking formula, which depends on the type of the valid outputs. For example, we can ascertain that “a real Z belongs to the $[0..100]$ interval” by checking that both $Z \geq 0$ and $Z \leq 100$ are true.

A poly-dimensional output can demand more complex formulas. For example, the result “a point on a plain with coordinates (x, y) belongs to the circumference with the center in $(0, 0)$ and radius A ” can be checked with the help of the pair of inequalities $A^2 - e \leq x^2 + y^2 \leq A^2 + e$, where e is an admissible (and strictly specified in the problem’s description) error.

If the author of the problem would rather avoid such difficulties, the problem statement should be revised and the type of the output changed.

Judging. If the acquired output coincides with the example output (when $|R|$ if finite) or meets the differently stated conditions (when $|R|$ if infinite), the *check case result* is correct. Otherwise, it is incorrect.

After all check cases are processed, a judgment about the correctness of the whole method can be formed. And there are two ways for this.

⁵ Mostly, experts in programming (see, for example, [11, 12, 14, 15]) use the term *test suite* to name a pack of test cases. Still, *ISO/IEC/IEEE 24765:2010 International Standard — Systems and software engineering — Vocabulary* [10] does not mention this word at all. Instead, it uses the *test set* (3.3091). So we use the *test set* as the synonym of the *test suite* too.

The first way is the dichotomy “*all check case results are correct*” vs. “*at least one check case result is incorrect*”. The method is considered correct if and only if all its check case results are correct.

The second way is a gradation based upon the number of correct check case results. The metric for this gradation can be determined in various ways. For example, in an equipollent metric, each check case gives 1 point or $100/n$ percents of the result. On the other hand, in a weighted metric, check cases make different contributions to the result. On this way, a method can be *more correct* or *less correct* than the other method, according to their metric values.

Irrespectively to its actual correctness or erroneousess, the exemplar outcome is the base for judging about correctness of a method under examination. Therefore, it is important to eliminate the possibility of errors in the exemplar method and the exemplar outcome. And here an automated system for preparation of problem complexes can be of great use (see Section 4).

4. Automation

Now let us look at a problem as the subject for automated checking.

4.1. Problem complexes

The *problem complex* should include (refer to [2] or [3]):

- **Description** of the problem;
- **Specifications** of a valid input and output;
- A **check set**, which is a pack of exemplar *input-output* pairs;
- An **exemplar solution method**.

The first and the second parts are “exterior”. Contestants may see them. The third and the fourth parts, on the contrary, are for the inner use of checkers and judges only.

Problem’s Description. The full description of a problem must contain, explicitly or implicitly, the following parts (for more details, refer to [5]):

- **Introduction;**
- **Definitions**, agreements, terms, if necessary;
- **Statement.** A formalized presentation of the problem, its conditions and restrictions;
- **Task.** Requirements whose fulfilment means that the problem is solved;
- **Formats for the input and output data;**
- **Example** of a correctly written down solution and result.

Specifications. Input and output data specifications, restrictions and clauses are specified in the problem’s description written in a natural language. A textual analysis can automatically extract the preliminary specification list, which should be revised manually [2, 3].

Check set and Exemplar method were considered earlier in Sections 2.2 and 3.3.

4.2. Preparation of problem complexes

The process of preparing a problem complex is iterative: creating or changing each part (see Section 4.1) can impel changes in any other parts.

Stage 1. According to the original idea of the problem, the author of the problem’s description
a) defines restrictions R on all variables in use;

b) makes a preliminary decomposition D^* of the valid input data domain D into equivalence classes showing all possible partial cases;⁶

and c) sets specifications S for the input and output data.

Stage 2. From specifications S and decomposition D^* , a check set CS is prepared. This can be done manually or automatically with the help of an automated test-preparing system (for more details, refer to [1]).

Stage 3. An exemplar solution ES is written (manually) and is debugged with the help of the check set CS .

To reduce the number of possible errors, it is recommended that problem complexes are created collegially. If two authors A_1 and A_2 write two different exemplar solutions ES_1 and ES_2 and use two check sets CS_1 and CS_2 for debugging, both of them fulfil stages 1 to 3, and then Stage 4 arises.

Stage 4. Two check sets are compared and combined. Both solutions ES_1 and ES_2 must be tested on the united check set $CS = CS_1 \cup CS_2$. If no cross-errors were detected, it is necessary to ascertain that this united check set agrees with the final decomposition D and meets the final specifications S . Most likely, the united check set CS will be superfluous; and, therefore, some surplus check cases should be excluded.

The 4th stage can also be useful for the individual preparation of a problem complex. The author's initial check set and the automatically generated check set can be treated as CS_1 and CS_2 .

4.3. Automated Systems for Preparation of Problem Complexes

Irrespectively to its actual correctness or erroneousess, the exemplar outcome is the base for judging about correctness of the method under examination (see Sections 2.2 and 3.4). Therefore, it is important to eliminate the possibility of errors in the exemplar method and the exemplar outcome. And here an *automated system for preparation of problem complexes* (ASPPC) can be of great use. Such systems are described in detail in [2, 3].

At Stage 1 (see Section 4.2), an ASSPPC should

- Extract a preliminary set of specifications S_0 from the description of the problem by means of the textual analysis
- Check the consistency of specifications
- Extract possible information about boundaries, exceptional points and so on from the problem's description and specifications S_0
- Construct a preliminary partition P of the valid data domain D into equivalence classes (basing on the specifications S_0 and additional information provided by the author(s) of the problem)
- Compare, join and intersect partitions P_1 and P_2

At Stage 2, an ASPPC should

- Create exemplar inputs basing on the partition P
- Ascertain that the author's check set CS_0 covers the partition P
- Verify that the check set CS_0 meets specifications S_0 and restrictions R

⁶ On the very first stage, it is impossible to use the exemplar solution since it is not created yet.

If the necessity to change the initial specification set S_0 is detected, the process of creating an exemplar check set should be started anew, now basing on the renewed specification set S_0' .

At Stage 3, with the exemplar outputs generated with the help of the exemplar solution (method), an ASSPPC should

- Check that the exemplar outputs meet specifications S (which is the final version of the specification set)

At Stage 4,

- Define the equivalent check cases
- Propose variants of reducing the joint check set

Conclusion

Our aim is to automate processes of preparing the problem complexes in any subject field, in order to make the automated checking easier and its use wider.

We have considered notions *checking* and *correctness* and have ascertained that not only programming problems but problems from other subject fields too can be checked automatically.

We have studied processes that constitute the preparation of a contest or a quiz and the check of their results and have shown which of these processes can be automated.

We have shown that automated systems make the preparation of problem complexes easier and more accurate, especially in case of co-working.

The future aims of our work are a) to design means for the coverage analysis of the partitions created automatically from exemplar solutions, b) to develop the mathematical apparatus for operations with partitions of different types, c) to create means of partition analysis in order to ascertain that all important equivalence classes inspired by the current subject field are considered, and d) to develop means that can suggest additional partition variants basing on the analysis of the type, the power and the dimensions of the input data.

References

1. *Andreyeva T.* Automated generation of test sets // Science in the Modern Information Society IX: Proceedings of the conference. North Charleston, USA, 2016. P. 110–112.
2. *Andreyeva T.* Automated preparation of problem complexes // Science today: Theoretical and practical aspects: Proceedings of the conference. Vologda, 2017. Part 1. P. 25–26. URL: http://volconf.ru/files/archive/01_27.12.2017.pdf.
3. *Andreyeva T.* Automated preparation of problem complexes for programming contests // Science. Informatisation. Technologies. Education XI: Proceedings of the international scientific-practical conference. Ekaterinburg, 2018. URL: <http://nito.rsvpu.ru/files/nito2018/nito2018.pdf>.
4. *Andreyeva T.* Serial problems in programming // Perspectives of information systems V. Educational informatics section. Proceedings of the international conference. Novosibirsk, 2003. P. 2–4. (in Russian)

5. *Andreyeva T.* Structure and classification of contest problems' texts. Computer instruments in education. 2002. № 3–4. P. 50–59. URL: <http://ipo.spb.ru/journal/index.php?article/223/> or <http://ipo.spb.ru/journal/index.php?magazines/2002/34/e/>.
6. *Ashby W.R.* Introduction to cybernetics. London : Chapman & Hall, 1957. 156 p.
7. *Beizer B.* Black-box testing: Techniques for functional testing of software and systems. N. Y., USA : John Wiley & Sons, Inc, 1995.
8. *Eysenck H.J.* Know your own I. Q. Penguin Books. Harmondsworth, Eng. 1962.
9. *Floyd R. W.* Assigning meanings to programs // Mathematical Aspects of Computer Science. Proceedings of Symposium on Applied Mathematics. 19. American Mathematical Society. 1967. P. 19–32.
10. ISO/IEC/IEEE 24765:2010 International Standard. Systems and software engineering — Vocabulary. IEEE. DOI:10.1109/IEEESTD.2010.5733835.
11. *Myers G.J.* The art of software testing. N. Y. : John Wiley & Sons, 1979.
12. *Myers G.J., Badgett T., Sandler C.* The art of software testing. 3rd ed. N. Y. : John Wiley & Sons, 2011.
13. *Ponrod C.* The study of black-box testing technique for collateral management system. Bangkok : Mahidol University Press, 2014.
14. *Singh Y.* Software testing. UK : Cambridge University Press, 2012. Chapter 1.3.4.
15. *Spillner A., Linz T., Schaefer H.* Software testing fundamentals: A study guide for the certified tester exam. 4th ed. Santa Barbara : Rocky Nook Inc, 2014.

ИСПОЛЬЗОВАНИЕ ЭЛЕКТРОННЫХ ТАБЛИЦ ПРИ РЕШЕНИИ ЗАДАЧ НА ПРОЦЕНТЫ, СПЛАВЫ И СМЕСИ ПРИ ПОДГОТОВКЕ К ОГЭ В РАМКАХ ИНТЕГРИРОВАННОГО КУРСА ПО МАТЕМАТИКЕ И ИНФОРМАТИКЕ

О. В. Воронцова

учитель математики, МБОУ СОШ № 80 г. Новосибирска,
630098, Новосибирск, ул. Энгельса, 6, e-mail: vov1975_3@mail.ru

Т. И. Роженко

учитель информатики, МБОУ СОШ № 80 г. Новосибирска,
630098, Новосибирск, ул. Энгельса, 6, e-mail: tiroz@ngs.ru

Аннотация. Умение решать задачи является одним из основных показателей уровня математического развития, глубины освоения учебного материала. Поэтому любой экзамен по математике (ОГЭ или ЕГЭ), любая проверка знаний содержат в качестве основной и, пожалуй, наиболее трудной части решение задач. В данной статье рассмотрены приемы решения задач на растворы, смеси и сплавы методами математики и информатики.

Ключевые слова: смесь, сплав, процент, электронная таблица

Умение решать задачи является одним из основных показателей уровня математического развития, глубины освоения учебного материала. Поэтому любой экзамен по математике (ОГЭ или ЕГЭ), любая проверка знаний содержат в качестве основной и, пожалуй, наиболее трудной части решения задач.

Научить решать большинство текстовых задач, содержащихся в открытом банке, можно практически любого выпускника. Конечно, при этом определяющими факторами являются желание и стремление ученика и владение простыми вычислительными навыками. В данной статье мы хотим показать прием решения задач на растворы, смеси и сплавы не только с помощью математики, но и информатики. Нам кажется, что именно такой тип задач вызывает основные трудности, поэтому для лучшего усвоения надо рассматривать его как по математике, так и по информатике.

В качестве практического материала нами были использованы задачи «от составителей» из «открытого банка заданий».

Существует много способов решения задач на растворы, смеси и сплавы. Но мы хотим остановиться на одном из них, который, по нашему мнению, самый простой для усвоения решения таких задач — табличный.

Итак, объект исследования — математика и информатика.

Предмет исследования — задачи «на смеси, сплавы, растворы».

Цель работы — научить учащихся решать и программировать задачи на смеси и сплавы.

Чтобы лучше понимать условия задач, необходимо знать следующие понятия.

1. **Что такое концентрация вещества в растворе, смеси, сплаве?** Концентрация вещества в растворе (смеси, сплаве) — это отношение массы или объема вещества к массе или объему всего раствора (смеси, сплава). Как правило, концентрация выражается в процентах.

2. **Что такое процент? Процент** — это сотая доля числа. Она может выражаться либо в виде десятичной дроби, либо в виде процента.

3. **Что такое масса раствора, смеси, сплава?** Масса раствора (смеси, сплава) равна сумме масс всех составляющих. При смешивании нескольких растворов (смесей, сплавов) масса нового раствора становится равной сумме всех смешанных растворов. Масса растворенного вещества при смешивании двух растворов суммируется.

Задачи на смеси и сплавы бывают двух основных видов:

1) две смеси определенной массы с некоторой концентрацией вещества сливают вместе. Нужно определить массу и концентрацию этого вещества в новой смеси;

2) в некоторый раствор с некоторой концентрацией вещества добавляют, например, чистую воду (с нулевой концентрацией этого вещества). Нужно определить, какой стала концентрация вещества.

Алгоритм решения задачи на сплавы, растворы и смеси:

- изучить условия задачи;
- выбрать неизвестную величину (обозначить ее буквой);
- определить все взаимосвязи между данными величинами;
- составить математическую модель задачи (выбрать способ решения задачи, составить пропорцию или уравнение относительно неизвестной величины) и решить ее;
- провести анализ результата.

Рассмотрим несколько задач и решим их с помощью таблицы.

Задача 1. Первый сплав содержит 10 % меди, второй — 40 % меди. Масса второго сплава больше массы первого на 3 кг. Из этих двух сплавов получили третий сплав, содержащий 30 % меди. Найдите массу третьего сплава. Ответ дайте в килограммах.

В этой задаче примем за x массу первого сплава и выразим через него второй и третий.

	Масса сплава	Процентное содержание вещества	Масса вещества
Первый сплав	x	10 % = 0,1	$0,1 x$
Второй сплав	$x + 3$	40 % = 0,4	$0,4 (x + 3)$
Третий сплав (результат слияния первых двух)	$2x + 3$	30 % = 0,3	$0,3 (2x + 3)$

Составим и решим уравнение:

$$0,1x + 0,4(x + 3) = 0,3(2x + 3)$$

$$0,1x + 0,4x + 1,2 = 0,6x + 0,9$$

$$0,3 = 0,1x$$

$$x = 3 \text{ (кг) масса первого сплава.}$$

$$2 \times 3 + 3 = 9 \text{ (кг) масса третьего сплава.}$$

Ответ: масса третьего сплава 9 кг.

Задача 2. Смешав 60%-й и 30%-й растворы кислоты и добавив 5 кг чистой воды, получили 20%-й раствор кислоты. Если бы вместо 5 кг воды добавили 5 кг 90%-го раствора той же кислоты, то получили бы 70%-й раствор кислоты. Сколько килограммов 60%-го раствора использовали для получения смеси?

Для решения этой задачи будем составлять две краткие записи. До слов «если бы вместо 5 кг воды...» и после. Примем за x массу первого раствора, а за y — массу второго.

	Масса раствора	Процентное содержание вещества	Масса вещества
Первый раствор	x	60 % = 0,6	$0,6x$
Второй раствор	y	30 % = 0,3	$0,3y$
Вода	5	0	0
Третий раствор (результат слияния первых двух)	$x + y + 5$	20 % = 0,2	$0,2(x + y + 5)$

Составим уравнение с двумя переменными: $0,6x + 0,3y + 0 = 0,2(x + y + 5)$

	Масса раствора	Процентное содержание вещества	Масса вещества
Первый раствор	x	60 % = 0,6	$0,6x$
Второй раствор	y	30 % = 0,3	$0,3y$
Раствор	5	90 % = 0,9	4,5
Третий раствор (результат слияния первых двух)	$x + y + 5$	70 % = 0,7	$0,7(x + y + 5)$

Составим уравнение с двумя переменными: $0,6x + 0,3y + 4,5 = 0,7(x + y + 5)$.

Объединив полученные два уравнения в систему и решив ее, получим x . Это и будет показатель того, сколько килограммов 60%-го раствора использовали для получения смеси. Ответ: 2 кг.

Для нас важным является практический аспект в решении математических задач, поскольку в настоящее время даже хорошо знающий математику, но не умеющий применять математические методы на компьютере выпускник, не может считаться успешным. Поэтому необходимо научить детей проводить требуемые вычисления на компьютере.

Наиболее универсальным для решения рассматриваемого класса задач является табличный процессор Excel.

Для решения задачи 1 в электронных таблицах изменим форму расчетов. В начало таблицы поместим известные данные, во второй столбец расчеты, в третий — расчет результата:

	A	B	C	D
1		Процентное содержание вещества		Масса сплава
2	Первый сплав	0,1		= C5/C6
3	Второй сплав	0,4	= B3 × 3	
4	Третий сплав	0,3	= B4 × 3	
5			= C3 – C4	
6			= B4 × 2 – B3 – B2	

Получаем:

	A	B	C	D
		Процентное содержание вещества		Масса сплава
1	Первый сплав	0,1		3
2	Второй сплав	0,4	0,9	
3	Третий сплав	0,3	1,2	
4			0,3	
5			0,1	

Задачу 2 решаем в Excel матричным методом. Используем полученную систему линейных уравнений. Составляем матрицу, помещаем ее в Excel:

	A	B	C	D
1	0,4	0,1		1
2	0,1	0,4		1

Для нахождения обратной матрицы используем функцию МОБР:

2,666666667	-0,666666667
-0,666666667	2,666666667

Для умножения обратной матрицы на массив свободных членов используем функцию МУМНОЖ. Получаем массив из двух двоек, которые являются корнями системы уравнений.

Умение решать системы уравнений в Excel — очень полезный навык для учащихся. Они учатся проверять математическое решение на компьютере, искать подходящие методы, что расширяет кругозор и учит решать задачи различными способами. Ученики изменяют процентное содержание вещества в сплаве и наблюдают, как меняется результат.

Таким образом, учащиеся закрепляют знания, полученные на уроках математики, и осваивают способы решения задач на компьютере.

Заключение

В задачах этого типа прослеживается системный подход к решению задач. Происходит успешная отработка и закрепление интеллектуальных умений (анализ, синтез, аналогия, обобщение, конкретизация и *т. д.*). Данная система задач на смеси, растворы и сплавы была апробирована в ходе КПР (контрольно-проверочной работы) по математике в 8 классе в 2016–17 учебном году. Опыт показал, что учащиеся, не знавшие вначале, как подойти к решению этих задач, в конце темы успешно заполняли таблицу и получали верный ответ.

Список литературы

1. Открытый банк заданий ОГЭ. URL: <http://www.fipi.ru/content/otkrytyy-bank-zadaniy-oge>.
2. Каталог заданий. Задачи на проценты, сплавы и смеси: URL <https://oge.sdamgia.ru>.
3. *Гельман В. Я.* Решение математических задач средствами Excel : практикум. СПб. : Питер, 2003.

UDK 519.711.2

USING SPREADSHEETS TO SOLVE PROBLEMS ON PERCENTAGE, ALLOYS AND MIXTURES WHEN TRAINING SCHOOLCHILDREN FOR THE BASIC STATE EXAM WITHIN THE INTEGRATED COURSE ON MATHEMATICS AND COMPUTER SCIENCE

O. V. Vorontsova

Teacher of mathematics, MBOU SOSh № the 80th Novosibirsk,
630098, Novosibirsk, Engels St., 6, e-mail: vov1975_3@mail.ru

T. I. Rozhenko

Teacher of computer science, MBOU SOSh № the 80th Novosibirsk
630098, Novosibirsk, Engels St., 6, e-mail: tiroz@ngs.ru

Abstract. The ability to solve tasks is one of the key indicators of the level of the mathematical development and the depth of study a training material. Therefore, any examination in mathematics (Basic State Exam or Unified State Exam) or any knowledge check contains the main and, perhaps, the most difficult part — solving tasks. This paper discusses techniques for solving tasks in solutions, mixtures, and alloys by the methods of mathematics and computer science.

Keywords: mixture, alloy, percent, spreadsheet

References

1. Otkrytyi bank zadaniy OGE. URL: <http://www.fipi.ru/content/otkrytyy-bank-zadaniyoge>.
2. Katalog zadaniy. Zadachi na procenty, splavy i smesi: URL <https://oge.sdamgia.ru>.
3. *Gel'man V. Ia.* Reshenie matematicheskikh zadach sredstvami Excel : praktikum. SPb. : Peter, 2003.

КОНСТРУИРОВАНИЕ УЧЕБНЫХ ИГР

Л. В. Городняя

канд. физ.-мат. наук, ст. науч. сотрудник,
Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, пр. Акад. Лаврентьева, 6,
доц., Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 2, e-mail: lidvas@gmail.com

С. Е. Демидов

студент, Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 2, e-mail: sersh_96@mail.ru

Н. В. Шаченко

студент, Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 2, e-mail: shachenko153@gmail.com

Аннотация. В статье описан подход к созданию учебных игр и тренажеров методом визуализации деталей игры на клетчатой доске. Такой подход достаточен для демонстрации феноменов организации параллельных процессов и других парадигм программирования.

Ключевые слова: учебные игры, параллельные процессы, обучение программированию

Введение

Происходит переход от доминирующей парадигмы императивно последовательного программирования к парадигмам организации асинхронных и синхронизованных параллельных процессов в информационных сетях над многопроцессорными комплексами с незащищенной общей памятью. Круг возникающих проблем достаточно ярко проявился на задачах суперкомпьютерных параллельных вычислений. Многие затруднения с параллельным программированием имеют образовательный характер. Обучение параллельному программированию осложнено дистанцией между уровнем абстракций, в которых описываются решения сложных задач на современных языках программирования, и уровнем реализации аппаратуры и архитектурных принципов управления параллельными вычислениями (потактовая синхронизация, сигналы, семафоры ...). Проблемы организации работоспособного взаимодействия процессов следовало бы осваивать при начальном знакомстве с информационными технологиями и про-

граммированием. Есть гипотеза, что полезнее познакомиться с миром параллелизма в детские годы, до того, как начинается ознакомление с программированием.

1. Общее представление и предшественники

Сами по себе принципы параллелизма не являются сложными [1]. Все мы живем в мире множества взаимодействующих процессов и владеем интуитивным имплицитным невербализованным знанием о принципах их взаимодействия, поэтому правильная вербализация такого знания поможет сознанию, еще не привыкшему все сводить к однозначности и последовательностям [2], открыть для себя понимание многообразия миров параллелизма. Нужна система обучения, приспособленная не только к ознакомлению с отдельными эффектами параллельных вычислений, но и поддерживающая конструирование учебно-игровых тренажеров, пригодных для самостоятельного осознания особенностей взаимодействия процессов и для экспериментов по формированию навыков подготовки работоспособных программ и понимания новых возможностей аппаратуры.

В конце 1970-х гг. в работах Г. А. Звенигородского по безмашинному обучению младших школьников программированию была серия задач по демонстрации взаимодействия процессов. Использовался образ исполнителей класса «Машинист»¹, работающих на железнодорожной станции, формирующих составы из нужных вагонов [3]. Подобные сюжеты таких задач были опубликованы и в книге В. А. Вальковского, В. Э. Малышкина из серии «Информатика» в начале 1990-х гг.

В середине 1990-х гг. студенты ММФ НГУ С. Коваль и М. Агамиров реализовали две демонстрационные программы на языке Java, показывающие отдельные эффекты взаимодействия процессов при решении задач на клетчатой доске, используя образы исполнителей класса «Муравей» (уборка мусора, сбор фуража, охрана входов в муравейник). На летних школах юных программистов, ежегодно проводимых ИСИ СО РАН (<https://www.iis.nsk.su/edu/school>), обычно бывают выполняемые школьниками проекты по реализации игр, сюжеты которых содержат взаимодействия разных исполнителей (пчелиный улей, теремок, сбор грибов в лесу, погрузка урожая и др.). В мировой практике изучения параллельных вычислений и ознакомления с проблемами взаимодействия процессов уже признан целесообразным переход от старших курсов к младшим, что можно рассматривать как тенденцию снижения возрастного барьера. Новые учебные языки программирования содержат конструкции для представления параллельных потоков на основе знакомства с функциональным программированием, изучаемым раньше императивного программирования (OZ, Mozart, Синхро) [4].

2. Программная поддержка учебных игр

Одна их сложностей вербализации интуитивных образов связана с разнообразием индивидуальных интересов и успешного опыта. Поэтому программная поддержка такой работы должна быть устроена как система из ядра и оболочки, поддерживающих оперативную настройку на созвучные, привлекающие образы [5, 6]. Ядро обеспечивает поддержку базовой семантики учебных игр подобно абстрактной машине языка программирования. Оболочка создает расширение ядра для конкретной игры подобно абстрактному синтак-

¹ Среди учащихся были дети железнодорожников — жители Первомайского района Новосибирска.

сису языка программирования, дополненному пользовательским интерфейсом с информационным наполнением: картинки, тексты и вспомогательные процедуры по сюжетам, выбранные в соответствии с индивидуальными особенностями. Эффект параллельных вычислений визуализируется как одновременная перерисовка клеток, представляющих разные процессы.

Ядро содержит информацию об игровом поле и всех объектах и способно проводить симуляцию заранее заданных действий объектов так, что каждый объект выполняет свою последовательность действий. Прорисовки действий совместных процессов исполняются одновременно, благодаря чему достигается возможность демонстрации явлений, присущих только миру параллелизма. Модуль ядра используется для создания сцен, создания объектов, настройки объектов, наполнения сцен объектами, задания объектам инструкций, симуляции в соответствии с этими инструкциями и предоставления результатов симуляции. Помимо этого, ядро должно содержать в себе систему классификации и обработки ошибок. Таким образом оболочка сможет получать данные о причинах некорректной или неожиданной работы системы и осуществлять действия, направленные на устранение ошибок и восстановления работоспособности системы. Подразумевается, что каждая элементарная команда выполняется за фиксированный промежуток времени, равный для всех элементарных команд. Отдельный процесс для каждого исполнителя состоит из очереди шагов. На каждом шаге процесса исполняется первая элементарная команда из его очереди и удаляется из очереди. Сценарий взаимодействия исполнителей выполняется до того момента, пока не опустеют очереди команд каждого исполнителя. Возможен дискретный запуск симуляции, при котором будет исполнен только один шаг. Ядро позволяет:

- создавать свободное клеточное поле размером $N \times M$, где N и M задаются пользователем;
- размещать в произвольные клетки поля элементы (неподвижная «стена», подвижный «ящик» и стандартный робот-исполнитель);
- задавать каждому исполнителю предопределенную последовательность команд, которую можно изменять в процессе игры.

Оболочка передает ядру внутренние команды по одной или в виде сценария, представляющего собой комплект слоев, состоящих из набора параллельных потоков. Первый случай соответствует нажатию кнопок, второй — результату программирования взаимодействия процессов или редактирования ранее полученных протоколов хода игры. Оболочка должна предоставлять грамотный графический интерфейс и являться буфером между вводом команд и работой ядра, облегчающим процесс управления игровыми объектами и позволяющим в понятной человеку форме визуализировать происходящее. Возможны различные реализации оболочек вокруг существующего интерфейса ядра. Имеется возможность запоминания истории действий и произвольного «передвижения» внутри истории действий, произошедших на клеточном поле.

Любая учебная игра представляет собой набор уровней с возможностью их редактирования оператором. Каждый уровень представляет собой клеточное поле с заданными на нем характеристиками и сценариями. Поддержана возможность выбора анимации с помощью простых изображений или спрайтов. Можно выделять специальные виды объектов, например, «Ящик» — это объект, который занимает одну клетку и не имеет своих активных действий, но его можно двигать, а объект «Стена» неподвижен. Задаются не только размер поля, специальные объекты и их координаты, роботы-исполнители, их команды, но и условия выполнения команд. Каждый уровень имеет свои условия успешного прохождения в соответствии с постановкой задачи, например, успешного собирания роботами всех объектов определенного типа.

Самой важной функцией является возможность создания новых действий для роботов на основе более простых (атомарных) с возможностью последующего использования, включая совместные действия. Редактор позволяет:

- задавать и изменять размеры поля;
- изменять количество роботов и объектов;
- изменять виды роботов доступных на данном уровне;
- создавать собственных роботов и задавать их возможные команды;
- задавать собственные команды и добавлять их в стандартное меню команд;
- сохранять созданный уровень в списке стандартных уровней;
- определять и добавлять условие завершения уровня (успех или провал).

Такие возможности редактирования позволяют на небольшом комплекте команд ядра конструировать широкий спектр учебных игр, варьируя визуализацию действий и набор сценариев игры. Простейшая система команд для роботов-исполнителей содержит базовые действия, такие как:

- передвинуться вверх;
- передвинуться влево;
- передвинуться вправо;
- передвинуться вниз;
- пропуск хода.

3. Примеры прорисовки учебных игр

Подготовка учебной игры может выглядеть как клетчатое поле с размеченными позициями и изображением персонажей (примеры прорисовки будут представлены в презентации к докладу). Прорисовку персонажей можно изменять. Можно изменить размеры поля и расположение командного меню, а затем добавить новых исполнителей. Часто исполнителей и объекты размещают в одну клетку, но можно делать и крупных исполнителей, реализуя их прорисовку как одновременное изменение вида комплекта клеток. Можно завести поле для изображения доступных персонажей. Меню можно сделать бестекстовым, в виде общепонятных знаков. При желании можно фон игры сделать достаточно зрелищным и динамическим, а командам дать текстовые варианты в соответствии со сценарием игры.

Например, можно сделать игру «Философы», на которой демонстрируются разные эффекты организации взаимодействующих процессов. Пример обедающих философов является одной из классических иллюстраций проблем взаимодействия параллельных процессов. Он примечателен простотой и наглядностью концепции, а также наличием большого числа возможных тонких решений, изученных на уровне теории организации процессов. Задание поведения философов через встроенный текстовый редактор позволяет вручную настраивать количество философов, а также последовательность действий из следующего набора команд:

- 1) размышлять, пока не освободится левая вилка. Когда вилка освободится — взять ее;
- 2) размышлять, пока не освободится правая вилка. Когда вилка освободится — взять ее;
- 3) приступить к еде;
- 4) положить левую вилку;
- 5) положить правую вилку.

При выборе действия исполнитель пытается его осуществить в течение неограниченного количества времени, поэтому иллюстрация дедлока получается наглядной. Оператор-игрок имеет возможность отмены хода, в результате поле возвращается к предыдущему состоянию, а сценарий игры можно подкорректировать.

Система может быть расширена добавлением новых команд, встраивание которых не потребует дополнительных усилий. Подобным образом удастся продемонстрировать такую проблему параллелизма, как *состояние гонки (race condition)*: Это гонка между двумя роботами за обладание соседней клеткой. Два робота, разделенные ею, пытаются захватить ее. Так как объекты исполняют свои действия в случайном порядке, то нельзя предугадать, какой из них займет клетку. В результате, кто-то достигнет своего, а кто-то нет. Возможны варианты *взаимной блокировки (deadlock)*, реализованной в виде двух множеств роботов, каждое из которых хочет занять соседние клетки, занятые другим множеством. В таком примере клетки рассматриваются как ресурсы.

4. Оформление ветвлений по ходу игры

Одной из проблем при создании игр детьми является сложность представления и понимания сложных логических условий управления действиями персонажей. Эта проблема обусловлена противоречием между интуитивной и формальной оценкой истинности формул. Интуитивно истинность связки «&&» оценивается выше, чем истинность составляющих, а реально она ниже. Противоречие, имеющее лингвистический характер, замечено в исследованиях нобелевского лауреата Д. Канемана.

Формат и примеры встроенных команд имеет вид:

```
// <Условие → Изменение_разметки_игрового_поля>
```

На языке программирования это можно выразить в форме условной команды:

```
движение_от_Двери_к_Столу (int i, j) =
{ ( i < N/2 && д [i][j]==2 && д [i+1][j]==0 )
  → ( д [i+1][j]=2 ; д [i][j]=0 ) ,
  // Для Ф2 если нужная клетка свободна, то Ф2 идет вверх

( i > N/2 && д [i][j]==2 && д [i-1][j]==0 )
  → ( д [i-1][j]=2 ; д [i][j]=0 )
  // Для Ф1 если нужная клетка свободна, то Ф1 идет вниз
}
```

Несколько понятнее можно написать более традиционное условное выражение:

```
если ( i < N/2 && д [i][j]==2 && д [i+1][j]==0 ) то ( д [i+1][j]=2 ; д [i][j]=0 )
```

Возможно, понятнее будет техника таблиц решений, на которую в начале 1970-х гг. обратил внимание Дж. Шварц при создании языка Setl. По аналогии с ней можно представлять таблицы решений, в которых левая колонка содержит простые выражения, участвующие в выборе решений, а в остальных столбцах располагаются значения этих выражений, соответствующие каждому выбору.

Пример таблицы решений

если	$\Phi 2$: ему нужна клетка сверху, если она свободна, то $\Phi 2$ идет вверх	$\Phi 1$: ему нужна клетка снизу, если она свободна, то $\Phi 1$ идет вниз
$i < (N/2-1)$	истина	–
$i > (N/2+1)$	–	истина
д [i][j]	2	2
д [i+1][j]	0	–
д [i-1][j]	–	0
то	(д [i+1][j]=2 ; д [i][j]=0)	(д [i-1][j]=2 ; д [i][j]=0)

Линеаризация таблицы решений для размещения в тексте программы может иметь вид текста, выравнивание колонок в котором достигается табуляцией, а строки выделяются концами строк.

РЕШЕНИЕ	$\Phi 2$		$\Phi 1$
$(i < (N/2-1))$	истина	–	
$(i > (N/2+1))$	НЕТ		ДА
(д [i][j])	2		2
(д [i+1][j])	0		–
(д [i-1][j])	–		0

$\Phi 1 = (д [i-1][j]=2 ; д [i][j]=0)$

$\Phi 2 = (д [i+1][j]=2 ; д [i][j]=0)$

Такой формат выглядит более понятным, свободным от ложных лингвистических ассоциаций.

Заключение

Эффективным методом решения проблемы организации параллельных вычислений может стать постепенное внедрение в программы обучения программированию инструментов для ознакомления с основами параллельных явлений. Наиболее эффективной данная методика окажется для школьников, получающих возможность в простой и доступной игровой форме увидеть и осознать базовые принципы параллелизма, его основные проблемы и приобрести навыки решения простейших задач, имеющих отношение к взаимодействию параллельно исполняемых процессов. Не менее важны навыки конструирования программ, обладающих заданным поведением.

Список литературы

1. Хоар Ч. Взаимодействующие последовательные процессы. М. : Мир, 1989.
2. Кушниренко А. Г., Лебедев Г. В., Сворень Р. А. Основы информатики и вычислительной техники : пробный учебник для средних учебных заведений. М. : Просвещение, 1990.

3. *Звенигородский Г. А.* Первые уроки программирования. М. : Наука, 1985.

4. *Городня Л. В.* Язык параллельного программирования Синхро, предназначенный для обучения. Новосибирск, 2016.

5. *Демидов С. Е.* Создание интерфейса оболочки обучающих параллелизму роботов : мат-лы 56-й Междунар. науч. студ. конф. «Студент и научно-технический прогресс». 2018. С. 72.

6. *Шаченко Н. В.* Проектирование и разработка ядра обучающей параллелизму системы роботов: мат-лы 56-й Междунар. науч. студ. конф. «Студент и научно-технический прогресс». 2018. С. 85.

UDK 004.43

DESIGNING EDUCATIONAL GAMES

L. V. Gorodnyaya

Ph.D., Researcher A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6,
Associate professor, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova st., 2, e-mail: lidvas@gmail.com

S. E. Demidov

Student, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova st., 2, e-mail: sersh_96@mail.ru

N. V. Shachenko

Student, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova st., 2, e-mail: shachenko153@gmail.com

Annotation. The purpose of the article is the approach to the creation of educational games and simulators by visualizing the details of the game on a checkered board. Such approach is sufficient to demonstrate the phenomena of organization of parallel processes and other programming paradigms.

Keywords: educational games, parallel processes, learning programming

References

1. *Hoare C. A. R.* Interacting sequential processes. M.: Mir, 1989.
2. *Kushnirenko A. G., Lebedev G. V., Svoren R. A.* Fundamentals of computer science and computing: A test textbook for secondary schools. Educational edition. M. : Prosveshcheniye publishing house, 1990.
3. *Zvenigorodsky G. A.* The first programming lessons. M. : Nauka, 1985.
4. *Gorodnyaya L. V.* Programming language SYNCHRO for learning. Novosibirsk, 2016.
5. *Demidov S. E.* Creating a shell interface for learning parallelism on the system of robots. 56th International Scientific Student Conference "Student and Scientific and Technological Progress". 2018. P. 72.
6. *Shachenko N. V.* Design and development of the core of the learning parallelism on the system of robots. 56th International Scientific Student Conference "Student and Scientific and Technological Progress". 2018. P. 85.

ПРОПУСК В БУДУЩЕЕ

М. В. Григоренко

учитель информатики, МБОУ Лицей № 130 им. акад. М. А. Лаврентьева,
630090, Новосибирск, ул. Ученых, 10, e-mail: margrig@yandex.ru

Аннотация. Статья посвящена вопросам подготовки будущих ИТ-специалистов в школе. В ней рассматривается дорожная карта по профориентации будущих студентов. В лицее есть определенная специализация и набор предполагаемых олимпиад для поступления.

Рассмотрены формы обучения, которые могут быть использованы для учащихся не только профильных, но и общеобразовательных классов в процессе выбора ИТ-специальности.

Ключевые слова: средняя школа, подготовка ИТ-специалистов, дистанционное образование, олимпиады

Одной из проблем цифровой экономики в области информационных технологий является отсутствие достаточного количества кадров.

Согласно исследованию Фонда развития интернет-инициатив (ФРИИ), к 2027 г. в России будет не хватать около 2 млн ИТ-специалистов. Официальных данных о том, сколько их на 2019 г., в открытых источниках не найдено.

По данным Росстата, в 2016 г. в сфере ИТ работали 1,05 млн человек (1,45 % трудоспособного населения), а в 2017 г. — 1,077 млн (1,49 % трудоспособного населения).

В ФРИИ утверждают, что для того, чтобы обеспечить кадровые потребности страны, набор на ИТ-специальности в высших и средних специальных заведениях должен расти на 40 тыс. человек в год. При этом на открытые согласно государственной политике места должны прийти не просто те студенты, которые набрали необходимые баллы по ЕГЭ, но те, кто способен обучаться, закончить свое образование и стать успешными специалистами в выбранной отрасли.

В данной работе мы не рассматриваем вопросы подготовки вузов по ИТ-специальностям. Давайте вернемся в школу, откуда все начинается.

«Необходимо заинтересовать детей информационными технологиями еще в школе и дальше вести ребят, мотивировать, развлекать. Мы не должны терять из вида тех, кто хочет заниматься ИТ профессионально, — говорит министр связи Никифоров, — На это сегодня направлены значительные усилия государства и бизнес-сообщества». Но научить тех, кому это неинтересно, крайне сложно. Как выбрать тех, кому это действительно необходимо? Чаще всего это учащиеся профильных классов, проявляющие интерес к информатике и программированию.

Но последнее время растет еще и число учащихся непрофильных классов, которые также проявляют интерес к информатике. Это может быть объяснено следующими факторами:

- физическое отсутствие на момент формирования профильных классов;
- неспособность определиться в раннем возрасте (конец 7-го класса);

– желание родителей;
 – инфантилизм в старших классах;
 – понимание того, что цифровизация развивается в любой отрасли, без информатики не обойтись в любой профессии (чаще всего это причина для наиболее продвинутых учащихся классов другого профиля).

У нас в лицее дорожная карта выглядит так:

Класс	Процесс	Направления деятельности для учащегося	Направления деятельности для учителя
2–6 класс	Предпрофильная подготовка	Посещение курсов по выбору	Информационная
7 класс	Появление предмета «Информатика»	Адаптация к предмету	Выявление учеников, занимающихся предметом или проявляющих к нему интерес
8 класс	Формирование профильных классов	Адаптация к классу. Знакомство с предметом на профильном уровне	Формирование групп по интересам в области информатики. Уточнение направления развития
9 класс	Выбор и сдача ОГЭ	Совместное формирование индивидуальной образовательной траектории, включая летний отдых. Выбор олимпиад из списка РСОШ	Менеджмент в образовании: формирование образовательной траектории, выбор направления, график олимпиад, формирование и тренинг команд
10 класс	Выбор предметной области	Участие в олимпиадах. Встречи с выпускниками, участие в круглых столах соответствующих вузов, экскурсии	Профориентация. Организация мероприятий. Подготовка к участию в олимпиадах
11 класс	Выбор и сдача ЕГЭ	Знакомство с профстандартами, будущей специальности	Встречи с родителями, посещение родительских собраний. Помощь в подготовке к ЕГЭ, определении стратегии поступления

В таблице перечислены олимпиады, в которых участвовали лицеисты в этом году.

Мероприятия	Организаторы	Статус	Участие
Открытая региональная межпредметная олимпиада «Золотая середина» с участием стран СНГ	Гимназия № 1 г. Новосибирска	Региональный	Поощрительное место в 7-м классе
Межрегиональной олимпиады школьников им. И. Я. Верченко (по математике и криптографии)	Академия ФСБ	Олимпиада 1-го уровня	Диплом 3-й степени в 9-м классе
Открытая Олимпиада университета Иннополис	АНО ВО «Университет Иннополис» Министерство образования и науки РТ (г. Казань)	Олимпиада 1-го уровня	Участие команды лицея в финале

Мероприятия	Организаторы	Статус	Участие
Олимпиады НТИ инженерные биологические системы	http://nti-contest.ru/about/	Олимпиада 3-го уровня	Участие
Олимпиада «Информационные технологии»	СПб ИТМО	Олимпиада 1 уровня	
Участие в профильных сменах Сириуса			Участие в смене по кибербезопасности

Выбор олимпиад основан на списке Российского совета олимпиад школьников (РСОШ) на текущий учебный год. На рисунке представлен пример подобной выборки для 2018–2019 учебного год.

Информатика

Наименование олимпиады	№ в перечне	Профиль	Уровень
Всесибирская олимпиада школьников	10	информатика	1
Московская олимпиада школьников	35	информатика	1
Олимпиада школьников «Ломоносов»	46	информатика	1
Открытая олимпиада Университета Иннополис	57	информатика	1
Открытая олимпиада школьников	58	информационные технологии	1
Открытая олимпиада по программированию	59	информатика	1
Олимпиада школьников по информатике и программированию	73	информатика	1
Олимпиада «Высшая проба»	23	информатика	2
Олимпиада СПбГУ	54	информатика	2
Открытая олимпиада по программированию «Когнитивные технологии»	60	информатика	2
Всероссийский конкурс научных работ школьников «Юниор»	9	инженерные науки	3
Олимпиада по дискретной математике и теоретической информатике	43	информатика	3
Олимпиада школьников «Надежда энергетики»	47	информатика	3

Как правило, для детей, готовых участвовать в этих олимпиадах, нужны информативные сообщения о статусе, времени регистрации, системе подготовки.

Но есть достаточно много ребят из непрофильных классов, для которых участие в олимпиадах не является выходом (им просто не хватает знаний). Они пытаются выбрать информатику в 9-м классе, готовятся и сдают ее хорошо, а потом задают вопрос: а что делать дальше?

У них совершенно другой вариант. Им нужны курсы из дополнительного образования, которые помогут им подтянуть программирование, и конкурсы, в которых они могут поучаствовать для опыта, проявляя свои сильные стороны.

Очень удачно, что в Летней школе юных программистов, которая уже в течение 15 лет проводится под эгидой ИСИ СО РАН, формируются не только спецкурсы, связанные с программированием, но и гуманитарные мастерские (например, связанные со СМИ). Это позволит направить туда не только ребят-программистов, но и тех, кто занимается какими-либо приложениями.

Примером возросшего интереса может явиться тот факт, что во второй образовательной акции по профориентации в сфере ИТ «Час кода» в этом году участвовало восемь миллионов школьников — на миллион больше, чем в прошлом.

Но какими бы разными ни были эти две группы учащихся, вопросы, которые появляются при обучении очень похожи:

– ребята, готовясь к олимпиадам, выезжая на профильные смены, часто фактически отсутствуют на уроках;

– родители, пытаясь помочь ребенку определиться как можно раньше, записывают его на всевозможные платные курсы, и в результате возникает перегрузка ребенка, он болеет и тоже отсутствует на занятиях;

– нередко с подачи родителей ребенок сам переставляет акценты на дополнительное образование.

Выход для школы в сложившейся ситуации — рассматривать дистанционное образование как один из видов образования наряду с классно-урочной системой.

Что касается выбора профессии, то с детьми надо обязательно говорить о перспективных направлениях и сферах, развивать навыки, такие как способность к самообучению, интерес к языку, возможность получить новые знания посредством дистанционных курсов.

Но вот наконец ребенок выбрал предметную область, ЕГЭ, даже ответил на вопрос о факультете. Но, к сожалению, ответить на вопрос о том, его ли это профессия, достаточно сложно. Если есть возможность, то надо пробовать себя в профессии. Такая проба может оценить, насколько это твое. Очень помогают в этом вопросе круглые столы, которые ФИТ НГУ проводит ежегодно в начале января. Встреча со студентами, магистрантами и выпускниками факультета, которые уже состоялись как профессионалы, помогает ориентироваться старшеклассникам.

И вопрос «Куда пойти учиться» по-прежнему остается самым актуальным в старших классах.

Список литературы

1. Программа «Цифровая экономика Российской Федерации». Утверждена распоряжением премьер-министра РФ от 28.07.2017 г.

2. Указ Президента РФ от 9.05.2017 «О стратегии развития информационного общества в Российской Федерации на 2017–2030 годы».

3. Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 г.

UDK 372.8

KEY TO THE FUTURE

M. G. Grigorenko

Teacher of Informatics of Municipal Budgetary Secondary Educational Establishment Lyceum № 130,
Named After Acad. M. Lavrentyev, 630090, Novosibirsk, Uchenikh st.,10, e-mail: margrig@yandex.ru

Abstract. This article focuses on the training of future IT professionals in school. It describes the road-map for career guidance of future students. There are specialized classes and a set of Olympiads for university admission in the Lyceum.

In the article we consider the forms of training that can be used in the process of choosing an IT specialty for specialized classes' students and general education classes' students.

Key words: secondary school, training of IT specialists, distance education, Olympiads

References

1. Programma «Tsifrovaya ekonomika Rossiiskoi Federatsii». Utverzhdena rasporyazheniem premer-ministra RF ot 28.07.2017 g.
2. Ukaz Prezidenta RF ot 09.05.2017 «O strategii razvitiya informatcionnogo obshchestva v Rossiiskoi Federatsii na 2017–20130 gg».
3. Federalnyi zakon «Ob obrazovanii v Rossiiskoi Federatsii» ot 29.12.2012 g.

О ТЕХНОЛОГИЯХ ПОЗНАНИЯ В ИНТЕГРИРОВАННЫХ ВЫЧИСЛИТЕЛЬНЫХ ОКРУЖЕНИЯХ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

В. П. Ильин

д-р физ.-мат. наук, гл. науч. сотрудник,
Институт вычислительной математики и математической геофизики СО РАН
630090, Новосибирск, пр. Акад. Лаврентьева, 6,
проф., Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 1, e-mail: islin@sscc.ru

Аннотация. Рассматриваются проблемы и технологии получения новых фундаментальных и прикладных знаний, а также принятия практических решений путем математического моделирования процессов и явлений на супер-ЭВМ пост-петафлопсного уровня. Исследуемые задачи относятся ко всем основным критическим направлениям научно-технического прогресса страны в наступившую эпоху 4-й индустриальной революции.

Согласно мировым тенденциям развития программного обеспечения, концептуальное решение заключается в создании интегрированных вычислительных окружений (ИВО), поддерживающих различные технологические стадии крупномасштабного машинного эксперимента: интерактивные описания прямых и обратных междисциплинарных постановок со сложными геометрическими и материальными характеристиками, высокопроизводительные дискретизационные, аппроксимационные, алгебраические и оптимизационные алгоритмы с масштабируемым распараллеливанием, эффективные дружественные интерфейсы с конечными пользователями разных профессий.

Реализация ИВО рассчитана на гибкое расширение состава бурно развивающихся новых моделей и алгоритмов, адаптацию к эволюции компьютерных платформ, высокопроизводительное использование внешних программных продуктов и согласованное участие различных групп разработчиков, что в совокупности должно обеспечить длительный жизненный цикл проекта, его высокую востребованность и экономическую успешность в производственной и социальной сферах. Одновременные требования эффективной наукоемкости и прагматической робастности ИВО определяют безусловную тесную кооперацию в данной разработке и академических институтов, и университетских компетенций, и профессиональных программистских производственных коллективов.

Исторически обусловленная глобализация создаваемой индустриальной среды невозможна без формирования средств искусственного интеллекта нового поколения, в отношении как внешних, так и внутренних интерфейсов ИВО. Здесь неразделимо сочетаются и базы знаний по математическим моделям и алгоритмам, и средства работы с большими данными, в том числе неточными, неполными, избыточными и противоречивыми, и способы автоматизации построения алгоритмов с их отображением на архитектуру ЭВМ, и семантическое моделиро-

вание, и реализации онтологических методик познавательного процесса, и когнитивные принципы принятия решений. Рассматриваемый круг проблем представляет собой проект масштаба «мегасайенс», аналогичный активно развиваемым национальным проектам в передовых странах (например, в Германии — SPPEXA—Strategic Priority Program on Exascale Computing с участниками из более чем 40 институтов). Создание высокоинтеллектуального ИВО нацелено на математизацию всех научных областей и отраслевых приложений, без которых невозможно построение цифровой экономики.

UDK 681.3.06

**ABOUT COGNITIVE TECHNOLOGIES
IN INTEGRATED COMPUTING ENVIRONMENTS FOR MATHEMATICAL MODELING**

V. P. Il'in

Principal Research Scientist, Dr. Sc.,
Institute of Computational Mathematics and Mathematical Geophysics of SB RAS,
630090, Novosibirsk, Acad. Lavrentiev pr., 6,
Full professor of Novosibirsk State University,
6300906, Novosibirsk, Pirogova St., 1, e-mail: islin@sscc.ru

Abstract. The problems and technologies of obtaining the new fundamental and applied knowledge, as well as making practical decisions with the help of mathematical modeling of processes and phenomena on the post-petaflops super-computer are considered. The tasks studied relate to all the main critical areas of the country's scientific and technological progress in the new era of the 4th industrial revolution.

According to the global trends in the software development, the conceptual solution is to create the integrated computing environments (ICE) supporting various technological stages of a large-scale computer experiment: interactive descriptions of direct and inverse interdisciplinary, problems with complex geometric and material characteristics, high-performance discretization and approximation, algebraic and optimization algorithms with scalable parallelization, effective, friendly interfaces for end-users of different professions.

The ICE implementation is designed for a flexible expansion of rapidly developing new models and algorithms, adaptation to the evolution of computer platforms, the high-performance use of external software products and coordinated participation of various teams of developers, which cooperation should ensure a long life cycle of the project, its high demand and economic success in industrial and social spheres. The simultaneous requirements temps of high-tech and pragmatic robustness for fICE determine unconditionally close collaboration in this development both of academic institutions, and university competencies, and professional programmer production teams.

The historically conditioned globalization of the created industrial environment is impossible without formation of artificial intelligence tools of the new generation, with respect to both external and internal interfaces of the ICE. There are inseparably combined knowledge bases on mathematical models and algorithms, tools for manipulation with big data, including inaccurate, incomplete, redundant and contradictory ones, and the ways to automate the construction of algorithms with their mapping onto a computer architecture, and semantic modeling, and the implementation of ontological methods of the cognitive process, and cognitive principles of decision making. The range of the problems under consideration is a “megascience”- scale project, similar to actively developing national projects in advanced countries (for example, in Germany — SPPEXA — Strategic Priority Program on Exascale Computing, with participants from more than 40 institutions). Creating a highly intelligent ICE is aimed at the mathematization of all scientific fields and industry applications, without which constructing the digital economy is impossible.

СИСТЕМЫ АВТОМАТИЗИРОВАННОЙ ОЦЕНКИ ЗАДАНИЙ ПО ПРОГРАММИРОВАНИЮ: ИСПОЛЬЗОВАНИЕ В УЧЕБНОМ ПРОЦЕССЕ

Д. В. Иртегов

доц., зав. лабораторией, Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 1, e-mail: fat@nsu.ru

Т. В. Нестеренко

науч. сотрудник, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, пр. Акад. Лаврентьева, 6, e-mail: nest@iis.nsk.su

Т. Г. Чурина

ст. науч. сотрудник, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, пр. Акад. Лаврентьева, 6, e-mail: tanch@iis.nsk.su

Аннотация. В работе описана практика использования автоматизированной системы для проверки заданий по программированию в учебном процессе. Приведен анализ опыта использования таких систем и обозначены перспективы их развития.

Ключевые слова: тестирование знаний, обучение программированию, олимпиады по программированию, системы тестирования, NSUts

Введение

В НГУ работы по автоматизированному тестированию заданий по программированию в той или иной форме ведутся с 1998 г. [1]. Было создано несколько поколений систем, начиная от скрипта объемом около 1000 строк на Perl и заканчивая используемой в настоящее время системой NSUts. Опыт их эксплуатации, а также знакомство с другими системами, создаваемыми в России и за рубежом, позволил уточнить и сформулировать требования к системам такого рода [2]. Опыт применения NSUts и ее предшественников в учебном процессе позволил сформулировать, чем требования к олимпиадным системам отличаются от требований к системам поддержки обучения [3].

Заметим, что многие разработчики относят исследования по системам автоматизированной оценки заданий по программированию либо к категории автоматизированного контроля качества программного обеспечения, либо к категории педагогического тестирования. Задача автоматизированного контроля навыков программирования требует иных подходов и иных инструментов.

Система NSUts

Система NSUts¹ поддерживает проведение соревнований как по правилам ACM-ICPC с тестированием программ на языках C, C++, Java, Pascal, Python, C# для Win32, так и по прави-

© Д. В. Иртегов, Т. В. Нестеренко, Т. Г. Чурина, 2019

¹ Система NSUts. URL: <https://olympic.nsu.ru/nsuts-new/login.cgi>.

лам Всероссийской олимпиады школьников. Язык программирования явно указывается участником при отправке задания, что позволяет проводить в рамках одного тура тестирование программ на нескольких диалектах одного языка, например, Visual C и Borland C.

Система предоставляет участникам веб-интерфейс для отправки задач, просмотра состояния их тестирования и рейтинга, а также для коммуникации с жюри. Задачи, очередь тестирования и рейтинг хранятся в реляционной СУБД. Сохраняется полная история всех попыток отправки решения, все исходные тексты и точные времена их отправки. Это необходимо как для обработки апелляций, требующих перетестирования, так и для расследования попыток мошенничества.

Тестирование решений производится на выделенных компьютерах, поскольку время исполнения задачи в многозадачной среде определяется сложными сценариями конкуренции за разные ресурсы: оперативную память, кэш процессора, время переключения контекста процессора и т. д. Исполнение решения на выделенных компьютерах сводит влияние такой конкуренции до минимума.

Олимпиады могут проводиться и по правилам, напоминающим Marathon Match на TopCoder, когда участникам предлагается одна игровая или исследовательская задача на пять часов работы. Первая номинация Открытой Всесибирской олимпиады по программированию им. И. В. Поттосина² проводится именно по таким правилам. Для проведения подобного вида соревнований в системе NSUts выполняется специальная настройка. Например, в заочный этап Всесибирской олимпиады школьников по информатике³ была включена задача по искусственному интеллекту, разработанная сотрудниками Академии искусственного интеллекта для школьников при поддержке Сбербанка⁴. Для решения этой задачи был создан специализированный тур в системе тестирования NSUts, а для ее проверки реализован дополнительный модуль.

Требования, предъявляемые к автоматизированным системам тестирования знаний, описаны в работах [3, 4], архитектура системы — в работе [1].

Опыт использования системы NSUts

Важной частью использования системы NSUts является ее применение в промежуточном контроле знаний и навыков по информатике и программированию студентов высших и средне-технических учебных заведений, организации уроков по информатике и кружков по программированию для школьников.

Начиная с 2012 года, система NSUts используется в НГУ для организации проверки заданий студентов первого курса ФИТ и ММФ по дисциплине «Программирование». Основанием для этого послужили следующие ее свойства:

– в NSUts принята двухуровневая система организации данных. На верхнем уровне — олимпиады, внутри олимпиад — туры. Каждый тур состоит из набора задач. В терминах учебного процесса можно рассматривать олимпиады как годовой или семестровый учебный практический курс, а каждый тур — недельное задание, состоящее из задач, объединенных темой, которая разбирается на семинарских занятиях. Список участников такой «олимпиады» ограничивается студентами одной группы с одним преподавателем, что позволяет обеспечивать некоторую конфиденциальность информации;

² Сайт открытой Всесибирской олимпиады по программированию им. И. В. Поттосина. URL: <https://olympic.nsu.ru/widesiberia/2018/news>.

³ Сайт Всесибирской открытой олимпиады школьников по информатике 2018–2019 гг. URL: <https://olympic.nsu.ru/school/siberia/2018-2019/news>.

⁴ Академия ИИ для школьников при поддержке Сбербанка. URL: <https://contest.ai-academy.ru/competition>.

– круглосуточная работа системы позволяет сдавать задачи в любое время, что поддерживает самостоятельную работу студента;

– решения студентов проверяются на заранее подготовленном и загруженном в систему наборе тестов. Это значительно экономит время преподавателя при проверке заданий, позволяя больше внимания уделять разбору алгоритмов, правилам оформления кода, выявлению понимания студентом того, что он закодировал. Преподаватель также может проверять задания студентов удаленно, помогая им даже во внеурочное время. Для этого интерфейс системы NSUts предоставляет соответствующие средства. Однако проверка заданий посредством автоматизированной системы не должна исключать личного общения преподавателя и студента, во время которого первый может, например, не зачесть решение, даже если оно прошло все тесты и получило вердикт «принято»;

– для стимулирования регулярной работы студентов преподаватель может ввести систему баллов, которые учитываются при выставлении оценок. Важную роль здесь играет время сдачи задания, которое хранится в системе NSUts для каждой посылки. Стремление сдать все задачи в кратчайшее время порождает здоровую конкуренцию среди одноклассников, что, несомненно, идет на пользу обучению;

– заранее озвученный набор правил оценивания работы и выставления зачета способствует формализации отношений между преподавателем и студентом, практически исключает необъективность и предвзятость.

В первом семестре первокурсникам ФИТа требуется выполнить 15 заданий — по одному в неделю. Каждое задание содержит от трех до восьми задач. Всего 60 задач разного уровня сложности. Если задача сдается в систему в течение первой недели со дня выдачи задания, то за нее начисляется 10 баллов, если позже — то баллы понижаются. Если же студент сдает задачу в первый день после соответствующего семинара, то он получает еще один бонусный балл.

Для получения отличной оценки необходимо набрать не менее 75 процентов от базового количества баллов, т. е. не менее 450 баллов. Среднее количество баллов, которое набирали студенты, очень близко к этой цифре, а в половине случаев даже ее превосходит. Максимальное же количество баллов значительно превосходит 600, что говорит о том, что в каждой группе были студенты, которые стремились все сдать в первый день и получить за каждую задачу по 11 баллов.

Пути развития системы

Поскольку система NSUts изначально проектировалась для проведения олимпиад по программированию различного уровня, то для использования ее в учебных занятиях в ней не было предусмотрено ряда функций и свойств. Отметим некоторые из них.

1. Решение участника должно быть записано в одном файле, поэтому большой проект, состоящий из нескольких модулей, в данной системе проверить нельзя.

2. Для усвоения некоторых структур данных, например, списков, необходимо научиться эти структуры преобразовывать: написать функцию, которой на вход подается уже готовая построенная структура, а на выходе — результат преобразования. В системе NSUts, как и в любой другой системе проверки решений, этого сделать нельзя, так как входные и выходные данные для решения должны быть записаны в файлы. Поэтому студенту кроме самого преобразования необходимо еще и структуру построить, что требует дополнительных усилий. Если не проверять код студента, то может оказаться, что его решение выдало правильный ответ, но задача решена не та, которая была поставлена. Рассмотрим, например, задачу удаления повторов в односвязном списке. На вход подается последовательность чисел, которые должны храниться в списке. Такой список нужно построить, а затем из него удалить рядом стоящие повторяющиеся элементы, по-

сле чего полученный список распечатать, чтобы проверить правильность. Т. е. на входе — файл с последовательностью целых чисел, на выходе — тоже файл с последовательностью целых чисел. Если бы это была задача на олимпиаду, то никто бы не стал строить список, а анализировал бы входную последовательность и выводил в выходной файл числа без повторений. Такое решение получило бы вердикт «принято». Но обучению работе с динамическими списками оно никак бы не способствовало. В данном случае большую роль играет преподаватель, которые не только смотрит на вердикт системы, но еще и проверяет код программы студента.

3. В системе рейтинг участников строится в туре. Если хочется построить общий рейтинг по всем турам, то это можно сделать только внешними средствами. Автоматизация процесса построения такого рейтинга была бы желательна.

4. Как правило, в олимпиадах по программированию доступ к набору тестов имеет только жюри, участникам тесты недоступны. Хотелось бы, чтобы в процессе обучения преподаватель мог давать возможность смотреть тесты студентам в системе. Это часто помогает найти ошибку в программе. В настоящее время эта проблема решается только посредством почты.

5. За многие годы накопилось большое количество задач разного уровня сложности и разной тематики. Все эти задачи хранятся в своих турах, найти нужную очень сложно. При проектировании системы предполагалось сделать банк задач без привязки к конкретной олимпиаде или туру. Этот пункт стоит в планах развития системы.

Заключение

Многолетний опыт использования нескольких поколений систем автоматической оценки заданий по программированию [3] позволил разработать мощное и гибкое решение, применимое для широкого спектра мероприятий: проведения олимпиад, практикумов по программированию, тренировок олимпиадных команд и др.

Также были выявлены возможные направления дальнейшего улучшения системы. Самым главным является адаптация системы к требованиям учебного процесса. Например, модель олимпиад и туров, т. е. краткосрочных мероприятий, не очень хорошо соответствует процессу обучения. В ходе соревнований участнику выдают набор заданий, которые он может решать в произвольном порядке, лишь бы уложиться в общий лимит времени. При обучении полезнее выдавать студенту последовательности заданий, например, в соответствии с темами лекций или по нарастанию сложности. Также система могла бы устанавливать индивидуальные сроки выполнения заданий и следить за их соблюдением. Это позволило бы реализовать поддержку для адаптивных стратегий обучения.

Еще одно возможное направление развития — это добавление других стратегий проверки заданий, кроме прогона, на заранее заданных наборах тестов. Например, для начального обучения языкам программирования, когда учащиеся еще не знакомы с операциями ввода-вывода, полезно было бы иметь возможность проверять структуры данных, создаваемые программой в памяти. Это позволило бы поставить задачу «построить в памяти связный список из 5 элементов», а затем проверить, действительно ли этот список связный и сколько элементов он содержит. Данный подход требует дополнительных исследований в области реализации изолирующих сред для тестируемых программ.

Список литературы

1. *Иртегов Д. В., Нестеренко Т. В., Чурина Т. Г.* Разработка систем автоматизированной оценки заданий по программированию // Журнал «Системная информатика». 2017. № 11. С. 91–116.
2. *Боженкова Е. Н., Иртегов Д. В., Киров А. В., Нестеренко Т. В., Чурина Т. Г.* Автоматизированная система тестирования NSUts: требования и разработка прототипа // Вестн. НГУ. Серия : Информ. технологии. 2010. Т. 8, Вып. № 4. С. 46–53.
3. *Чурина Т. Г., Иртегов Д. В.* Требования к автоматической системе тестирования знаний : труды VI Междунар. конф. «Интеллектуальные технологии в образовании, экономике и управлении». Воронеж, 2009. С. 309–317.
4. *Боженкова Е. Н., Воронков А. Д., Иртегов Д. В., Коньшева Е. Н., Черненко С. А., Чурина Т. Г.* Модель разграничения прав доступа в системе автоматизированной проверки корректности программных приложений // Вестн. НГУ. Серия : Информ. технологии. 2011. Т. 9, Вып. № 4. С. 79–85.

UDK 004.4, 37.018

SYSTEMS FOR AUTOMATED EVALUATION OF PROGRAMMING TASKS: USE IN THE EDUCATIONAL PROCESS

D. V. Irtegov

Assistant professor, Novosibirsk State University,
6300906, Novosibirsk, Russia, Pirogova St., 2, e-mail: dmitry.irtegov@gmail.com

T. V. Nesterenko

Researcher, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentiev pr., 6, e-mail: nest@iis.nsk.su

T. G. Churina

Senior Researcher, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentiev pr., 6, e-mail: tanch@iis.nsk.su

Annotation. The paper describes the practice of using an automated system to test programming tasks in the educational process. The analysis of the experience of using such systems is given and the prospects for their development are indicated

Keywords: knowledge testing, programming training, programming olympiads, testing systems, NSUts

References

1. *Irtegov D. V., Nesterenko T. V., Churina T. G.* Development of Automated Evaluation Systems for Programming Tasks // System Informatics Journal. 2017. № 11. P. 91–116. (in Russian)
2. *Bozhenkova E. N., Irtegov D. V., Kirov A. V., Nesterenko T. V., Churina T. G.* NSUts Automated Testing System: Prototype Requirements and Development // Vestn. NSU. Series : Inform. Technologies. 2010. Vol. 8, № 4. P. 46–53. (in Russian)
3. *Churina T. G., Irtegov D. V.* Requirements for an automated knowledge testing system : Proceedings of the VI International Conference “Intellectual Technologies in Education, Economics and Management”. Voronezh, 2009. P. 309–317. (in Russian)
4. *Bozhenkova E. N., Voronkov A. D., Irtegov D. V., Konysheva E. N., Chernenok S. A., Churina T. G.* Model of access rights delimitation in the system of automated verification of software applications correctness // Vestn. NSU. Series : Inform. Technologies. 2011. Vol. 9, № 4. P. 79–85. (in Russian)

ОБРАЗОВАТЕЛЬНОЕ ЗНАЧЕНИЕ ИСТОРИИ ИНФОРМАТИКИ: СОСТОЯНИЕ И ПЕРСПЕКТИВЫ

И. А. Крайнева

канд. ист. наук, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, пр. Акад. Лаврентьева, 6, e-mail: cora@iis.nsk.su

Ю. А. Ревич

Независимый исследователь, журналист, издатель
e-mail: revich@lib.ru

Аннотация. В статье излагаются подходы к преподаванию истории отечественной информатики. Приведена краткая библиография вопроса, результаты опроса сотрудников институтов и университетов, которые участвуют в просветительской работе по истории информатики.

Ключевые слова: академик А. П. Ершов, история науки и техники СССР, образование, вычислительная техника, ЭВМ, история программирования в СССР

Введение

Академик А. П. Ершов (1931–1988) — выдающийся советский ученый, создатель сибирской школы информатики, внес большой вклад и в изучение истории информатики и программирования. Он понимал важность новой науки, стремился сделать достоянием потомков ее основные проблемы и достижения, полученные усилиями его поколения. Многие исследования в области истории информатики были опубликованы при жизни Андрея Петровича, многое стало доступно благодаря электронному архиву академика А. П. Ершова в Интернете (<http://ershov.iis.nsk.su>). Архив охватывает период с середины 40-х и до конца 90-х гг. прошлого века, поскольку последние дополнения относятся уже к периоду существования Института систем информатики СО РАН. Архив А. П. Ершова — уникальное собрание свидетельств о развитии информатики в СССР. Он играет большую роль в продвижении истории этого направления практически на всех уровнях научно-образовательной вертикали.

Рекомендации SoRuCom

В решениях конференции «Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы» (SoRuCom) с 2006 г. регулярно поднимается вопрос о просветительской деятельности в области истории отечественной ВТ и программирования. Пару лет назад мы утверждали, что «время идет, и постепенно стирается память о страницах

истории отечественной IT-отрасли, вытесненной современными реалиями. Кроме того, пора сформировать программу исследований, которая бы помогла систематизировать наше представление о прошлом, изложить альтернативные точки зрения на некоторые вопросы». Мы предложили и возможные направления такой программы.

1. Государственная политика в области создания отечественных машин, включая переход к ЕС ЭВМ. Роль АН СССР, ведущих профильных министерств и ведомств.

2. Элементная база — создание центра в Зеленограде, основные тенденции и проблемы развития микроэлектроники в СССР.

3. Основные центры, выпускавшие ЭВМ в СССР и союзных республиках. Типы ЭВМ. Ведущие конструкторы и их школы.

4. Направления использования отечественных ЭВМ. Решение научных и прикладных задач, роль ЭВМ в освоении космоса, создании ядерного щита, в оборонной отрасли в целом. Проекты управления народным хозяйством, создание АСУ и АСУТП.

5. Ведущие советские школы программирования и их место в мировой науке.

6. Подготовка кадров инженеров, вычислителей, программистов.

7. Образовательные программы в области информатики.

Мы писали также, что «эта сложная и объемная программа должна объединить профильных специалистов и историков, поскольку историей вычислительной техники и программирования неспециалисту заниматься сложно. Однако историк владеет методикой исследования, специалист — фактической и содержательной стороной дела. Для того чтобы такое объединение состоялось, совершенно не требуется принимать глобальные программы на правительственном уровне и выделять огромные средства из бюджета. Достаточно мотивировать профессиональное сообщество: создать соответствующие направления при исторических факультетах университетов, объявить конкурсы и назначить премии, поддержать существующие конференции и организовать новые и т. д.» [1]. Эта программа отчасти выполняется, но она пока не приобрела большого влияния.

Краткая библиография

Члены сообщества SoRuCom вносят посильный вклад в изучение истории отечественной ВТ и программирования. В последнее время отдельные разрозненные исследования, в отличие от начального этапа их становления в 1990-х — начале 2000 гг., аккумулированы в солидных изданиях, сопровождаются аналитическими выкладками.

Монография В. В. Липаева хотя и посвящена истории отечественной программной инженерии, содержит многочисленные экскурсы в историю создания конкретных ЭВМ и программирования. Книга производит двойственное впечатление. С одной стороны, автор, который много лет работал в советском ВПК, раскрыл нам многие неизвестные страницы истории этой гигантской корпорации. Теперь мы твердо уяснили, что ВПК работал, как правило, по принципу черной дыры [2. С. 25, 135]. Правда, известны случаи, когда разработки ВПК выходили за его пределы. Так, усилиями академика М. В. Келдыша группа разработчиков ОС «Диспак» В. Ф. Тюрина была переведена в ИПМ из НИИ-1011 (ВНИИТФ им. акад. Е. И. Забабахина) [3]. Другая особенность монографии Липаева — отсутствие твердой авторской позиции в некоторых вопросах. Это касается, в частности, проблемы копирования зарубежных ЭВМ с конца 1960-х [2. С. 13, 28–29]. Но это несколько не умаляет других достоинств книги, ее богатой фактологии.

Другая монография, которую хотелось бы упомянуть как содержащую не только последовательное изложение событий, но и честный анализ и критику отечественных реалий

в области создания ВТ, микроэлектроники, личностных характеристик отдельных, в том числе высокопоставленных участников событий — И. С. Брука, Б. М. Наумова, министров, директоров заводов, — это книга Б. Фельдмана [11]. Автор противопоставил отталкивающему термину «внедрение» понятие «массового производства», которое выявляло нестыковки между победными реляциями НИИ, в том числе академических, и заводским производством.

Особо следует отметить масштабную работу, которую провели отечественные инженеры, программисты и историки науки по созданию энциклопедий, вобравших в себя историю отечественной вычислительной техники и программирования гражданского и широкого оборонного применения, персоналии. Огромным упущением в этом труде явился практический отказ украинских коллег от участия в его составлении. Энциклопедии содержат обширную библиографию публикаций по теме издания [4].

Ю. Ревич выпустил два сборника (один из них создавался в сотрудничестве с Б. Н. Машиновским из Киева), в которых собраны очерки, освещающие основные страницы истории отечественной вычислительной техники и информационных технологий в целом, — как наиболее разработанные направления (биографии С. А. Лебедева, В. М. Глушкова и др.), так и относительно малоизвестные (некоторые моменты истории Центра микроэлектроники в Зеленограде, создания советской ПРО, история уникальной системы мобильной связи «Алтай» и др.) [5, 6].

Продолжают активно работать и развиваться информационные ресурсы Виртуального компьютерного музея. К сожалению, поисковая система ВМК оставляет желать лучшего: на сайте трудно находить нужную статью. ВМК издал уже пять выпусков книг в серии «Страницы истории отечественных ВТ», последняя вышла в 2019 г. Э. Пройдаков записал серию интервью, которые, несомненно, дополняют наши представления об отечественных исторических реалиях, описанных их участниками (Б. А. Бабаяном, Е. П. Велиховым, Е. Л. Дшхуняном, В. П. Иванниковым, А. Н. Томилиным и другими специалистами).

Пополняется список книг о персоналиях. Одной из последних вышла книга о жизни и деятельности Льва Николаевича Королева (1926–2016) [7]. Концептуальным аспектам деятельности А. А. Ляпунова и А. П. Ершова посвящена в том числе монография И. Крайневой [8]. Но биографические исследования о некоторых известных отечественных ученых и инженерах еще не проведены. Например, до сих пор не удостоился книги глава московской школы программирования ИПМ М. Р. Шура-Бура (1918–2008).

Стоит указать и на две важные темы, практически не получившие освещения в эти годы, за редким исключением [9]. Одна из них — народнохозяйственное значение использования ЭВМ в Советском союзе. О создании отраслевых АСУ, внедрении отдельных АСУТП мало исследований, содержащих анализ роли этих начинаний, их эффективности, если она имела место. Хотелось бы понять, не играли ли АСУ роль проекта, навязанного без того не слишком благополучной советской экономике?

Вторая малоизученная тема — история советских исследований в области искусственного интеллекта. Немаловажно, что еще в советские времена эти исследования отечественных ученых весьма высоко котируются в мире. Многие современные разработки мирового уровня, которыми может гордиться Россия в области программной индустрии, выросли в конечном итоге именно из них. Имеются в виду лингвистические алгоритмы поиска, задействованные в Яндексе (наша страна единственная в мире, кроме Китая и США, имеющая собственную поисковую систему). Нужно упомянуть программы распознавания фирмы АBBYY (Fine Reader) или Cognitive Technologies (2017 — награда разработчику систем искусственного интеллекта для беспилотных транспортных средств мирового уровня), эвристические алгоритмы борьбы с вирусами Лаборатории Касперского и другие продукты.

В целом имеющаяся литература по истории ВТ и ее применений, микроэлектроники, программирования достаточно разнообразна и многопрофильна. Она служит стартовой основой для тех, кто интересуется этой историей, может самостоятельно к ней обратиться.

Что показал блиц-опрос?

В преддверие подготовки данной статьи мы провели блиц-опрос среди некоторых наших наиболее активных в просветительской деятельности коллег. На 11 отправленных анкет получено 10 ответов на следующие два блока вопросов:

1. Ваш вклад в образовательный процесс в области истории информатики:

- вуз, институт, музей;
- курсы;
- отдельные лекции;
- статьи;
- книги;
- другие формы.

2. Известно ли вам о продвижении истории информатики в вузе, где вы, возможно, преподаете?

- вуз;
- кафедра;
- научное направление;
- музей;
- отдельное мероприятие;
- другие формы.

Ответы были получены преимущественно на первый блок вопросов, т. е. можно предположить, что систематически продвижением истории информатики занимаются мало (см. таблицу).

Итоги блиц-опроса, февраль 2019 г.

№	Участники блиц-опроса	Место работы	Курсы	Книги	Отд. лекции	Статьи	Конф.
1	Терехов А. Н.	СПбГУ	+ , видео		+	+	+
2	Шилов В. В.	ВШЭ	+	+	+	+	+
3	Тумбинская М. В.	КНИТУ-КАИ			+	+	+
4	Гейн А. Г.	УФУ		+(учебник)	+	+	+
5	Мартыненко Б. К.	СПИИРАН		+		+	+
6	Китов В. А.	РЭУ		+	+	+	+
7	Бадрутдинова М. Ш.	Музей	+	+	+	+	+
8	Городняя Л. В.	ИСИ-НГУ			+	+	+
9	Крайнева И. А.	ИСИ		+	+	+	+
10	Панкрашкина Н. Г.	ННГУ, музей		+		+	+

Исключение составляет обязательный курс по истории ЭВМ и программирования для магистров 1-го курса на математико-механическом факультете СПбГУ. Об этом сообщил заве-

дующий кафедрой системного программирования СПбГУ профессор А. Н. Терехов. Он прокомментировал ситуацию следующим образом: «К сожалению, посещаемость низкая, мне приходится буквально “вбивать” в их головы хотя бы основные имена советских ученых. Возможно, это связано с тем, что из 30 магистров первого курса только 3–4 окончили бакалавриат нашего мат-меха, а остальные буквально ничего не знают по архитектуре ЭВМ, а рассказывать историю ЭВМ людям, которые не знают, что такое “водопровод” или “расслоенная память” — очень трудно, им не оценить величия С. А. Лебедева».

Еще один цикл лекций для студентов 1-го курса Института прикладной математики и компьютерных наук ТГУ читает канд. физ.-мат. наук, доц. Б. А. Гладких. По результатам курса студенты сдают тестовый зачет. Как сообщил руководитель образовательной программы «Программная инженерия» ВШЭ канд. техн. наук В. В. Шилов, он готов читать курс из 12–14 лекций, но он не востребован. Отдельные лекции Шилов читает для студентов МАИ, они готовят рефераты, работают в РГБ, где снимают копии статей по информатике по заданию преподавателя. Тем самым студенты получают представление о библиотечных реалиях.

Л. В. Городняя (Новосибирск, ММФ НГУ) привлекает студентов к историческим изысканиям, подготовке соответствующих рефератов, она ставит задачу перед учащимися верифицировать сведения, почерпнутые в Сети. Она прививает студентам мысль, что Википедия не является безупречным источником информации, а лишь своего рода навигатором, от которого можно отталкиваться, не доверяя ему безоговорочно. Тем самым она ставит во главу образовательного процесса способность к критическому мышлению.

Большая научно-просветительская работа проводится Музеем «История вычислительной техники в Казани» (создан в 2003 г. как структурное подразделение АО «ICL-КПО ВС», директор М. Ш. Бадрутдинова). Здесь проводят экскурсии для студентов, школьников, специалистов, вновь поступающих на фирму сотрудников и гостей компании. (История создания отрасли ВТ в Казани; История Казанского завода ЭВМ; биография Б. И. Рамеева, С. А. Лебедева, А. И. Китова, Н. П. Брусенцова, Н. В. Васильева, создателя «Воображаемой логики», и др.)

Сотрудники музея регулярно участвуют в конференции SoRuCom, на ежегодных декабрьских конференциях в Политехническом музее (Москва), посвященных Дню Российской информатики и других. О музее снят фильм [10]. В настоящее время подготовлена к изданию книга об основателе первого завода пишущих машин в СССР — изобретателе пишущей машины на арабице и разработчике пишущей машины «Яналиф» на латинице — Аскаре Алиевиче Шейх-Али (1885–1968). Работает со студенческой аудиторией Н. Г. Панкрашкина, сотрудник Музея истории информатики Университета Лобачевского (Нижний Новгород).

В основном же специалисты в области ВТ и программирования, историки науки читают лекции периодически, выступают на семинарах, где присутствует студенческая аудитория. Для школьников проводятся лекции в рамках празднования Дней науки, во время работы Летней школы юных программистов в Новосибирске. Политехнический музей регулярно проводит научную конференцию 4 декабря, в День российской информатики.

Заключение

Говоря об образовательном значении истории информатики в России, мы не можем рассматривать его вне значения изучения отечественной истории в целом и особенно развития такого ее важного раздела, как краеведение. Недавняя акция по «переименованию» аэропортов, возможно, и всколыхнула национальное самосознание, но результат этой акции выявится, скорее, позднее. Ни для кого не секрет, что следование традициям формируется веками, поколениями, системной работой в этих направлениях. Не хотелось бы идти по ложному пути.

В этом контексте история отечественной информатики является одним из важных звеньев истории научно-технического развития СССР. Появление ЭВМ и программирования в СССР были настоящим прорывом, феноменами, которые привели к формированию нового поля науки, новых отраслей промышленности, новых профессий. Коллапс этих отраслей и направлений неверно связывать с переориентацией научно-технической политики в сторону «копирования прототипов». Причины гораздо глубже, и кроются они в системном кризисе всей отечественной экономики и политике XX века. Эта проблема требует специального изучения, как и все вышеперечисленные.

Отметим также, что почти все упомянутые исследования и программы по истории информатики в СССР сегодня, так или иначе, основаны на деятельности энтузиастов и очевидцев: тех, кто в свое время оказался вовлечен в отдельные направления происходивших событий. По естественным причинам таких людей становится все меньше, и с ними уходит главное: понимание того, что в основе многих ключевых свершений советской науки и техники — в первую очередь космонавтики и атомного проекта — немаловажное значение имело развитие отечественной информационной отрасли.

Список литературы

1. Ревич Ю., Крайнева И. История отечественной IT-отрасли: надо ли ее изучать? URL: <http://d-russia.ru/istoriya-otechestvennoj-it-otrasli-nado-li-ee-izuchat.html>.
2. Лунаев В. В. Очерки истории отечественной программной инженерии 1940-е–1980-е годы. М. : СИНТЕГ, 2012. С. 13. 25, 28–29, 135.
3. Тюрин В. Ф., Зельдинова С. А., Крайнева И. А. Операционная система Диспак. URL: <http://www.computer-museum.ru/articles/operatsionnye-sistemy/789/>.
4. Отечественная электронно-вычислительная техника. Биографическая энциклопедия. М. : ЗАО Издательский дом «Столичная энциклопедия», 2014. 400 с.
5. История отечественной вычислительной техники. Изд. 2-е, испр., доп. М. : ООО Издательский дом «Столичная энциклопедия», 2017. 680 с.
6. Ревич Ю., Малиновский Б. Информационные технологии в СССР. Создатели советской вычислительной техники. СПб. : БХВ-Петербург, 2014. 336 с.
7. История информационных технологий в СССР. Знаменитые проекты: компьютеры, связь, микроэлектроника / под ред. Ю. В. Ревича. М. : Книма, 2016. 416 с.
8. Лев Николаевич Королев: биография, воспоминания, документы / сост. и ред. В. К. Власов, Р. Л. Смелянский, А. Н. Томилин. М. : МАКС Пресс, 2016. 272 с.
9. Крайнева И. А. Научное наследие советских ученых в Электронных архивах СО РАН: Мастер. Проповедник. Лидер / ред. Т. М. Бульонкова, Н. А. Черемных. Новосибирск : ИПЦ НГУ, 2018. 386 с.
10. Пармонов В. Н. Внедрение АСУ в народном хозяйстве СССР в 1960-1970-х годах: замыслы и реализация // Труды SoRuCom-2014. С. 275–280.
11. Фильм «История вычислительной техники в Казани». URL: <https://retrotexnika.ru/kolection/muzej-istorii-kazanskogo-zavoda-evm.html>.
12. Фельдман Б. Я. От калькулятора к суперкомпьютеру. Записки разработчика. М. : Изд-во «РТСофт», 2014. 400 с.

**EDUCATIONAL VALUE OF COMPUTER
SCIENCE HISTORY: STATE-OF-THE-ART AND PROSPECTS**

I. A. Krayneva

Ph.D., Researcher, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentiev pr., 6, e-mail: cora@iis.nsk.su

Yu. A. Revich

Independent researcher, journalist, publisher,
Moscow
e-mail: revich@lib.ru

Annotation: The article outlines approaches to teaching the history of Russian informatics. A brief bibliography of the problem is given. The results of a survey of employees of institutes and universities who are involved in educational process on the history of computer science are given.

Keywords: academician A.P. Ershov, history of science and technology of the USSR, education, computer technology, computers, history of programming in the USSR

References

1. *Revich Iu., Krayneva I.* Istoriya otechestvennoi IT–otrasli: nado li ee izuchat? [The history of domestic IT: do we need to study it?]. URL: <http://d-russia.ru/istoriya-otechestvennoj-it-otrasli-nado-li-ee-izuchat.html>.
2. *Lipaev V. V.* Ocherki istorii otechestvennoi programmnoi inzhenerii 1940-e–1980-e gody. [Essays on the history of domestic software engineering 1940s–1980s] M. : SINTEG, 2012. P. 13. 25, 28–29, 135.
3. *Tyurin V. F., Zeldinova S. A., Krayneva I. A.* Operatsionnaya sistema Dispak [DISPAC operating system]. URL: <http://www.computer-museum.ru/articles/operatsionnye-sistemy/789/>.
4. Otechestvennaya elektronno-vychislitel'naya tekhnika. Biograficheskaya entsiklopediya [Domestic electronic equipment. Biographical encyclopedia] M. : ZAO Izdatelskii dom «Stolichnaya entsiklopediya», 2014. 400 p.
5. Istoriya otechestvennoi vychislitel'noi tekhniki [The history of domestic computing]. Izd. 2-e, ispr., dop. M. : OOO Izdatelskii dom «Stolichnaya entsiklopediya», 2017. 680 p.
6. *Revich Iu., Malinovskii B.* Informatcionnye tekhnologii v SSSR. Sozdатели sovetskoi vychislitel'noi tekhniki [Information technologies in the USSR. The creators of the Soviet computing technology]. SPb. : BKHV-Peterburg, 2014. 336 p.
7. Istoriya informatcionnykh tekhnologii v SSSR. Znamenitye proekty: kompiutery, sviaz, mikroelektronika [The history of information technologies in the USSR. Famous projects: computers, communications, microelectronics] / Ed. by Iu. V. Revich. M.: Knima, 2016. 416 p.
8. Lev Nicolaevich Korolev: Biografiya, vospominaniya, dokumenty` [Lev Nicolaevich Korolev: biography, memories, documents] / Comp. and eds. Vlasov V. K., Smelianskii` R. L., Tomilin A. N. M. : MAKS Press, 2016. 272 p.
9. *Krayneva I. A.* Nauchnoe nasledie sovetskikh uchenykh v Elektronnykh arhivakh SO RAN: Master. Propovednik. Leeder [Scientific heritage of Soviet scientists in the Electronic Archives of the SB RAS: Master. Preacher. Leader] / Eds. T. M. Bulyonkova, N. A. Cheremnykh. Novosibirsk : RINTC NGU, 2018. 386 c.
10. *Paramonov V. N.* Vnedrenie ASU v narodnom hozyaistve SSSR v 1960–1970-kh godakh: zamysly i realizatsiya [The introduction of automated control systems in the national economy of the USSR in the 1960-1970s: plans and implementation] // Trudy SoRuCom-2014. P. 275–280.
11. Film «Istoriya vychislitel'noi tekhniki v Kazani» [The film “The history of computing technology in Kazan”]. URL : <https://retrotexnika.ru/kolekcion/muzej-istorii-kazanskogo-zavoda-evm.html>
12. *Feldman B. Ya.* Ot kalkuliatora k superkompiuteru. Zapiski razrabotchika [From calculator to super-computer. Developer's Notes]. M. : Izd-vo «RTSoft », 2014. 400 p.

РАЗВИТИЕ ПРОФЕССИОНАЛЬНОЙ ОРИЕНТАЦИИ СТУДЕНТОВ ИТ-СПЕЦИАЛЬНОСТЕЙ

М. М. Лаврентьев

д-р физ.-мат. наук, Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 1, e-mail: mmlavr@nsu.ru

Т. С. Васючкова

Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 1

Л. В. Городняя

канд. физ.-мат. наук, ст. науч. сотрудник,
Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6,
доц., Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 2, e-mail: lidvas@gmail.com

М. А. Держо

доц., Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 1, e-mail: m_derjo@mail.ru

Н. А. Иванчева

доц., Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 1, e-mail: iva@ci.nsu.ru

И. В. Белого

науч. сотрудник, Институт автоматизации и электрометрии СО РАН (ИАиЭ СО РАН), СофтЛаб-НСК,
630090, Новосибирск, пр. Акад. Коптюга, д. 1, e-mail: misc@softlab-nsk.com

В. С. Бартош

инженер-программист, Институт автоматизации и электрометрии СО РАН (ИАиЭ СО РАН), СофтЛаб-НСК,
630090, Новосибирск, пр. Акад. Коптюга, д. 1, e-mail: vas@sl.iae.nsk.su

Аннотация. В статье рассматриваются некоторые аспекты профессиональной подготовки студентов ИТ-специальности ФИТ НГУ.

Ключевые слова: ИТ-специалист, профессиональный стандарт, образовательные среды Интернет, ФИТ НГУ

Ключевая роль в профессиональной подготовке студентов IT-специальности при обучении как на этапе бакалавриата, так и в магистратуре ФИТ НГУ принадлежит практикам — учебной, производственной и преддипломной. Местами прохождения практики студентов ФИТ НГУ являются IT-компании, научно-исследовательские, производственные и другие организации. На практике студенты работают над темами, связанными с реальными проектами или представляющими часть реальных проектов, и обеспечиваются поддержкой научного руководителя, соруководителя и консультанта. В период практики студенты выполняют и выпускную квалификационную работу бакалавра (ВКР).

Выполнение конкретной задачи в период практики и при разработке ВКР погружает студента в реальную среду IT-индустрии или научно-исследовательской работы. Важно также, чтобы студент представлял все разнообразие видов своей будущей профессии и присущих ей IT-специальностей. Это знание существующих профессиональных IT-стандартов, функциональных обязанностей каждой IT-специальности, понимание требуемого уровня образования и также необходимых личностных качеств. Поэтому кроме освоения дисциплин и прохождения практики в рамках учебного плана студентам для расширения профессионального кругозора и лучшей профессиональной ориентации рекомендуется проявлять некоторые дополнительные активности. Это более пристальное изучение существующих профессиональных стандартов в области информационных технологий и участие в специальных мероприятиях, организуемых IT-компаниями, а также исследование и активное использование образовательных ресурсов Интернет. Для того чтобы направить их к этому, в рамках дисциплины «Введение в профессию» на кафедре Систем информатики ФИТ студентам предлагается выполнить ряд специальных заданий.

Задание 1. Подготовить эссе о том, почему после окончания школы студентом был сделан выбор профессии IT-специалиста, почему для получения IT-образования был выбран конкретный ВУЗ и факультет. Насколько правильность этого выбора подтвердили прошедшие годы обучения. Какой видится студенту его будущая работа после получения диплома.

Задание 2. Познакомиться с профессиональными стандартами, размещенными на сайте АПКИТ, получить подробное представление о цели и содержании той или иной профессиональной деятельности в сфере IT, о наборе трудовых функций, требуемом уровне образования, личностных качествах и т. д. Выбрать три — четыре наиболее привлекательных вида деятельности, рассказать о них и обосновать свой выбор.

Задание 3. Подытожить личный опыт участия в конференциях, олимпиадах, профессиональных мероприятиях и встречах, предлагаемых IT-компаниями, проанализировать и оценить, насколько этот опыт оказался интересен и полезен для профессионального развития.

Многие студенты достаточно активны в этом плане и отметили полезность участия, например, в следующих мероприятиях:

- соревнования по информационной безопасности в формате CTF (capture the flag);
- подготовка тезисов для участия в МНСК (ежегодная международная научная студенческая конференция, проводится в НГУ в апреле);
- Docfactor — конференция о технической документации в IT и ее роли в разработке ПО;
- семинар компании Intel «Искусственный интеллект»;
- митапы компании JetBrains;
- стажировки в ЦФТ, стажировки в JetBrains;
- зимняя школа CompTech-NSK и некоторых других.

Задание 4. Проанализировать практику изучения дополнительной литературы по информационным технологиям и образовательных ресурсов Интернета и сообщить о результатах. Отметить достоинства и недостатки конкретных книг, Интернет-порталов и учебных курсов. Дать свои рекомендации.

Результаты выполнения задания показывают значительную активность и широкий кругозор студентов. В сообщениях студентов о выполнении этого задания в числе полезных и интенсивно осваиваемых источников Интернета наряду с широко известными порталами и курсами Интуит.ru и Coursera были названы также следующие сайты: opennet.ru, habr.com, docs.oracle.com, ru.stackoverflow.com, qaru.site, code.mu, nlpх.net, www.cyberforum.ru, javatutor.net, лекции CS-центра, лекции компании Яндекс и др. В числе авторов книг по IT-тематике, выбранных студентами, выполнившими задания в текущем семестре, были названы имена таких авторов, как Джеймс Куроуз, Кит Росс, Эндрю Таненбаум.

Большую помощь в профессиональном развитии, образовании и самообразовании студентам оказывает также система дистанционного обучения и хранилище учебных курсов, развернутые на базе Moodle в НГУ при содействии специалистов компании СофтЛаб-НСК и ИАиЭ СО РАН.

Задание 5. Проанализировать и изложить свое мнение относительно учебного плана, по которому студенты обучаются сейчас, и разработать свой вариант учебного плана, лучшего по мнению студента с точки зрения полезности для будущей профессии. Сравнить эти планы. Обычным стремлением студентов при выполнении данного задания является увеличение числа специальных IT-дисциплин.

Дополнительные активности студентов в плане профессионального развития и профессиональной ориентации необходимы и полезны в дополнение к дисциплинам общего учебного плана и учебной и производственной практике.

Список литературы

1. Материалы АПКИТ. URL: <https://apkit.ru/>.
2. *Лаврентьев М. М., Городняя Л. В., Держо М. А., Мигинский Д. С.* Вопрос карьерных перспектив в области IT : мат-лы конф. АПКИТ «Преподавание IT в РФ — 2019». 2019.

UDK 378

CAREER GUIDANCE FOR IT STUDENTS

M. M. Lavrentiev

Dr. of ph. and math. sc., Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova st., 2, e-mail: mmlavr@nsu.ru

T. S. Vasyuchkova

Ph.D., Novosibirsk State University
630090, Novosibirsk, Russia, Pirogova st., 2

L. V. Gorodnyaya

Ph.D., Researcher A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6,

Associate professor, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova st., 2, e-mail: lidvas@gmail.com

M. A. Derzho

Associate professor, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova st., 2, e-mail: m_derjo[at]mail.ru

N. A. Ivancheva

Associate professor, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova st., 2, e-mail: iva@ci.nsu.ru

I. V. Belago

Researcher
Institute of Automation and Electrometry SB RAS,
630090, Novosibirsk, Russia, Acad. Koptyuga pr., 1, e-mail: misc@softlab-nsk.com

V. S. Bartosh

Software engineer
Institute of Automation and Electrometry SB RAS,
630090, Novosibirsk, Russia, Acad. Koptyuga pr., 1, e-mail: vas@sl.iae.nsk.su

Abstract: The article discusses some aspects of the professional training of students of IT-specialty FIT NSU.

Keywords: IT-specialist, professional standard, Internet educational environments, FIT NGU

References

1. Materialy APKIT. URL: <https://apkit.ru/>.
2. *Lavrentiev M. M., Gorodnyaya L. V., Derzho M. A., Miginskii D. S.* Vopros kar`ernykh perspektiv v oblasti IT : mat-ly konf. APKIT «Prepodavanie IT v RF — 2019». 2019.

ОРГАНИЗАЦИЯ КРИТЕРИАЛЬНОГО ОЦЕНИВАНИЯ ДЕЯТЕЛЬНОСТИ ШКОЛЬНИКОВ НА УРОКАХ ИНФОРМАТИКИ

А. В. Мартынова

учитель информатики высшей категории,
гимназия «Самопознание», республиканское государственное
казенное предприятие ННПООЦ «Бобек»,
050027, Казахстан, Алматы, мкр. Дарын, 47, e-mail: anzhi71@mail.ru

Е. В. Дудышева

канд. пед. наук, доц., Алтайский государственный
гуманитарно-педагогический университет им. В. М. Шукшина,
659333, Бийск, ул. В. Короленко, 53, e-mail: dudysheva@yandex.ru

Аннотация. В статье рассматриваются проблемы критериального оценивания деятельности обучаемых в основной школе на уроках информатики и возможные способы их решения. Предложена схема организации критериального оценивания школьников на уроках информатики на основе формулировки критериев совместно с учителем и дальнейшим самооцениванием. Описана опытно-экспериментальная работа для учеников пятых классов гимназии «Самопознание» Республики Казахстан при изучении темы «Компьютер и безопасность».

Ключевые слова: школьное образование, преподавание информатики, критериальное оценивание, самооценивание

Введение

Проблема аутентичного оценивания учителем деятельности школьников остается актуальной проблемой как в педагогической теории, так и в педагогической практике [1]. К современным методам относят критериальное оценивание [2]. Под критериальным оцениванием принято понимать применение определенных критериев (дескрипторов), в которых достаточно полно отражен образовательный результат. Такой подход нацелен на повышение мотивации и большей вовлеченности обучаемых в образовательный процесс. Критериальное оценивание позволяет обеспечить систему взаимосвязанных контрольно-оценочных действий всех участников образовательного процесса для достижения поставленных целей и задач обучения на основе формирующего оценивания, т. е. оценивания для более эффективной организации самого процесса обучения [3].

Критериальное оценивание как инновация применяется в международной школьной практике с 70-х гг. XX века. С прошлого года система образования Республики Казахстан

также претерпевает довольно серьезные изменения. В обновленной системе преподавания школьных предметов введено критериальное оценивание на официальном, нормативном уровне. Учителям для организации образовательной деятельности предложены рекомендации и примеры дескрипторов по разделам; кроме того, критерии оценивания некоторых работ (в основном, исследовательского характера) прописаны в учебниках. Но по данным критериям выставляются баллы только за суммативные, итоговые проверочные работы, которые проводятся два или три раза в четверть в виде теста, по которым выставляется итоговая оценка путем выведения процентного соотношения. Все же практические работы оцениваются только формативно, а значит, к итоговой оценке баллов не добавляются. Такая система непривычна в первую очередь учителям, а детей даже расхолаживает. Ведь у школьников пятых-восьмых классов еще нет высокой сознательности и мотивации к учебе. В начальной школе они привыкли получать систематический набор оценок за свои работы, а так как текущие оценки не ставят, значит можно и не делать или не стараться. Поэтому к итоговому суммативному оцениванию школьники часто оказываются неподготовленными, получают низкие оценки тогда, когда корректировать образовательный процесс уже поздно. Этому, на наш взгляд, способствует еще и тот факт, что критерии, предложенные в учебнике и методичках расписаны очень подробно и объемно, но слишком по-взрослому, поэтому дети зачастую просто не воспринимают такой большой объем текста.

На уроках информатики критериальное оценивание имеет свои особенности, оно включает оценивание самой деятельности обучаемых, применяемых ими приемов и информационных технологий. Причем в компьютерных классах на практических занятиях школьники работают большей частью самостоятельно. Т. е. требуется не только оценить получаемый результат — презентацию, документ и тому подобное, но и навыки работы с аппаратным и программным обеспечением для решения поставленной задачи, фактически объединить критерии самого образовательного результата и критерии оценивания процесса его достижения. Стоит отметить, что при самостоятельном выполнении задания в классе большим количеством обучающихся, особенно в основной школе, отследить и оценить процесс индивидуального выполнения работы учителю достаточно сложно. Но не меньшую проблему представляет собой объяснение школьникам и их родителям объективных критериев, по которым обучаемому была выставлена итоговая оценка. В статье предлагаются возможные способы преодоления данных проблем.

1. Обсуждение и формулировка критериев со школьниками

Критерии, данные в учебниках и методических указаниях подробно описаны и служат основой для оценивания каждого ученика. Но как показала практика, учащиеся не воспринимают такое количество текста, тем более что эти критерии навязаны им. Например, в таблице приведены критерии для презентации проекта обучаемых, по которым оцениваются работы, начиная с пятого класса (85–100 баллов — «отлично»; 70–84 балла — «хорошо»; 50–69 баллов — «удовлетворительно»; менее 49 баллов — «неудовлетворительно»). Формулировка критериев также может быть непонятной для школьников.

Поэтому основной идеей нашего исследования стало постепенное введение предлагаемых в учебниках критериев оценивания на нескольких уроках с обсуждением их с самими школьниками. В результате обучаемые должны вместе с учителем прийти к схожей формулировке, но понятой ими, а не искусственно введенной. Для того чтобы определить, понятны ли критерии школьникам, используется процедура самооценивания обучаемых по рассмотренным критериям, а также, при необходимости, взаимооценивания (как текущая, формативная

оценка, а не как итоговая оценка по теме). Для информатики удобно формулировать критерии для оценивания деятельности школьников (а не только результата) в виде алгоритмической схемы рекомендуемого выполнения задания.

Критерии для презентации группового проекта обучаемых

Критерий	Обоснование критериев	Балл
Содержание		
Полнота раскрытия темы	– Полно	10
	– Частично	5
	– Не раскрыта	0
Логика изложения информации	– Логичное изложение материала	10
	– Нарушение логики	5
	– Отсутствие логики	0
Самостоятельная работа группы		
Слаженная работа в группе	– Четко спланированная работа группы	5
	– Работа группы частично спланирована	3
	– Не спланирована работа в группе	0
Авторская оригинальность	– Уникальная работа. Содержится большое число оригинальных, изобретательных примеров	5
	– В работе присутствуют авторские находки	3
	– Стандартная работа, не содержит авторской индивидуальности	0
Оформление работы (презентации)		
Идеи и содержание	– Полностью раскрыты и обоснованы основные идеи проекта. Включены соответствующие ссылки на другие источники информации по тематике	5
	– Раскрыты основные идеи проекта	3
	– Основные идеи показаны на основе готовых таблиц, выводов	0
Грамотность	– Организационная структура ясна и очевидна. Грамматика и используемая терминология верны. Синтаксис правильный	5
	– Грамматика и используемая терминология почти верны. Есть синтаксические ошибки. Синтаксис почти верный	3
	– Грамматика и использование терминов неверны. Синтаксис неверен	0
Внешний вид	– Внешний вид дает возможность легко воспринимать содержание. Фон и текст соответствуют друг другу. Графические элементы необходимы и достаточны	5
	– Внешний вид почти всегда дает возможность легкого восприятия содержания. Фон почти всегда соответствует тексту. Возможно несоответствие количества или качества графических элементов	3
	– Внешний вид не соответствует эстетическим требованиям, содержание трудно воспринимается. Слабое соответствие между фоном и текстом. Графика плохо соответствует теме	0
Защита работы		
Качество доклада	– Аргументированность основных позиций, логичность композиции доклада, полнота представления в докладе результатов работы, отражает межпредметные связи	10
	– Нарушение логики выступления, неполное представление результатов работы, неполная система аргументации, не отражает межпредметные связи	5
	– Не заявлены аргументы по основным позициям, полное нарушение логики, не представлены результаты исследования	0

Для обучаемых в основной школе в рамках курса информатики мы разработали следующую схему организации критериального оценивания школьников, представленную ниже.

1. Подготовительный этап для учителя:

1.1. планирование серии уроков информатики и ИКТ по разделу с предварительным планированием всех критериев оценивания, их разбиения для обсуждения на уроках;

1.2. составление заданий для учащихся соответственно темам (в условиях индивидуальной или совместной работы);

1.3. разработка алгоритма выполнения практических работ.

2. Постепенное введение учащихся в систему самостоятельной формулировки критериев оценивания (со временем уровень самостоятельности учащихся возрастает, степень участия учителя уменьшается; алгоритмы выполнения работы становятся менее подробными, вплоть до крайне схематичного описания):

2.1. после введения в тему урока обсуждение предложенного учителем алгоритма выполнения практической работы;

2.2. опираясь на формулировки в алгоритме, учащиеся предлагают критерии оценивания работы, совместно с учителем обсуждают и четко их формулируют.

3. Выполнение практической работы и оценивание качества ее выполнения по сформулированным критериям:

3.1. при выполнении работы ученики опираются на выработанные критерии, корректируя работу соответственно им;

3.2. используется само- и взаимопроверка соответственно критериям;

3.3. суммативное (итоговое) оценивание по разделу для выставления официальной оценки по нормативным критериям, понятным обучаемыми в ходе предшествующего обучения.

Для проведения экспериментальной работы разработаны и апробированы поурочные планы по информатике для пятых гимназии «Самопознание» ННПООЦ «Бобек» по разделу «Компьютер и безопасность», изучаемому в первой четверти. Основных критерия для данного раздела два: выделение последствий нарушения техники безопасности и определение названия устройств по назначению, также планом предусмотрена практическая проектная работа обучаемых. При проведении уроков со школьниками обсуждались критерии и конкретные дескрипторы, за которые выставляются баллы, а также совместно с обучаемыми был разработан алгоритм подготовки презентации с источниками, взятыми из сети Интернет. Для последующего оценивания защиты проектов с учениками пятых классов обсуждались возможные критерии (учитель опирался на рекомендации из таблицы), которые формулировались вместе с ребятами понятным для них текстом. К таким критериям отнесены (в формулировке учеников):

– раскрытие темы;

– соответствие теме;

– соблюдение авторских прав (наличие ссылок на источники);

– красота оформления;

– грамотность;

– качество презентации.

Максимальный балл за каждый критерий составляет 5 баллов. Систему распределения баллов предложили ребята сами, что, несомненно, облегчило оценивание работы. Проговорив формулировки данных критериев, школьники уже выполняли работу придерживаясь их, корректируя свою деятельность. Так как в течение всей работы у учеников имелись листочки с критериями, они могли корректировать работу с их учетом, то они далее достаточно быстро оценили собственные результаты, не испытывая особых затруднений.

2. Диагностика экспериментальной работы

Для проведения опытно-экспериментальной работы выбраны группы из двух пятых классов гимназии «Самопознание» ННПООЦ «Бобек»: они первый год изучали информатику и первый год работали по обновленной системе критериального оценивания.

На констатирующем этапе эксперимента учащимся экспериментального и контрольного классов был предложен опросник / анкета самооценивания со следующими вопросами (ответы — «да», «часто», «редко», «нет») и возможностью прокомментировать ответ («почему?»).

Нравится ли мне, как выставляется оценка по критериям?

Нравится ли мне больше оценивание в этом году, чем в прошлом году?

Показывает ли оценка по критериям правильно мой уровень знаний и умений?

Понятно ли мне, как выставляется оценка в этом году?

Влияет ли сейчас на оценку отношение учителя и одноклассников?

Соглашаюсь ли я с критериями, по которым оценивается работа?

Легко ли по заданным критериям самостоятельно оценить себя или товарищей?

В этом году после оценивания у меня желание учиться больше, чем в прошлом?

По результатам анализа анкет стало видно, что они некоторыми ответами противоречат сами себе, что можно объяснить их неуверенностью и недопониманием создавшейся ситуации.

Основной задачей в ходе проведения эксперимента было проверить, насколько меняется отношение к критериальному оцениванию на нескольких уроках, если разрабатывать критерии совместно с учащимися по предложенной и описанной схеме, а также выяснить, насколько меняется отношение к само- и взаимооцениванию. Для экспериментальной группы (класса) критерии формулировались вместе с учениками, как описано выше. Для контрольной группы (класса) проводились занятия с суммативным применением критериев по результатам выполнения проектной работы и теста по теме «Компьютер и безопасность». Необходимо отметить, что обе группы успешно справились с обязательными тестами по теме. Далее проведен повторный опрос с анкетированием для выявления положительных сдвигов на основе критерия знаков. Он выявил некоторые статистически значимые отличия между контрольным и экспериментальным классом в отдельных аспектах. Пример заполнения повторной анкеты приведен на рисунке.

~ 0,00

Анкета учащегося (выходная)

- Можно сделать лучше оценивание по критериям?
 Да Часто Редко Нет
 Почему? так лучше
- Стало ли мне нравится больше мое оценивание за неделю?
 Да Часто Редко Нет
 Почему? лучше самой себя оценивать
- Стала оценка более верно показывать мой уровень знаний за неделю?
 Да Часто Редко Нет
 Почему? я вникаю
- Понятней ли мне стало за неделю, как выставляется оценка?
 Да Часто Редко Нет
 Почему? не знаю
- Изменилось ли влияние на мою оценку учителя и одноклассников за неделю?
 Да Часто Редко Нет
 Почему? не знаю
- Изменилось ли мое согласие с критериями оценивания за неделю?
 Да Часто Редко Нет
 Почему? не знаю
- Стало ли мне легче самостоятельно оценивать себя и товарищей?
 Да Часто Редко Нет
 Почему? ну, немного трудно
- Стало ли на этой неделе после оценивания у меня желание учиться больше, чем в раньше?
 Да Часто Редко Нет
 Почему? уточа это хорошо

Пример повторного опроса / анкетирования обучающихся

Проанализировав полученные данные, сделаны следующие выводы:

– у учащихся обоих классов незначительны изменения показателей объективности и понятности выставления оценки по критериям, что говорит о том, что учащиеся изначально понимали принципы критериального оценивания и в ходе эксперимента (в контрольном классе в том числе, в связи с использованием критериального оценивания на всех уроках), скорее, в своем мнении просто утвердились, что также связано с относительной краткостью экспериментальной работы;

– ребятам из экспериментального класса стало несколько легче оценивать себя и своих одноклассников после того, как они стали самостоятельно формулировать критерии оценки;

– с предыдущим вопросом связано и то, что многие ребята из экспериментального класса на вопрос «нравится ли выставление оценок по критериям?» изменили ответ в сторону положительного, чего нельзя сказать про контрольный класс;

– и, конечно же, немаловажным достижением нашей работы стало то, что у ребят экспериментальной группы возросло желание учиться, так как в начале четверти они были напуганы сменой привычной системы оценивания: конечно, в обоих классах за четверть прошла некоторая адаптация к данной системе, но в контрольном классе значительных изменений в этом вопросе не наблюдалось.

Таким образом, предложенная схема организации критериального оценивания школьников на уроках информатики позволяет улучшить, но не окончательно сформировать понимание принципов критериального оценивания (объективности и прозрачности критериев); при этом значительно улучшается отношение школьников к обучению с применением критериального оценивания, к само- и взаимооцениванию, а также формируются первичные умения само- и взаимооценивания школьников основной школы.

Заключение

В статье рассмотрены проблемы критериального оценивания деятельности обучаемых в основной школе на уроках информатики и возможные способы их решения. Предложена схема организации критериального оценивания школьников на уроках информатики на основе формулировки критериев совместно с учителем и дальнейшим самооцениванием. Описана опытно-экспериментальная работа для учеников пятых классов гимназия «Самопознание» Республики Казахстан при изучении темы «Компьютер и безопасность».

Список литературы

1. *Воронцов А. Б.* Педагогическая технология контроля и оценки учебной деятельности. Образовательная система Д. Б. Эльконина В. В. Давыдова. М. : Изд-во Рассказовъ, 2002.
2. *Gillet J. W., Temple Ch., Crawford A.* Understanding reading problems: Assessment and instruction, 6th ed. White Plains, N. Y.: Longman, 2004.
3. *Красноборова А. А.* Критериальное оценивание в школе : учеб. пособие. Пермь : ПГПУ, 2010.

UDK 372.800.2

**ORGANIZATION OF CRITERIA-BASED
ASSESSMENT OF SCHOOLCHILDREN'S ACTIVITIES**

A. V. Martynova

Informatics teacher, Gymnasium "Self-knowledge", Bobek,
050027, Kazakhstan, Almaty, Kazakhstan, Daryn, 4, e-mail: anzhi71@mail.ru

E. V. Dudysheva

Cand. Ped. Scs, Associate Professor, The Shukshin Altai State Humanitarian-Pedagogical University,
659333, Biysk, V. Korolenko Str, 53, e-mail: dudysheva@yandex.ru

Annotation. The article deals with the problems of criteria-based assessment of students in basic school in informatics lessons and proposes possible solutions. The scheme has been proposed for organizing the criteria-based assessment of schoolchildren in informatics lessons based on the formulation of criteria in conjunction with a teacher and further self-assessment of students. Experimental work for fifth-grade pupils of the "Self-knowledge" gymnasium of the Republic of Kazakhstan is described in the study of the topic "Computer and Security".

Keywords: school education, informatics teaching, criteria-based assessment, self-assessment

References

1. *Voroncov A. B.* Pedagogicheskaya tekhnologiya kontrolya i ochenki uchebnoj deyatel'nosti. Obrazovatel'naya sistema D. B. EHI'konina V. V. Davydova. M. : Izd-vo Rasskazov, 2002. (in Russian)
2. *Gillet J. W., Temple Ch., Crawford A.* Understanding reading problems: Assessment and instruction, 6th ed. White Plains, N. Y. : Longman, 2004.
3. *Krasnoborova A. A.* Kriterial'noe ocenivanie v shkole: uchebnoe posobie. Perm' : PGPU, 2010. (in Russian)

ИСПОЛЬЗОВАНИЕ КОМПЕТЕНТНО-ОРИЕНТИРОВАННЫХ ЗАДАНИЙ РАЗНОЙ СЛОЖНОСТИ ДЛЯ ОБУЧЕНИЯ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЮ В КЛАССАХ С РАЗНЫМ УРОВНЕМ ГОТОВНОСТИ ШКОЛЬНИКОВ К ИЗУЧЕНИЮ ИНФОРМАТИКИ

В. И. Новожилова

зав. кафедрой информатики, учитель информатики,
БОУ «Югорский физико-математический лицей-интернат»,
628011, Ханты-Мансийск, ул. Мира, 151, e-mail: valentina.novozhilova@uriit.ru

Аннотация. В данной работе представлены методика составления компетентностно-ориентированных заданий и задачник для обучения основам программирования в профильном курсе «Информатика и ИКТ» с применением этой методики. Методика и задачник разрабатывались с учетом формирования у школьников ключевых компетенций, определяющих современное качество образования в рамках ФГОС. Задачник и методика опробованы при обучении алгоритмизации и программированию учеников 10–11-х классов Югорского физико-математического лицея-интерната в период с 2005 по 2018 гг. Компетентностно-ориентированные задания разработаны для всех изучаемых тем в базовой и повышенной сложности. Таким образом, для разных групп школьников с разной подготовкой и разными целями можно построить индивидуальный план изучения программирования, варьируя скорость и сложность освоения программы.

Предлагаемый сборник задач связанная с методикой составления компетентностно-ориентированных заданий разной сложности могут использоваться не только в школах с разными программами по информатике базового и профильного уровней, но и в классах с разным уровнем готовности школьников к изучению информатики. А также в школах и учреждениях дополнительного образования при подготовке участников олимпиад по информатике школьного и муниципального уровней.

Ключевые слова: информатика, компетентность школьника, алгоритмизация и программирование, компетентностно-ориентированные задания, индивидуальные траектории обучения, базовый и профильный уровни обучения, дополнительное образование

Введение

Существует несколько причин, почему необходима методика компетентностно-ориентированных индивидуальных заданий в преподавании алгоритмизации и программирования.

Причина первая. Необходимость подготовки лицейстов в кратчайшие сроки к участию в олимпиадах различного уровня заставила пересмотреть методику преподавания алгоритмизации, в частности, методику подбора задач. Задач опубликовано очень много, но решить их все

невозможно [1–12]. Задачники, как правило, разделены на части в соответствии с изучаемым алгоритмом, типом данных или способом программирования. Некоторые из них позволяют более подробно проработать применение различных операторов или решение некоторых типов задач [2, 5]. Поэтому требуется выбрать подмножество задач, каждая из которых несет новую информацию, развивает новый необходимый навык, увеличивая тем самым компетентность школьника в разработке алгоритмов и технике программирования.

Причина вторая. В физико-математический лицей школьники отбираются в зависимости от суммарного проходного балла по физике и математике, но никто не проверяет их знания по информатике. В лицей приходит около 63 % школьников, которые не изучали программирование или изучали «немного». Поэтому возникла необходимость в индивидуальных маршрутах обучения программированию различных групп школьников. А для этого все задачи нужно разделить на группы по степени сложности в соответствии с классификацией ФИПИ: базовая, повышенная, высокая сложность. Кроме того, нужны задачи преолимпиадного и олимпиадного уровня для особо одаренных школьников.

Причина третья. Индивидуальные потребности школьников. Часть школьников подходит к обучению крайне практично. Предметы, которые являются обязательными для поступления в ВУЗ, изучаются школьниками углубленно, остальные предметы — в базовом формате. Поэтому даже школьникам, занимающимся углубленно химией, биологией и другими дисциплинами, может понадобиться изучение информатики на базовом уровне.

Причина четвертая. Как правило, опубликованные задачи имеют уже формализованное условие (например, «в заданной матрице найти ...»), при решении которых не нужно думать о представлении входных и выходных данных. Тем самым исключаются для начинающих программистов такие этапы технологии программирования, как формализация, проектирование данных, выбор математической модели. Олимпиадные же задачи имеют другие особенности: входные и выходные данные описаны, требуется разработать алгоритм и запрограммировать, применяя современные технологии. Это заставляет при составлении текста задачи учитывать ее сложность и индивидуальную подготовку школьника, которому адресована задача.

1. Методика формирования компетентностно-ориентированных заданий

Разработке компетентностно-ориентированных заданий и составлению сборника задач предшествовала большая методическая работа. На первом этапе были собраны различные типы задач из сборников задач разных авторов [1–12] по каждой теме обучения алгоритмизации и программированию. Решения этих задач были проанализированы и явно выписаны типы использованных алгоритмов. В список алгоритмов были добавлены обязательные алгоритмы из спецификации ЕГЭ по информатике и простые алгоритмы решения олимпиадных задач. Далее были явно сформулированы навыки, которыми должен обладать школьник, чтобы научиться решать все выделенные типы задач. Навыком на данном этапе будем считать не только умение использовать простые алгоритмы и их части, но и знание типов данных, действий над ними, операторов изучаемого языка программирования и умение манипулировать такими данными как время, дата, цифры числа, правильно выбрать тип оператора ветвления или цикла и пр. Чтобы приобрести каждый из этих навыков, была собрана серия однотипных задач разной сложности. Минимальное количество задач в серии равно количеству учеников в группе. Чтобы составить серию задач для выработки одного навыка, использовались задачи других авторов, или типичные задачи, модифицированные путем добавления фабулы, или разрабатывались новые задачи. Каждая серия задач имеет заголовок, в котором явно описан

тренируемый навык. Явная идентификация навыка, иначе — явная формулировка проблемы позволяет школьнику легко понять, что он него требуется при решении именно этой задачи и как он должен доказать свою компетентность в решении этого типа задач.

Задачи для составления компетентностно-ориентированных заданий разработаны для каждой изучаемой темы, поэтому и задачник содержит 11 глав: линейный, ветвящийся алгоритмы, циклы одинарные и вложенные, массивы одномерные и двумерные, символы и строки, процедуры и функции, файлы, записи, множества. Внутри темы задачи разделены на два блока по уровню сложности: базовая, повышенная и высокая. Оба блока задач разделены на серии по типам навыков. Внутри каждого блока серии задачи, соответствующие разным навыкам, упорядочены по нарастанию сложности задач, их формирующих. Блоки задач базовой сложности сопровождаются тестами с коротким ответом. В сборник задач они не вошли.

Компетентностно-ориентированное задание составляется следующим образом: сначала нужно выбрать сложность, далее в рамках этой сложности перебрать все серии задач, соответствующие разным навыкам, и из каждой серии взять по одной задаче. Самый простой вариант получится, если из каждой серии взять первую задачу, самый сложный — если из каждой серии взять последнюю задачу.

2. Примеры выбора навыков для различных тем

Далее приведены списки навыков для первых пяти тем, изучаемых в школах с программами по алгоритмизации и программированию базового уровня.

Линейный алгоритм. Для изучения линейного алгоритма разработан только один блок задач базовой сложности, который содержит четыре серии задач для формирования четырех типов навыков:

- 1) задачи для изучения синтаксиса операторов присваивания, ввода, вывода, использования целых и вещественных переменных;
- 2) задачи, требующие ввода рабочих переменных;
- 3) простые задачи на целочисленное деление по модулю 10;
- 4) задачи на выделение цифр в десятичной записи натурального числа путем целочисленного деления.

Ветвящийся алгоритм. Для изучения ветвящегося алгоритма разработаны два блока задач базовой и повышенной сложности, которые содержат 11 серий задач для формирования 11 навыков.

Таблица 1

Формирование навыков по теме «Ветвящийся алгоритм и условные операторы»

Блок задач базового уровня сложности для формирования следующих пяти навыков	Блок задач повышенного и высокого уровня сложности для формирования следующих шести навыков
1. Задачи на составление условных выражений; 2. Составление условных и логических выражений. Целочисленная арифметика; 3. Составление условных и логических выражений. Простые задачи на геометрическое место точек; 4. Задачи с использованием оператора выбора; 5. Использование сложного условного оператора	1. Пересечения геометрических тел; 2. Алгоритмы ЕГЭ. Найти минимальное (максимальное) из трех или четырех чисел; 3. Преобразование целых десятичных чисел без использования циклов; 4. Целочисленное деление по любому модулю; 5. Даты и время; 6. Сложные задачи на геометрическое место точек

Циклический алгоритм. Для изучения циклического алгоритма разработаны два блока задач базовой и повышенной сложности, которые содержат 9 серий задач для формирования 9 различных навыков.

Таблица 2

Формирование навыков по теме «Циклический алгоритм»

Блок задач базового уровня сложности для формирования следующих пяти навыков	Блок задач повышенного и высокого уровня сложности для формирования следующих трех навыков
1. Понятие итерации, использование цикла со счетчиком. Различать значение и номер элемента последовательности. Последовательность задается итерационной формулой; 2. Вычисление суммы или произведения элементов последовательности; 3. Обработка данных фиксированной длины во время ввода; 4. Использование операторов цикла с предусловием; 5. Использование операторов цикла с постусловием	1. Вычисление сумм по сложным формулам или с заданной точностью; 2. Преобразование целых чисел без использования циклов, при разработке алгоритма описать и запрограммировать полную систему решений задачи; 3. Применение различных операторов цикла в зависимости от контекста задачи; 4. Применение переменных-сумматоров и переменных-«флагов»

Условия входа для оператора цикла с предусловием и условие выхода для оператора с постусловием сформулировать так, чтобы после цикла не пришлось корректировать результат работы цикла. Задания базового уровня сложности выполняют все обучающиеся алгоритмизации и программированию. Приобретенных навыков достаточно, чтобы решить задачи ЕГЭ с циклами (8, 20, 24).

Вложенные циклы. Практически все задачи, в которых нужно применять алгоритм вложенных циклов, являются задачами повышенной или высокой сложности. Но этот раздел необходимо включить в программу базового уровня, так как навыки 3) и 4) необходимы школьнику для решения задачи 27 ЕГЭ. Кроме того, в этой главе есть ограничение: мы не рассматриваем задачи с одномерными и двумерными массивами. Поэтому все задачи объединены в один блок и выделено 4 серии задач для формирования 4 различных навыков:

- 1) манипуляции с цифрами десятичного числа;
- 2) алгоритмы работы с целыми числами, делители составные и простые;
- 3) задачи с вводом данных во время выполнения программы;
- 4) различные задачи с перебором всех вариантов и выбором ответа по условию задачи, перебор с отсечением.

Одномерные массивы. Для изучения алгоритмов работы с одномерными массивами разработаны три блока задач базовой (7 серий задач), повышенной (2 серии задач) сложности и специальных алгоритмов (4 серии задач), которые содержат суммарно 13 серий задач для формирования 13 различных навыков.

Таблица 3

Формирование компетентностно-ориентированных навыков по теме «Вложенные циклы»

Блок задач базового уровня сложности для формирования следующих пяти навыков	Блок задач ЕГЭ, повышенной и высокой сложности для формирования следующих семи навыков
1. Индексы и значения элементов массива; 2. Манипуляции с элементами массива;	<i>Алгоритмы ЕГЭ</i> 1. Вставить элемент с заданным значением в упорядоченный по возрастанию (или убыванию) массив;

Блок задач базового уровня сложности для формирования следующих пяти навыков	Блок задач ЕГЭ, повышенной и высокой сложности для формирования следующих семи навыков
3. Выбрать из массива элементы с заданными свойствами и вычислить их сумму и среднее арифметическое; 4. Вычислить минимальный и максимальный элементы массива с использованием алгоритма, который не зависит ни от типа элементов массива, ни от диапазона данных, хранящихся в массиве; 5. Манипуляции с цепочками подряд идущих элементов массива; 6. Вычислить максимальный или минимальный элементы в подмножестве данных массива; 7. Поиск в массиве элемента с заданным значением	2. Удаление элементов массива; 3. Найти два (или три) максимальных (или минимальных) элемента за один просмотр массива; 4. Циклический сдвиг элементов массива. <i>Задачи повышенного и высокого уровня сложности</i> 1. Применение нескольких базовых алгоритмов работы с элементами одномерных массивов в зависимости от контекста задачи; 2. Одномерные массивы и вложенные циклы

Таким образом, даже этот перечень навыков для пяти тем показывает, как для разных групп школьников можно построить индивидуальную траекторию изучения программирования с разной скоростью и разной сложностью освоения программы.

Сборник задач содержит блоки и серии задач для всех тем, изучаемых в разделе алгоритмизация и программирование курса «Информатика и ИКТ» как базового, так и для профильного уровней.

Заключение

Методика обучения с использованием индивидуальных компетентностно-ориентированных заданий различной сложности позволяет одновременно обучать школьников из одного класса с разным уровнем подготовки по программированию.

Методика обучения с использованием индивидуальных компетентностно-ориентированных заданий является универсальной, не зависит от языка программирования. При смене языка программирования меняется лишь содержание лекций по синтаксису и семантике языка и примеры разбора задач на языке программирования, но не меняется содержание задач и методика составления индивидуальных компетентностно-ориентированных заданий.

Задачи базовой сложности решают все обучающиеся. Задачи повышенной и высокой сложности решают школьники в зависимости от мотивации, индивидуальных способностей и результатов обучения решения задач базового уровня. Школьники, которые хорошо освоили решение задач базового уровня, на потоковой контрольной работе не могут получить неудовлетворительную оценку. Школьники, которые хорошо освоили решение задач повышенного и высокого уровней сложности, могут без дополнительной подготовки участвовать в олимпиадах школьного и муниципального уровней.

Список литературы

1. Программирование в примерах и задачах Т. Ю. Грацианова. М. : БИНОМ. Лаборатория знаний, 2014. 349 с. : ил.

2. *Андреева Е. В.* Программирование — это так просто, программирование — это так сложно. Современный учебник программирования. М. : МЦНМО, 2009. 184 с.
3. *Белоусова Л. И.* Сборник задач по курсу информатики Л. И. Белоусова, С. А. Веприк, А. С. Муравка / под ред. Л. И. Белоусовой. М. : Экзамен, 2007. 253 с.
4. *Златопольский Д. М.* Программирование: типовые задач, алгоритмы, методы. М. : БИНОМ. Лаборатория знаний, 2007. 223 с. : ил.
5. *Златопольский Д. М.* Сборник задач по программированию. 2-е изд., перераб. и доп. СПб. : БХВ-Петербург, 2007. 240 с. : ил.
6. *Казиев В. М.* Информатика в примерах и задачах : кн. для учащихся 10–11 кл. М. : Просвещение, 2007. 304 с. : ил.
7. Алгоритмизация и программирование / И. Н. Фалина и др. М. : КУДИЦ-ПРЕСС, 2007. 276 с.
8. Информатика. Задачник-практикум : в 2 т. / Л. А. Залогова и др.; под ред. И. Г. Семакина, Е. К. Хеннера. Т. 1. 3-е изд., испр. М. : БИНОМ. Лаборатория знаний, 2006. 309 с. : ил.
9. Информатика. Задачник-практикум : в 2 т. / Л. А. Залогова и др.; под ред. И. Г. Семакина, Е. К. Хеннера. Т. 2. 3-е изд., испр. М. : БИНОМ. Лаборатория знаний, 2006. 294 с. : ил.
10. *Погодина Т. П.* Сборник задач по программированию на языке Паскаль : учеб. пособие. Челябинск : Физ.-мат. лицей № 31, 2003. 64 с.
11. *Гуденко Д. А.* Сборник задач по программированию / Д. А. Гуденко, Д. В. Петроченко. СПб. : Питер, 2003. 475 с. : ил.
12. Задачи по программированию / С. А. Абрамов и др. М. : Наука, 1988. 224 с.

UDK 519.85:373

**USING COMPETENCE-ORIENTED TASKS OF VARYING COMPLEXITY
FOR TEACHING ALGORITHMS AND PROGRAMMING IN CLASSES
WITH DIFFERENT LEVELS READINESS OF PUPILS TO STUDY COMPUTER SCIENCE**

V.I. Novozhilova

Head of the It-Department, It-teacher,
Ugra Physics and Mathematics Boarding School,
628011, Khanty-Mansiysk, Mira St.,151, e-mail: valentina.novozhilova@uriit.ru

Abstract: This paper presents the methodology for developing competence-oriented tasks and a problem book for teaching the basics of programming in the profile course "Informatics and ICT". The methodology and the task book were developed taking into account the formation of key competences among schoolchildren, which determine the modern quality of education in the framework of the FSES. The problem book and methodology were tested in teaching algorithms and programming for students of the 10–11 grades of the Ugra Physics and Mathematics Boarding School from 2005 to 2018. Competence-oriented tasks are developed for all studied topics in basic and advanced complexity. It was revealed for different groups of schoolchildren with different training and goals, it is possible to create an individual plan depending from the tasks. The proposed collection of tasks and the associated methodology can be used in classes with different levels of schoolchildren's readiness to study computer science, while training participants in computer science and municipal computer science competitions.

Keywords: computer science, schoolchildren's competence, algorithmization, programming, competence-oriented tasks, individual training, levels of education, additional education

References

1. Programmirovaniye v primerah i zadachah T. Ju. Gracianova. M. : BINOM. Laboratoriya znaniy, 2014. 349 p. : il. (in Russian)
2. *Andreeva E. V.* Programmirovaniye — jeto tak prosto, programmirovaniye — jeto tak slozhno. Sovremennyy uchebnik programmirovaniya. M. : MCNMO, 2009. 184 p. (in Russian)
3. *Belousova L. I.* Sbornik zadach po kursu informatiki L. I. Belousova, S. A. Veprik, A. S. Muravka / pod red. L. I. Belousovoj. M. : Jekzamen, 2007. 253 p. (in Russian)
4. *Zlatopol'skij D. M.* Programmirovaniye: tipovye zadach, algoritmy, metody. M. : BINOM. Laboratoriya znaniy, 2007. 223 p. : il. (in Russian)
5. *Zlatopol'skij D. M.* Sbornik zadach po programmirovaniyu. 2-e izd., pererab. i dop. SPb. : BHV-Peterburg, 2007. 240 p. : il. (in Russian)
6. *Kaziev V. M.* Informatika v primerah i zadachah : kn. dlja uchashhihsja 10–11 kl. M. : Prosveshhenie, 2007. 304 p. : il. (in Russian)
7. Algoritmizacija i programmirovaniye / I. N. Falina i dr. M. : KUDIC-PRESS, 2007. 276 p. (in Russian)
8. Informatika. Zadachnik-praktikum : v 2 t. / L. A. Zalogova i dr.; pod red. I. G. Semakina, E. K. Henera. T. 1. 3-e izd., ispr. M. : BINOM. Laboratoriya znaniy, 2006. 309 p. : il. (in Russian)
9. Informatika. Zadachnik-praktikum : v 2 t. / L. A. Zalogova i dr.; pod red. I. G. Semakina, E. K. Henera. T. 2. 3-e izd., ispr. M. : BINOM. Laboratoriya znaniy, 2006. 294 p. : il. (in Russian)
10. *Pogodina T. P.* Sbornik zadach po programmirovaniyu na jazyke Paskal' : ucheb. posobie. Cheljabinsk : Fiz.-mat. licej № 31, 2003. 64 p. (in Russian)
11. *Gudenko D. A.* Sbornik zadach po programmirovaniyu / D. A. Gudenko, D. V. Petrochenko. SPb. : Piter, 2003. 475p. : il. (in Russian)
12. Zadachi po programmirovaniyu / S. A. Abramov i dr. M. : Nauka, 1988. 224 p. (in Russian)

ЭЛЕКТРОННЫЕ РЕСУРСЫ ДЛЯ РАЗВИТИЯ ОДАРЕННОСТИ ШКОЛЬНИКОВ

К. В. Мелешко

студент, ВКИ НГУ,
630058, Новосибирск, ул. Русская, 35

В. Г. Перкова

директор ОблЦИТ
630102, Новосибирск, ул. Восход, 26а, e-mail: pvg@oblcit.ru

Э. Э. Сапрыкин

зам. директора ОблЦИТ
630102, Новосибирск, ул. Восход, 26а, e-mail: go@oblcit.ru

Г. А. Сапрыкина

канд. пед. наук, методист ОблЦИТ
630102, Новосибирск, ул. Восход, 26а, e-mail: go@oblcit.ru

Аннотация. В работе рассматривается трансформация понятия «одаренность» с древних времен до наших дней. Приводится описание электронного ресурса для развития логического мышления у школьников. Логическое мышление рассматривается в статье в качестве важного компонента в развитии одаренности.

Ключевые слова: одаренность, интеллект, креативность, мотивация, мышление, логика

Введение

Вопросы, связанные с одаренностью, издавна вызывали повышенный интерес в педагогической практике. И не только в ней. Этим вопросом интересовались и государственные деятели, озабоченные решением государственной проблемы формирования интеллектуального потенциала общества, выявлением одаренных детей и созданием условий для развития их одаренностей. Именно решение государственных задач побуждало античных философов обращать внимание на формирование правящей элиты за счет одаренных детей. Платон (жил между 429/427 до н. э.—347 до н. э.) полагал, что надо специально отбирать и развивать наиболее способных детей. В «Государстве» Платон выдвинул идею общественного воспитания детей с самого младшего возраста. «...Только лица, у которых разумная часть души, до 30 лет изучают философию, развивают отвлеченное мышление, изучая в этих целях геометрию, арифметику, астрономию, теорию музыки, и готовятся к государственной деятельности. Выделившиеся в

этой группе самые сильные в интеллектуальном отношении продолжают свои занятия и, овладев философией, становятся правителями государства» [1].

К изучению природы гениальности исследователи вплоть до начала XIX века обращались лишь постольку, поскольку это было необходимо для выяснения общих проблем творчества. Эти представления породили и соответствующую терминологию. С древнейших времен, вплоть до XIX века (А. Баумгартен, Г. Гегель, И. Кант и др.), в научных трактатах охотно использовали термин «гений» (от лат. *genius* — дух). Практически одновременно с этим термином появился и термин «талант». Вероятнее всего появление термина «талант» в научном мире было связано с представлениями о возможности измерения степени гениальности и на этой основе ранжирования гениев. Постепенно сформировалось представление о таланте как о просто высокой степени развития способностей к определенному виду деятельности, в то время как под «гением» стали понимать высший, максимальный уровень их проявления, расположенный, образно говоря, над талантом [2].

Понятие «одаренность» в историческом ракурсе

Термин «одаренность» появляется позже. Корень слова — «дар». Если вдуматься, то это слово означает, что что-то даровано человеку свыше, а не выработано им самим. Одной из первых попыток глубокого психологического осмысления проблемы одаренности была предпринята в исследованиях испанского врача, жившего в эпоху Возрождения, — Хуана Уарте (1530–1588) [3], который был учителем и вдохновителем будущих философов Фрэнсиса Бэкона (1561–1626) и Баруха Спинозы (1632–1677) (последний философ учился по его материалам уже после смерти Уарте). Он связывал перспективу возрождения могущества Испанской империи с максимальным использованием на государственной службе особо одаренных людей. Его работа была одним из первых в истории психологии трудов, где рассматривалось в качестве основной задачи изучение индивидуальных различий в способностях с целью дальнейшего профессионального отбора.

В качестве основных способностей он выделял следующие: фантазию (воображение), память и интеллект. Х. Уарте подчеркивал зависимость таланта от природы, однако это, по его мнению, не означает бесполезности воспитания и труда. При этом, говоря о воспитании и обучении талантов, он акцентировал внимание на необходимости учета индивидуальных и возрастных особенностей обучаемых. Х. Уарте высказывал мысль о необходимости создания государственной системы профессионального отбора. Он писал о том, что для того, чтобы никто не ошибался в выборе профессии, которая больше всего подходит к его природному дарованию, следовало бы выделить уполномоченных людей великого ума и знания, которые открыли бы у каждого его дарование еще в нежном возрасте. Тогда они заставили бы его обязательно изучать ту область знания, которая ему подходит.

В конечном итоге Хуан Уарте пришел к следующим заключениям:

1) из многих видов талантов, имеющих в человеческом роде, ты можешь по преимуществу владеть только одним; если только природа в тот момент, когда она тебя формировала, не была очень сильна и не сосредоточила все свои силы, чтобы соединить в тебе два или три вида дарований, или же наоборот, вследствие бессилия, не сделала тебя глупым и лишенным всякого дарования;

2) каждому виду дарований соответствует, по преимуществу, только одна наука и не больше; таким образом, если тебе не удастся выбрать ту науку, которая соответствует твоей природной способности, то ты очень мало успеешь в других науках, хотя бы ты трудился дни и ночи;

3) после того как ты открыл, какова та наука, которая больше всего соответствует твоему дарованию, тебе остается преодолеть еще одну, более значительную трудность, а именно: исследовать, способен ли ты больше к теории или к практике. Ибо во всех родах наук теория и практика так противоречат друг другу и требуют столь различных дарований, что они взаимно ослабляют друг друга, как будто бы они были настоящими противниками [4].

На сегодняшний день большинство психологов признает, что уровень, качественное своеобразие и характер развития одаренности — это всегда результат сложного взаимодействия наследственности (природных задатков) и социальной сферы опосредованного деятельностью ребенка (игровой, учебной, трудовой). Коллектив русских психологов по заданию Министерства образования сделал интересную разработку. Руководили этим проектом известные психологи Д. Б. Богоявленская и В. Д. Шадриков [5]. По их мнению, «одаренность — это системное, развивающееся в течение жизни качество психики, которое определяет возможность достижения человеком более высоких, незаурядных результатов в одном или нескольких видах деятельности по сравнению с другими людьми» [5].

К числу наиболее популярных в современной психологии моделей одаренности относится концепция, разработанная американским ученым Джозефом Рензулли [6, 7]. Согласно этой концепции, одаренность есть не просто высокий коэффициент интеллекта или высокая креативность, это сочетание трех основных характеристик: интеллектуальных способностей (превышающих средний уровень), креативности и настойчивости (мотивация, ориентированная на задачу).

Следует отметить, что в самом названии данной теоретической модели Дж. Рензулли использует вместо термина «одаренность» термин «потенциал». Это свидетельство того, что данная концепция — своего рода универсальная схема, применимая для разработки системы воспитания и обучения не только одаренных, но и всех детей. *«Одаренные и талантливые дети — это дети, которые обладают данными характеристиками или способны развить и реализовать их в любой полезной деятельности».*

Кроме этого, в его теоретической модели учтены знания (эрудиция) и благоприятная окружающая среда.

Разработанная Дж. Рензулли концепция предполагает, что контингент одаренных детей может быть значительно шире, чем при их идентификации по тестам интеллекта, креативности или достижений.

В последнее время многие исследователи склоняются к мысли, что ключевой характеристикой потенциала личности следует считать не выдающийся интеллект, как считалось ранее, а *мотивацию*. Так, например, некоторые ученые отмечают, что люди, изначально менее способные, но целенаправленно решающие собственную, лично значимую задачу, оказываются в конечном счете более продуктивными, чем более одаренные, но менее заинтересованные.

Максимально реализует свой потенциал, а следовательно, и достигает высот чаще не тот, кто был более развит, умен, а тот, кто был более настойчив, кто упорно шел к выбранной цели.

Представленная концепция одаренности дает ответы на вопросы о том, по каким параметрам выявлять одаренность, как прогнозировать ее развитие, на что ориентироваться в обучении и воспитании молодого поколения.

Из вышесказанного видно, что многие психологи в определении одаренности исходят из генетических предпосылок. При этом оговаривается, что только труд, деятельность может задатки трансформировать в способности, которые развиваются в одаренность при занятиях разными видами деятельности.

Дж. Рензулли же ввел несколько параметров для развития одаренности. На наш взгляд, наиболее важным является мотивационный. Люди, ориентированные на выполнение определенных видов деятельности, доводящие до конца начатое дело, настойчиво добивающиеся получения результатов, могут, безусловно, относиться к одаренным.

Компоненты одаренности по Дж. Рензулли

Рассмотрим подробнее определение одаренности, данное Дж. Рензулли. Исходя из этого, следует рассмотреть компоненты этого определения, а именно: интеллект, креативность и мотивация. Эти компоненты необходимо рассмотреть под углом понимания их роли в развитии и сопровождении одаренности школьников.

Что такое интеллект? Интеллект — это способности мышления. Он не дается от природы, его надо развивать в течение всей жизни [8]. Другими словами, под интеллектом подразумевается определенный уровень развития мыслительной деятельности личности, обеспечивающий возможность приобретать все новые знания и эффективно их использовать в ходе жизнедеятельности.

Что понимается под понятием креативность? «Креативность» происходит от англ. *creativity*, что означает уровень творческой одаренности, способности к творчеству. Первоначально креативность рассматривалась как функция интеллекта, и уровень развития интеллекта отождествлялся с уровнем креативности. Впоследствии выяснилось, что уровень интеллекта коррелирует с креативностью до определенного предела, а слишком высокий интеллект препятствует креативности. В настоящее время креативность рассматривается как несводимая к интеллекту функция целостной личности, зависящая от целого комплекса ее психологических характеристик [9]. Иными словами, креативность — это творческие способности индивида, характеризующиеся готовностью к принятию и созданию принципиально новых идей, отклоняющихся от традиционных или принятых схем мышления и входящие в структуру одаренности в качестве независимого фактора.

Что касается мотивации, то основа этого термина — мотив (от франц. *motive*). Означает побуждения к деятельности, связанные с удовлетворением потребностей субъекта. Совокупность внешних или внутренних условий, вызывающих активность субъекта и определяющих ее направленность (мотивация). Настойчивость — это волевое качество, направленное на неуклонное, вопреки трудностям и препятствиям, достижение осуществления цели [10].

Очевидно, что для развития одаренности школьников необходимо развивать перечисленные компоненты одаренности. Для разных возрастных групп школьников имеются, безусловно, необходимые для этого способы и методики. Здесь мы рассмотрим способы развития одаренности у школьников разных возрастных категорий.

Как было отмечено выше, компонентами одаренности являются интеллект выше среднего и креативность мышления (Дж. Рензулли). Для развития обеих характеристик очень важно развивать *логическое мышление*.

Логика (др.-греч. Λογική) — наука о правильном мышлении, искусство рассуждения. Логическое мышление — это мыслительный процесс, в котором человек пользуется четкими и конкретными понятиями. Другими словами, логика — наука здравомыслия, наука правильно рассуждать. Логическое мышление — это вид мыслительного процесса, при котором человек использует логические конструкции и готовые понятия [11].

Мыслить логически — это значит выделять важное, отделять его от второстепенного, делать выводы. Логика помогает находить обоснование многим ситуациям и явлениям, осмысленно оценивать факты, грамотно выстраивать свои суждения. Логическое мышление, как и любой другой навык, надо постоянно тренировать. Это важно в любом возрасте. Но если в программе обучения дошкольников и школьников обязательно присутствуют упражнения на развитие логического мышления, *то школьники старших классов и взрослые редко в повседневной жизни вспоминают о таких занятиях*.

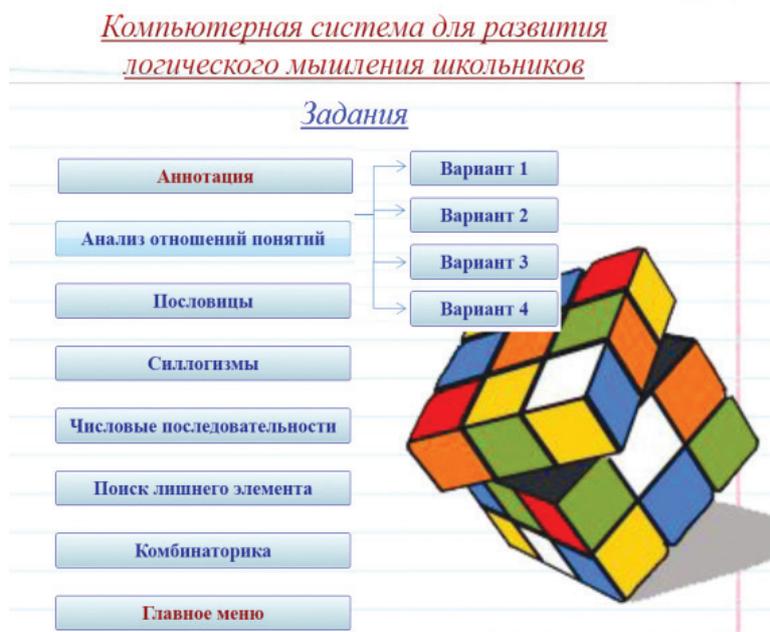
Электронный ресурс для развития логического мышления

Работа над разнообразными задачами и ситуациями (в теории, на практике) невозможна без использования следующих процессов мышления в различных комбинациях: анализ, синтез, сравнение, абстрагирование, конкретизация, обобщение.

Для развития логического мышления была разработана концепция компьютерной системы, а затем и реализована в рамках дипломного проекта студенткой ВКИ НГУ. Приложение предназначено для развития логического мышления у учащихся посредством решения различных логических задач, которые способствует развитию основных общелогических мыслительных умений, таких как:

- умение выделять существенные признаки из второстепенных;
- умение рассуждать, сравнивать, анализировать, классифицировать предметы;
- умение аргументировать свою точку зрения;
- умение устанавливать причинно-следственные связи;
- умение мыслить нестандартно.

В приложении имеются два режима работы: Разминка и Задания. Меню заданий имеет вид, представленный на рисунке.



Компьютерная система для развития логического мышления школьников

В таблице приведен тематический состав приложения, назначение отдельных заданий, количество заданий по каждой теме.

Состав заданий компьютерной системы для развития логического мышления школьников

№ п/п	Название	Назначение	Применение	Кол-во заданий
1	Анализ отношений понятий	Развитие абстрактных и логических связей	Для среднего и старшего звена учащихся	36 заданий

№ п/п	Название	Назначение	Применение	Кол-во заданий
2	Понимание переносного смысла пословиц и поговорок	Развитие абстрактного мышления, умения устанавливать и обобщать отношения между понятиями	Для любого возраста	10 заданий
3	Силлогизмы	Развитие аналитико-синтезирующих качеств мышления (абстрактное)	Для среднего и старшего звена	8 заданий
4	Числовые последовательности	Развитие логического мышления	Есть подсказки	10 заданий
5	Поиск лишнего элемента	Развитие качеств мышления по отвлечению и обобщению	Для младшего и среднего звена	10 заданий
6	Комбинаторика	Развитие логического мышления (процесс решения задачи «в уме»)	Для среднего и старшего возраста. Есть подсказки	11 заданий
ИТОГО:				85

В приложении имеется файл Статистика, в котором сохраняются результаты работы учащихся.

Заключение

Таким образом, в данной работе показан интерес общества к выявлению детской одаренности, как начального этапа формирования интеллектуальной элиты в государстве, кратко прослежена эволюция понятия «одаренность». Выявлено, что одаренность — это многокомпонентное понятие, включающее интеллект выше среднего, креативность мышления и целеустремленность (мотивацию). Развивая указанные компоненты школьников, мы можем надеяться получить на выходе из школы одаренных членов нашего сообщества. Для разных возрастных групп школьников необходимы, безусловно, особые способы и методики. Здесь мы рассмотрели один из способов развития одаренности. Этот электронный ресурс может быть использован для школьников разных возрастных категорий, указанных в таблице.

Список литературы

1. О Платоне. URL: https://studopedia.ru/1_28852_pedagogicheskie-idei-platona.html.
2. Понятие «одаренность». URL: http://studbooks.net/1384466/psihologiya/ponyatie_odaryonnost.
3. Генезис и история становления понятий «одаренность», «художественная одаренность». URL: <https://poisk-ru.ru/s12341t3.html>.
4. Детская одаренность. История развития. URL: https://revolution.allbest.ru/psychology/00333725_0.html.
5. Рабочая концепция одаренности Д.Б. Богоявленской, В.Д. Шадрикова. URL: http://studbooks.net/1614726/psihologiya/rabochaya_kontseptsiya_odarennosti_bogoyavlenskoy_shadrikova.

6. Кузнецова Ю.И. Трехкольцевая модель одаренности Дж. Рензулли. URL: https://superinf.ru/view_helpstud.php?id=5711.

7. Концепция одаренности Дж. Рензулли. URL: <http://spargalki.ru/psychologiya/115-psihiologia-odorennosti.html?start=10>.

8. Что такое интеллект человека. URL: <https://psihter.ru/teoriya/intellekt/>.

9. Общее понятие о креативности. URL: https://studbooks.net/1865263/pedagogika/obschee_ponyatie_kreativnosti.

10. Что такое мотивация: основные виды и характеристики. URL: <https://iklife.ru/samorazvitie/motivaciya-cto-eto-takoe-osnovnye-xarakteristiki-i-vidy.html>.

11. Филиппова К.А. Развитие логического мышления обучающихся средней школы на уроках математики // Молодой ученый. 2015 № 19. С. 622–624. URL: <https://moluch.ru/archive/99/22245>.

UDC 37.03

ELECTRONIC RESOURCES FOR DEVELOPING SCHOOLCHILDREN'S CAPABILITIES

K. V. Meleshko

Student, High College of Computer Science, Novosibirsk State University
630058, Novosibirsk, Russkaya st., 35

V. G. Perkova

Director, Regional IT Center,
630102, Novosibirsk, Voshod st., 26 a, e-mail: pvg@oblcit.ru

E. E. Saprykin

Deputy Director, Regional IT Center,
630102, Novosibirsk, Voshod st., 26 a, e-mail: go@oblcit.ru

G. A. Saprykina

Specialist, Regional IT Center,
630102, Novosibirsk, Voshod st., 26 a, e-mail: go@oblcit.ru

Abstract: The paper deals with the transformation of the concept of “giftedness” from ancient times to the present day. The description of the electronic resource for the development of logical thinking of students is given. Logical thinking is considered in the article as an important component in the development of giftedness.

Keywords: giftedness, intellect, creativity, motivation, thinking, logic

References

1. О Platone. URL: https://studopedia.ru/1_28852_pedagogicheskie-idei-platona.html О Platone.
2. Ponjatje «odarjonnost'». URL: http://studbooks.net/1384466/psychologiya/ponyatie_odaryonnost.
3. Genезis i istorija stanovlenija ponjatij «odarennost'», «hudozhestvennaja odarennost'». URL: <https://poisk-ru.ru/s12341t3.html>.

4. Detskaja odarennost'. Istorija razvitija. URL: https://revolution.allbest.ru/psychology/00333725_0.html.
5. Rabochaja koncepcija odarennosti D.В. Bogojavlenskoj, V.D. Shadrikova. URL: http://studbooks.net/1614726/psihologiya/rabochaya_kontsepsiya_odarennosti_bogoyavlenskoy_shadrikova.
6. *Kuznecova Ju. I.* Trehkol'cevaja model' odarennosti Dzh. Renzulli. URL: https://superinf.ru/view_helpstud.php?id=5711.
7. Koncepcija odarennosti Dzh. Renzulli. URL: <http://spargalki.ru/psihologiya/115-psihologia-odorennosti.html?start=10>.
8. Chto takoe intellekt cheloveka. URL: <https://psihter.ru/teoriya/intellekt/>.
9. Obshee ponjatie o kreativnosti. URL: https://studbooks.net/1865263/pedagogika/obshee_ponyatie_kreativnosti.
10. Chto takoe motivacija: osnovnye vidy i harakteristiki. URL: <https://iklife.ru/samorazvitie/motivaciya-chto-eto-takoe-osnovnye-xarakteristiki-i-vidy.html>.
11. *K. A. Filippova.* Razvitie logicheskogo myshlenija obuchajushhihsja srednej shkoly na urokah matematiki / K.A. Filippova; Molodoj uchenyj. 2015. №19. P. 622–624. URL: <https://moluch.ru/archive/99/22245>.

ОБУЧЕНИЕ ПАРАЛЛЕЛЬНЫМ ВЫЧИСЛЕНИЯМ НА РАННЕЙ СТАДИИ ИЗУЧЕНИЯ ПРОГРАММИРОВАНИЯ

И. Н. Скопин

канд. физ.-мат. наук, доц., Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 2,
ст. науч. сотрудник, Институт точной механики и математической геофизики СО РАН,
630090, Новосибирск, пр. Акад. Лаврентьева, 6, e-mail: iskopin@gmail.com

Аннотация. Изучение методов параллельного программирования обычно происходит после того, как обучаемый освоил методы последовательного составления программ на весьма высоком уровне. Однако из-за уже сложившихся стереотипов мышления такой подход затрудняет понимание проблем параллельного программирования и, в частности, того, как взаимодействуют и синхронизируются параллельно исполняемые процессы. Предлагается специальная методика изучения параллелизма еще до того, как обучаемый овладевает базовыми методами последовательного программирования. Она адаптирована к преподаванию программирования на ранних курсах вузов математического и технического профиля.

Ключевые слова: параллельное программирование, методика обучения, событийное программирование, взаимодействие процессов, синхронизация

Введение

Преподавание программирования сегодня исходит из того, что современному молодому технически грамотному человеку достаточно знать основные принципы построения алгоритмов и отображения программного представления алгоритмов в вычислительные средства, предоставляемые оборудованием. Тот факт, что занятие программированием формирует особый склад мышления, остается на заднем плане, а вместо поддержки этого процесса предлагается осваивать серии последовательно усложняющихся шаблонов. Таким образом, вместо развития способностей к творчеству ученика привыкает думать шаблонно.

В работе [1] мы показали перспективность задач, при решении которых требуется не поиск известного ответа, а решение осознанной проблемы, для мотивации обучаемых к творчеству. Предложенный подход разумно применять при обучении параллельному программированию. Эта учебная тематика становится все более актуальной в связи с прогнозируемым ростом потребностей в программистах, способных разрабатывать сложные программные продукты, использующие модели вычислений с огромным числом процессоров, для которых существующие решения оказываются неэффективными.

Не так давно мы предложили идею раннего обучения параллельному программированию [2], которая была частично проверена экспериментально. К сожалению, несмотря на заметную эффективность нашего подхода, внедрить его в массовый учебный процесс сегодня нереально, а потому приходится приспособливать методику к обучению тех, кто уже имеет представление

об информатике и первоначальный опыт программирования. Это студенты младших курсов с математической, физической и технической специализацией. Опыт преподавания тем, связанных с параллелизмом, студентам второго курса механико-математического факультета НГУ показывает продуктивность адаптации методики раннего обучения параллельному программированию к этим условиям. Для предметного обсуждения нашего подхода далее мы приводим верхний уровень тематик, предлагаемых нашим студентам в связи с изучением параллелизма.

1. Последовательные, параллельные и совместные вычисления

Материал раздела на простых примерах показывает вариативность реализации программ на оборудовании, имеющем несколько процессоров, и освещает проблемы компилятора, связанные с трансляцией программ в разных вычислительных обстановках.

2. Разработка параллельных программ

Вводятся понятия распараллеливания последовательной программы и конструирования программ, разрабатываемых специально для параллельных расчетов. Изложение материала дается в стиле демонстрации возможных проблем параллелизма при конструировании простой программы (по-видимому, первым, кто использовал такой стиль, был Дейкстра [3]).

В качестве основы изложения рассматривается пример, который демонстрирует возможные решения проблемы взаимоисключающего выполнения процессов разными способами в условиях наличия и отсутствия централизованного управления процессами. Обсуждаются семафоры Дейкстры, предложенные им для согласования параллельных процессов, а также алгоритм Деккера, реализующий синхронизацию без использования специальных языковых конструкций, а также без аппаратных средств поддержки.

3. Порождение параллельных процессов и передача сообщений между процессами

В качестве развития примера из раздела 2 мы приводим систему понятий, связанную с потоковым параллелизмом. В результате обучаемые готовятся к пониманию различных подходов к организации взаимодействия процессов, синхронизации, а также к механизмам событий, которые разбираются в следующем разделе.

4. Потоковый параллелизм

Вводятся понятия потоковых вычислений и описываются методы их реализации: механизм событий, блокирующие и неблокирующие прием и передача сообщений, понятие приоритетов реакций на происходящие события, метод, реализующий взаимодействие поставщиков и потребителей данных. Каждый из подходов обсуждается с точки зрения принципиальной применимости, неприменимости и целесообразности применения в различных ситуациях. Как и ранее, изложение сопровождается демонстрационными примерами.

5. Специализированные параллельные вычислительные системы

Тематика раздела очень многообразна, а потому выбор того или иного варианта изложения остается за преподавателем. Тем не менее обучаемые должны получить представление о том, для чего разрабатываются специализированные модели параллельных вычислений, какие критерии качества выставляются для оценки предлагаемых решений.

6. Классификация архитектур вычислительных систем

Излагается классификация архитектур систем, основой которой служат базовые решения, определяющие реализуемую модель вычислений в общих чертах. Она служит для определения того, для каких типов вычислений целесообразно использовать то или иное оборудование.

7. Поддержка распределенной и разделяемой памяти

Программирование параллельных программ всегда должно соответствовать модели вычислений, определяемой архитектурой оборудования, которое используется в расчетах. В этом отношении важно с самого начала понять, может ли задача ограничиваться при взаимодействии процессов эпизодическими обменов информацией, и тогда для ее решения достаточно использование вычислительной системы с процессорами с автономным адресным пространством, т. е. архитектуры с распределенной памятью. Альтернатива — когда для задачи нужен совместный доступ параллельно работающих процессов к общей разделяемой между процессорами памяти. Знание этих вариантов необходимо обучаемому для определения того, какие инструменты поддержки программирования потребуются.

8. Квазипараллельные системы

Параллельное программирование всегда приводит к недетерминированному выполняемым программам. Это связано с тем, что в общем случае нельзя точно знать время, которое потребуется для выполнения процессов, и порядок, в котором они будут выполняться. Поэтому приходится вводить в программу специальные действия, обеспечивающие синхронизацию, непосредственно с логикой программы не связанные. Недетерминизм не обязательно означает, что разные запуски на выполнение параллельной программы с одними и теми же данные приводят различным результатам. Если для программы такое возможно, то, как правило, она считается некорректной. К сожалению, распознать некорректность параллельной программы намного сложнее, чем для последовательной программы, и именно потому, что из-за недетерминизма ход параллельных вычислений обычно предсказать невозможно. И эта одна из проблем, связанных с параллелизмом, в частности, с отладкой. Другая проблема — понимание трудностей параллельного программирования, особенно теми, кто сталкивается с недетерминизмом впервые.

В обучении очень важно, чтобы у ученика была возможность повторять ситуации, возникающие при выполнении программы. Иными словами, требуется не параллельное выполнение, а его имитация с детерминированным поведением. При хорошо организованной имитации ею можно управлять: приостанавливать вычисления, выбирать тот или иной вариант выполнения программы, повторять расчеты и др. Тогда ученик в состоянии видеть эффекты, которые могут проявиться при реальном параллельном счете, и, соответственно, корректировать принимаемые решения. Эти же средства управляемой имитации полезны и при отладке параллельной программы, в частности, для выявления критических ситуаций с вероятным нарушением синхронизации, гонки данных и других случаев некорректного поведения программы.

Во многих случаях при разработке параллельной программы ее потребность в параллелизме превосходит доступные процессорные ресурсы. Эта ситуация характерна, в частности, для задач моделирования. Как и в предыдущих случаях, здесь возникает потребность в управляемом детерминизме, т. е. в имитации параллелизма, полностью или частично подменяя его последовательными вычислениями. Такие системы называют квазипараллельными (от лат. слова *quasi* — «якобы», «почти»), что подчеркивает генетическую связь логически параллельной программы с ее последовательным вариантом выполнения. Подходы, используемые при разработке квазипараллельных систем, различаются в зависимости от их целей.

Для демонстрации таких различий в учебном курсе мы предлагаем два в некотором смысле противоположных примера квазипараллельных систем. Первый из них основан на методах деления процессорного времени между процессами, а второй связан с построением так называемых систем с дискретными событиями.

1. *Разделение времени.* Этот механизм имитирует параллельное выполнение процессов путем выделения каждому процессу кванта времени для вычислений, исчерпание которого

приводит к приостановке процесса и активизации другого на время, определенное его квантом. При этом образуются очереди процессов на выполнение. Варианты организации таких очередей и правил постановки процесса в очереди связываются с принимаемой стратегией распределения процессорных ресурсов по задачам и с введением приоритетов процессов.

2. *Системы с дискретными событиями.* При моделировании требование детерминированности расчетов может оказаться решающим, и в таких случаях вместо частичной и полной линеаризации процессов, основанной на разделении времени, оказывается более предпочтительным использование так называемых систем с дискретными событиями. Наиболее развитая реализация этого подхода была предложена в языках Simula и Simula 67 [4]. Она провозглашает явную управляемую имитацию параллелизма и, как следствие, оставляет для программиста возможность думать о выполнении программы в терминах взаимного влияния параллельных процессов друг на друга и, вместе с тем, составлять программы, которые, ограничиваясь строго последовательными вычислениями, имитируют параллелизм выполнения процессов. Это общий метод преодоления противоречия между принципиальным параллелизмом и реально последовательной его реализацией, который оказывается очень естественным способом выражения многих алгоритмов (соответствующие примеры можно найти в работах [6] и [7]).

Заключение

Представленный обзор программы преподавания параллелизма показывает содержание, которое обучаемые осваивают по данной тематике на основе лишь начальных сведений из курсов программирования и информатики. Методически более эффективным было бы обучение параллелизму до формирования привычки последовательного программирования (обоснование этого подхода представлено в работах [2] и [6]). Однако сегодня массовое преподавание программирования и информатики традиционно связывается с шаблонами, заранее разработанными для последовательных алгоритмов. В результате ученик может даже не задумываться о том, почему и как работает его программа. Применительно к параллелизму такая методика обучения крайне неэффективна: без понимания проблем параллельного программирования и методов их преодоления нет никакой возможности конструировать программы, отвечающие естественным критериям качества. Наш подход основывается на постановке для обучаемых проблемных задач [1], требующих анализа, для чего создаваемая программа будет использоваться, как она будет работать в среде с ограниченными ресурсами, а также иных условий ее применения.

Сегодня порядок подготовки специалистов в области программирования и информатики опирается лишь на развитие шаблонного мышления, что не способствует формированию привычки ставить и решать проблемные задачи. В то же время сфера параллелизма требует именно этого. Следовательно, приходится адаптировать подход раннего обучения параллельному программированию к сложившимся условиям, т. е. предлагать паллиатив и надеяться на изменение ситуации в дальнейшем.

Оформление предлагаемого материала в виде учебного курса требует среди прочего подготовку задач, решение которых нацелено не только на усвоение тематики, но и на формирование творческого мышления. Достижению этой цели способствует создание мотивации у учащегося к анализу условий, в которых будет работать создаваемая программа. Мы хорошо осознаем важность решения проблемы подбора учебных задач, а потому не хотели бы затрагивать ее вскользь и оставляем данный аспект курса обучения параллельным вычислениям в качестве направления дальнейшей работы.

Список литературы

1. Скопин И. Н. Проблемные задачи при изучении общих методов информатики и программирования // Вестн. Рос. ун-та дружбы народов. Серия : Информатизация образования. 2011. № 4. С. 21–33.
2. Скопин И. Н. Раннее обучение параллельному программированию // Вестн. Моск. гор. пед. ун-та. Серия : Информатика и информатизация образования. № 2 (22). М. : МГПУ, 2011. С. 46–55.
3. Дейкстра Э. Взаимодействие последовательных процессов // Языки программирования / ред. Ф. Женюи; пер. с англ. В. П. Кузнецова; под ред. В. М. Курочкина. М. : Мир, 1972.
4. Дал У. И., Мюрхауз Б., Ньюгорд К. Симула-67. Универсальный язык программирования. М. : Мир, 1969.
5. Скопин И. Н. О формировании программистского мышления // Бюллетень Моск. гор. пед. ун-та, Лаборатория математического, естественнонаучного образования и информатизации. Т. 2. М. : Изд-во «Научная книга», 2012. С. 31–41.
6. Skopin I. N. Early Learning in Parallel Programming // In. NOVA science publishers. Series : Mathematics Research Developments / ed. M. S. Tarkov. Chapter 10, 2014.
7. Skopin I. N. An Approach to the Construction of Robust Systems of Interacting Processes// In. NOVA science publishers. Series : Mathematics Research Developments / ed. M. S. Tarkov. Chapter 9, 2014.

UDK 004.42

TUITION OF PARALLEL COMPUTING AT EARLY STAGES OF LEARNING PROGRAMMING

I. N. Skopin

Ph.D., Associate professor, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova st., 2,

Senior Research Scientist, Institute of Computational Mathematics and Mathematical Geophysics SB RAS
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: iskopin@gmail.com

Abstract. The study of methods of parallel programming usually occurs after the student has mastered the methods of sequential programming at a very high level. However, due to the already established stereotypes of thinking, such an approach makes it difficult to understand the problems of parallel programming and, in particular, how processes that are executed in parallel interact and synchronize. Here a special approach to studying parallelism is proposed. It may be used even before the student masters the basic sequential programming methods. We adapted the approach to the teaching of programming in the early courses of universities of mathematical and technical profile.

Keywords: parallel programming, teaching methods, event driving programming, process interaction, synchronization.

Referenses

1. Skopin I. N. Problemnyie zadachy pri izuchenii obshchih metodov informatiki i programmirovaniya // Vestn. Ross. Un-ta Druzhby Narodov. Seriya: Informatizatsiya obrazovaniya. № 4, 2011. P. 21–33. (in Russian)

2. *Skopin I. N.* Ranneye obuchenie parallelnomu programmirovayu // Vestn. Mosk. gor. ped.o un-ta. Seriya: Informatika i informatizatsiya obrazovaniya. № 2 (22). M. : MGPU, 2011. P. 46–55. (in Russian)
3. *Dijkstra E. W.* Vzfidodeystvie posledovateknyh protsessov // In. Iazyki programmirovaniya. Ed. F. Genuys. Trans. From English V. P. Kuznttsov, ed. V. M. Kurochkin. M. : Mir, 1972. (in Russian)
4. *Dahl O. J., Myhrhaug B., Nugaard K.* Simula-67. Universaknyjq Jazyk Programmirovaniya. Trans. From English. M. : Mir, 1969. (in Russian)
5. *Skopin I. N.* O formirovanii programmistskogo myshleniya // Bulletin Mosk. gor. ped. un-ta, Laboratoriya matematicheskogo, estestvennonauchnogo obrazovaniya i informatizatsii. Vol. 2. MGPU, M.: Izd. Nauchnaya kniga, 2012. P. 31–41. (in Russian)
6. *Skopin I. N.* Early Learning in Parallel Programming // In. NOVA science publishers. Series : Mathematics Research Developments / ed. M. S. Tarkov. Chapter 10, 2014.
7. *Skopin I. N.* An Approach to the Construction of Robust Systems of Interacting Processes// In. NOVA science publishers. Series : Mathematics Research Developments / ed. M. S. Tarkov. Chapter 9, 2014.

ТРАДИЦИИ РАБОТЫ С ОБУЧАЮЩИМИСЯ ИТ-ТЕХНОЛОГИЯМ

Т. И. Тихонова

науч. сотрудник, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, пр. Акад. Лаврентьева, 6,
ст. преподаватель, Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 2, e-mail: tanja@iis.nsk.su

Аннотация. В статье рассматривается работа со школьниками на летней школе юных программистов. Развитие научно-исследовательского потенциала детей организовано в виде мастерских. Привлечение к творческой деятельности посредством выполнения исследовательской работы с практической реализацией в виде проекта, выступления с докладом на научно-практической конференции формирует активную жизненную позицию, социализирует личность и закладывает качества, необходимые для профессиональной деятельности.

Ключевые слова: информационные технологии, проектная деятельность, образование, творчество

В стране на сегодняшний день отдельное внимание уделяется дополнительному образованию. Решение кадрового вопроса для формирования грамотного научного и инженерного состава как институтов, так и предприятий — задача современная и своевременная. Надо отметить, что новосибирская школа программирования всегда стояла на передовых позициях раннего вовлечения в профессиональную сферу деятельности как студентов, так и школьников. То, что в прежние времена не требовало рекламы, сегодня рассматривается с несколько иных позиций: дать инструменты и технологии самые передовые, востребованные, перспективные. Индустриальная держава, которая очень нуждается в высококвалифицированных кадрах в области информационных технологий, должна заниматься кадрами, начиная со школьной скамьи. Особую роль в этом играет дополнительное образование, общение со специалистами, которые решают проблемы на научном, производственном и др. направлениях деятельности.

Вузы сегодня также страдают от недостаточной мотивации студентов. Осознанное поступление в тот или иной университет, на определенную специальность — задача, которая не каждому выпускнику под силу.

ЛШЮП [1] помогает определиться с будущей профессией осознанно, на основе опыта практической деятельности, взаимодействия с научными кадрами и студентами, аспирантами, выпускниками НГУ и других вузов страны, которые рассказывают об особенностях как образовательных учреждений, так и получаемых квалификаций в них. Для НГУ и, в частности, для кафедр «Программирование» и «Вычислительные системы», а также ВКИ и СУНЦ НГУ проведение ЛШЮП — это «обкатка» методик раннего обучения современной информатике; привлечение в НГУ абитуриентов, интересующихся программированием,

способных в будущем участвовать в конкурсах и научных проектах; рост профессионального уровня студентов ВКИ и СУНЦ НГУ; привлечение к преподавательской деятельности студентов, приобретение студентами навыков работы в качестве руководителей проектов и постановщиков задач.

Одним из эффективных механизмов, направленных на раннее включение обучающихся в IT-деятельность, является выездная Летняя школа юных программистов. Первая Летняя школа юных программистов была организована в далеком 1976 г. замечательными учеными-энтузиастами под руководством академика Андрея Петровича Ершова. За истекший период подготовку в ЛШЮП прошло несколько тысяч ребят, для которых программирование стало делом жизни.

В областных школах ярко видна проблема кадров. Проведение Летних школ юных программистов для администрации Новосибирской области представляет интерес в части разработки механизма выездной работы со школьниками, изучающими информатику. Этот механизм может быть распространен на сельские районы, способствует повышению уровня подготовки сельских участников в вузы. Также в Новосибирск привлекаются иногородние школьники, которые получают возможность приехать в последующем в качестве абитуриентов в новосибирские вузы.

43-я ЛШЮП им. А. П. Ершова в 2018 г. была проведена с 28 июня по 11 июля в Доме отдыха «Спартанец» в сосновом бору на берегу Бердского залива. Открытие состоялось в 11.00 в Новосибирском Академгородке в малом зале Дома ученых. На открытии ЛШЮП приветствовали ребят и высказывали теплые слова напутствия. От Сибирского отделения ребят приветствовал академик Павел Владимирович Логачев, директор Института ядерной физики. Ректор НГУ, д-р физ.-мат. наук Михаил Петрович Федорук, заметил, что это мероприятие является историческим для Академгородка. От управления Детским Технопарком Татьяна Андреевна Колесникова пожелала всем плодотворной работы и хорошего отдыха на природе. Профессор Сергей Константинович Водопьянов, директор «Математического центра» НГУ, д-р физ.-мат. наук выразил уверенность, что участники Летней школы юных программистов станут достойными студентами. Он отметил, что НГУ является лучшим вузом. Директор ИСИ СО РАН Александр Гурьевич Марчук, являющийся председателем оргкомитета ЛШЮП, уже в 18-й раз на Летней школе ведет мастерскую. Он напомнил, что важно не только получить профессию, но и развивать в школьниках творческое начало, чем и занимается ЛШЮП. В этом году участвовали школьники, студенты и преподаватели из Абакана, Бердска, Кольцово, Новокузнецка, Новосибирска, Миасса, Москвы, Оби, Омска, Санкт-Петербурга, Саратова...

Для отбора учащихся проводится несколько собеседований. Новосибирские школьники проходят очное собеседование в виде решения задач и объяснения решений. Подбор задач включает в себя ранжированные по уровню задачи как алгоритмические, так и на логику и математику. Отбор иногородних участников проходит в виде собеседования по Скайпу. Для участия в работе ЛШЮП были приглашены учащиеся 4–11-х классов, отобранные Оргкомитетом по собеседованию. Многие из ребят зарекомендовали себя в различных мероприятиях: олимпиадах, конференциях, конкурсах, проводимых в течение учебного года. Многие из школьников участвовали в ЛШЮП этого года не в первый раз. Летняя школа формирует задел для значительного потенциального роста ребят. В течение учебного года они начинают активно участвовать в различных конкурсах и конференциях, более успешно выступают на олимпиадах, проходят отбор в специализированные классы, записываются в кружки и ходят на спецкурсы. По настоятельной просьбе приходится организовывать такие занятия силами наших мастеров на базе ИСИ СО РАН, Детского технопарка.

Для сохранения традиций [2] Летней школы приходится ограничивать набор. Организаторы стараются сохранить атмосферу ЛШЮП. Сегодня мы обращаемся к технологиям дистанционного обучения, новым методам и средствам. Безусловно, и техника, и методы, и содержание образования стало шире и разнообразнее. Но сама ЛШЮП, позволяющая на две недели собрать энтузиастов-преподавателей, талантливых школьников, которые погружаются в обстановку творческого занятия любимым программированием в кругу единомышленников, обладает бесспорными достоинствами и результатами. В этом году в ней приняли участие порядка 110 человек (85 школьников 4–11-х классов, четыре организатора (директор, завуч, сисадмин, ответственный за питание и быт), 14 мастеров, приезжающие лекторы).

Традиционное разнообразие тематик мастерских на ЛШЮП по численности составляло 14 (одна из мастерских была «сдвоенной» по тематике выполняемой работы). В каждой из мастерских были выполнены практические реализации проектов по заявленным темам. Некоторые темы в мастерских получили интересный результат: «Обработка больших данных», «Умный дом», «Квантовые вычисления», «Генетические алгоритмы и нейросети», «Создание языков программирования», «Работа с кластером, суперкомпьютеры» и др.

Следуя сложившейся традиции, программой ЛШЮП предусматривается проведение обзорных и учебных лекций, семинаров и конкурсов по проблемам современной информатики и другим научным направлениям. В этом году лекционная программа, как обычно, была насыщенной и интересной. Лекцию по физике ускорителей прочитал академик Василий Васильевич Пархомчук.

Сегодняшний день требует от программистов не только специальных компетенций, но и знаний из смежных областей. Например, бурно развивается биоинформатика. В этом направлении была прочитана лекция д-р биол. наук, проф. НГУ Григорием Моисеевичем Дымшицем. Из области программирования канд. физ.-мат. наук, доц. НГУ Зинаида Владимировна Апанович рассказала школьникам о визуализации графов. Лекцию нашего постоянного приглашенного участника, канд. физ.-мат. наук, доц. Саратовского политехнического университета о криптографии, Бориса Леонидовича Файфеля, удалось даже записать и выложить в сети Новосибирского ОблЦИТ (видео лекции Б.Л. Файфеля «Популярно о криптографии» ЛШЮП-2018: <http://edu54.ru/video/136620/>).

К сожалению, мы не всегда помним, как все начиналась. В бурном времени информационных технологий забываются имена и даты. Посему в рамках лекционной программы мы посчитали необходимым ввести историческую линию. Лекцию об Андрее Петровиче Ершове прочитала канд. ист. наук Крайнева Ирина Александровна. И другие лекции были интересны и востребованы. Лекции по запросу школьников о криптовалюте проф. А. Г. Марчука, к сожалению, даже не нашлось времени в учебной сетке. Школьники и студенты, а также все «взрослые» мастера, помимо работы в мастерских, с удовольствием и интересом поучаствовали в лекционной программе ЛШЮП, которая состояла не только из специализированных вопросов технологий программирования. Традиционно не только лекции понравились слушателям, но и лекторы очень тепло отзывались об аудитории ЛШЮП-цев. [3]

Основными задачами ЛШЮП является отбор талантливых старшеклассников, заинтересованных в овладении профессиональным программированием, обучение учеников среднего звена навыкам коллективной работы с применением современных информационных технологий и содействие развитию способностей к практическому программированию учащихся младших классов, а также поддержка педагогов, успешно преподающих информатику и программирование в общеобразовательной системе.

Спецификой этого года организаторы считают очередное снижение возрастного барьера. К сожалению, несмотря на огромное желание родителей и детей, неподдельный интерес к компьютеру как инструменту, учащиеся 3–4-х классов редко бывают подготовлены техниче-

ски для методически отработанных механизмов практической проектной деятельности в рамках ЛШЮП. Большинству из них явно не достает математического аппарата и вычислительных навыков. Психологию детства даже не берем в основной расчет. Потому организаторы осуществляют осознанно отбор учащихся — участников Летней школы — преимущественно среднего звена.

Неоднократное участие в ЛШЮП позволяет подготовить к серьезной проектной деятельности. Превалирует по количеству 9-й класс. Это обусловлено необходимостью приобщения детей к коллективной работе, пропедевтическая работа по изучению основ профессиональной деятельности, а также возможность пролонгированной работы со школьниками. 10-ти и 11-тиклассники, как правило, уже определились если не со своей будущей специальностью, то с направлением деятельности. Деятельность осуществляется через знакомство с программированием как с производственной задачей, с его проблематикой, методологией, творческими и технологическими аспектами. Новыми понятиями и объектами для изучения становятся программный продукт, технологический процесс разработки, грамотная постановка задачи и ее формализация, рациональное распределение и планирование работ, отладка, оформление, документирование, отчет.

Совокупность мастерских в Школе подбирается так, чтобы как можно лучше обеспечить многопрофильность и разный уровень учебного процесса с целью более адекватной его настройки на базовую подготовку, индивидуальные наклонности, интересы и способности учащихся.

Расписанием работы Летней школы были предусмотрены 1 день — для заезда заранее технических работников с целью открытия компьютерных классов Летней школы и распределения работ, формирования сети. 1 день — для заезда, открытия, презентации мастерских и окончательного формирования списка состава мастерских, 1 день — для закрытия ЛШЮП и отъезда участников, 11 учебных дней, 1 день заключительных отчетов мастерских на конференции и демонстрации выполненных проектов, 1 выходной день.

Начинать практическую работу над проектами удалось практически с первого учебного дня, благодаря предварительной подготовке техники, установки образов, настройки сервера и т. п. Открытие проведено до отъезда в торжественной обстановке. По традиции была сделана попытка памятной фотографии на ступеньках. Предварительное распределение по мастерским проводилось до начала ЛШЮП. После презентации мастерских по приезду в первый день проводились корректировка списочного состава мастерских и вводные занятия. Исходя из опыта прошлых лет, на место проведения ЛШЮП технику вывозили, устанавливали и монтировали сеть заранее, за день до начала учебного процесса на ЛШЮП.

Летняя школа юных программистов проводилась в 43-й раз. Несмотря на столь длительный этап существования, она остается востребованной и, в некотором роде, уникальной не только в Сибирском федеральном округе, но и в стране в целом. Ежегодно в связи со сложным поиском финансовой поддержки проведения ЛШЮП организаторы пытались сдерживать численность состава порядка 100 человек. В этом году активный рост желающих участвовать в работе ЛШЮП юных программистов способствовал рассмотрению вопроса об увеличении численного состава. ЛШЮП приняла в свои ряды участников на 30 % больше. Безусловно, это усложнило весь жизненный цикл, начиная от поисков финансов до решения технических проблем: организация транспорта, питания и проживания, увеличение преподавательского состава и т. д. Тем не менее Оргкомитет и преподаватели справились достойно, впрочем, как всегда. Работы, выполненные в мастерских тому доказательство.

Как уже было сказано, одна из целей Летней школы юных программистов — дать почувствовать учащимся технологию разработки. Опыт по обучению программистов технологии не богат в столь юном возрасте, взгляды на подходы значительно разнятся. Потому в мастерских

значительное время уделяется ретроспективе, осмыслению проделанной работы. Содержание этапов проектирования, макетирование, отладка, тестирование и составление документации показывает тесное взаимодействие в коллективе всех участников, выполняющих проект. Благодаря совместному анализу участников мастерской и преподавателей неудачи превращаются в достижения.

В ходе реализации проекта было выявлено несколько проблем. Одна из них, не зависящая от организаторов, базируется на проблеме поиска финансирования. ЛШЮП проводится как выездная школа, потому надо обеспечить оплату питания и проживания преподавательского состава, даже если оплата путевки в детское оздоровительное учреждение осуществляется за счет других источников (в том числе привлекаются родительские средства). Вторая проблема в связи с экономией финансовых средств — ограничение количества преподавательского состава. В течение всей ЛШЮП преподаватели (мастера) совмещают работу по контролю учащихся во время отдыха, сна, внеучебных мероприятий. С одной стороны, тесное взаимодействие во время мероприятий плана викторин, КВН, творческого диспута и т. п. помогает сплочению коллектива. С другой стороны, очень разгрузили бы работу мастеров несколько специально подготовленных преподавателей, выполняющих организационно-воспитательную работу (например, контроль за отбоем, проведение зарядки, культурно-массовые мероприятия).

Безусловно, присутствие интереса к пропедевтике программирования (раннее включение молодежи в процесс освоения информационных технологий) дает высокую эффективность. С другой стороны, требует высокой бдительности и контроля за пребыванием школьников на выезде. Летняя школа юных программистов — это не только новые знания, но также новые друзья и просто хорошее настроение. Помимо получения знаний, юные программисты расширяют круг общения, найдут единомышленников и будут развивать творческие способности для достижения высоких результатов не только в профессии, но и в жизни.

Список литературы

1. *Тихонова Т.И.* Отчет о работе новосибирской Летней школе юных программистов 2014 // Системная информатика. 2015. № 5. С. 75–104.
2. *Марчук А.Г., Тихонова Т.И.* Традиции в системе подготовки творческой молодежи // Компьютерные инструменты в школе. № 2. СПб., 2008. С. 3–11.
3. *Тихонова Т.И.* Становление личностных качеств программиста : сб. докладов и тезисов 9-й Ершовской конференции по информатике. Рабочий семинар «Информатика образования» (PSI14). 2014. С. 70–72.
4. *Марчук А.Г., Тихонова Т.И.* Летняя школа юных программистов — этап становления школьной информатики : сб. науч. трудов Междунар. науч.-практич. конф. «От информатики в школе к техносфере образования». М., 2016. С. 243–249.

UDK 37.018.536''322: 004.9, 004.438, 37.047

EFFECTIVE MECHANISMS FOR WORKING WITH IT LEARNERS

T. I. Tikhonova

Researcher, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentiev pr., 6,
Senior Lecturer, Novosibirsk State University,
6300906, Novosibirsk, Russia, Pirogova St., 2, e-mail: tanja@iis.nsk.su

Abstract. The article deals with work with students at the summer school for young programmers. The development of the research potential of children is organized in the form of workshops. Attraction to creative activity through the implementation of research work with practical implementation in the form of a project, making a presentation at a scientific and practical conference forms an active life position, socializes the personality and lays down the qualities necessary for professional activity.

Keywords: information technology, project activities, education and creativity

References

1. *Tikhonova T. I.* Otchet o rabote novosibirskoj Letnej shkole junyh programmistov 2014 // *Sistemnaya informatika*. 2015. № 5. P. 75–104. (in Russian)
2. *Marchuk A. G., Tikhonova T. I.* Tradicii v sisteme podgotovki tvorcheskoj molodezhi // *Komp'yuternye instrumenty v shkole*. № 2. SPb., 2008. P. 3–11. (in Russian)
3. *Tikhonova T. I.* Stanovlenie lichnostnyh kachestv programmista : sb. dokladov i tezisov obrazovanija» (PSI14). 2014. P. 70–72. (in Russian)
4. *Marchuk A. G., Tikhonova T. I.* Letnjaja shkola junyh programmistov — jetap stanovlenija shkol'noj informatiki : sb. nauch. trudov Mezhdunar. nauch.-praktich. konf. «Ot informatiki v shkole k tehnosfere obrazovanija». M., 2016. P. 243–249. (in Russian)

НАУЧНО-ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ ШКОЛЬНИКОВ КАК МОТИВАЦИЯ ИССЛЕДОВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ

Т. И. Тихонова

науч. сотрудник, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, пр. Акад. Лаврентьева, 6
ст. преподаватель, Новосибирский государственный университет,
630090, Новосибирск, ул. Пирогова, 2, e-mail: tanja@iis.nsk.su

Аннотация. Развитие исследовательского подхода к любой деятельности важно начинать в раннем возрасте. Любознательность является природным качеством человека. В статье рассматривается работа со школьниками, направленная на развитие научно-исследовательского потенциала детей. Привлечение к творческой деятельности посредством выполнения исследовательской работы с практической реализацией в виде проекта, выступления с докладом на научно-практической конференции (НПК) формирует активную жизненную позицию, социализирует личность и закладывает качества, необходимые для профессиональной деятельности.

Ключевые слова: информационные технологии, проектная деятельность, образование, творчество

Введение

В Новосибирске научно-практическая работа со школьниками является традиционной. Являясь не только индустриальным и аграрным комплексом, но еще и уникальным городом, благодаря построенному Академгородку, Новосибирск активно растит научные кадры. В этом направлении работает СУНЦ НГУ, ВКИ НГУ, вузы города и академические институты, которые постоянно взаимодействуют с талантливыми школьниками, осуществляющими выбор профессии. Для учащихся проводятся различные конкурсы, олимпиады, турниры, в том числе городские научно-практические конференции школьников, летние школы [1]. Мероприятия проводятся в целях создания условий для выявления и поддержки детей, одаренных в области научного творчества, развития научного потенциала школьников.

Педагогические основы

Великие педагоги Л. С. Выготский [2] и Я. Каменский [3] говорили не только о необходимости самообразования, но и о социальном аспекте обучения. Учиться ремеслу надо не только самостоятельно, но и в социуме посредством взаимодействия с другими людьми, учителями, коллегами.

Социальная ситуация развития — это «совершенно своеобразное, специфическое для данного возраста, исключительное, единственное и неповторимое отношение между ребенком и окружающей его действительностью, прежде всего социальной». Л. С. Выготский предлагал рассматривать социум в аспекте воспитания и обучения, для формирования деятельностной активности в совокупности с сопровождающими переживаниями. Ян Каменский в своих трудах писал: «Умственное воспитание — это познание самого себя и окружающего мира сообразно природе». Опираясь на возрастные группы, выделенные Выготским, можно сказать, что некоторые дети уже с начального школьного возраста способны к рефлексии. К старшему школьному возрасту необходимо сформировать не только профессиональное самоопределение, но и осознание себя в обществе [4], поддержанное умением общаться как со сверстниками, так и с другими возрастными категориями.

Знания, поддержанные умением и компетенциями, способствуют формированию у детей научной картины мира, стимулируют у школьников формирование интереса к фундаментальным и прикладным наукам. Очень позитивно сегодня осознание того, что молодежь хочет достичь понимания происходящих в природе процессов. Понимание принципов помогает перешагнуть через незнание деталей. Развитие творческой исследовательской активности детей дает умение ассоциативно видеть связи между предметом исследования и моделью. Получить представление об исследовательском обучении как ведущем способе учебной деятельности полезно не только детям, но и педагогам.

Как нельзя кстати такому роду деятельности соответствует участие в научно-практической деятельности с последующим докладом о проделанной работе для единомышленников.

Мероприятия

В 2019 г. НПК проводилась уже в 38-й раз. Учредитель конференции — Главное управление образования мэрии города Новосибирска. Организатор конференции — Дворец творчества детей и учащейся молодежи «Юниор». Мероприятие проводится в тесном сотрудничестве с вузами города и научно-исследовательскими учреждениями СО РАН с привлечением специалистов из НГПУ, НГУ, НГТУ, СибГУТИ и других вузов, а также представителей СО РАН по специальностям, представленным на секциях НПК.

Конференция проводится для учащихся, проявляющих способности к исследовательской деятельности в различных областях знаний. Секции, представленные на ней, отличаются многообразием, которое может представить поле деятельности практически в любой области знаний: математика, физика, информационные технологии, языкознание, лингвистика, история, культурология и т. д. Школьникам создаются условия для роста личностного и интеллектуального потенциала. Участие в научно-практической конференции воспитывает качества, необходимые для успешной самореализации.

Традиционно в жюри по направлению «Информационные технологии» принимают участие представители Института систем информатики СО РАН им. А. П. Ершова.

Конференция, участие в которой направлено на то, чтобы развивать творческий потенциал учащихся, приобщить их к исследовательской работе в различных областях научно-технической деятельности, состоит из нескольких этапов. На первом этапе учащиеся работают со своими наставниками (учителями и представителями вузов и научных учреждений в качестве руководителей). Затем, как правило, с выполненной работой школьник выступает перед школьной аудиторией. Несколько лучших работ выходят на уровень района, где осуществляется отбор на городской тур. Жюри городского тура сначала заочно знакомится с представленными работами. Безусловно, научная деятельность требует не только самостоятельности в

интеллектуально-творческой работе, формирования социальной активности в созданных учителями условиях профессионального самоопределения, но и воплощения своих знаний и способностей, полученных результатов деятельности в теоретическую и практическую часть отчета. Поэтому на заочном этапе члены жюри смотрят предоставленные работы потенциальных участников на предмет соответствия содержания работы заявленной теме, умение обосновать выбор темы согласно актуальности и новизне идеи.

При обосновании актуальности исследования необходимо решить, почему именно эту проблему нужно в настоящее время изучать, что является основным требованием к исследовательской работе, что необходимо для получения новых данных, проверки совсем новых методов и т. п. Для каждого школьника актуальность может быть обусловлена разными факторами. Кто-то восполняет недостающие научные знания, для другого важно развивать проблему в современных условиях. Некоторые просто обобщают накопленный как личный, так и всем человечеством опыт в данной области, иной осуществляет постановку новых проблем с целью привлечения общественного внимания. Традиционно кроме актуальности и новизны представленных работ жюри отслеживает владение техникой исследовательского поиска, степень проработки изложенного материала. Часто мы сталкиваемся с тем, что в качестве обоснования выбора темы школьники ссылаются на имеющиеся в Википедии материалы, формируют реферативную работу. Научно-практическая деятельность должна привести в соответствие именно полученные в процессе исследования результаты с заявленными целями и задачами. Кроме этого, обращается внимание на культуру оформления исследовательской работы, соответствия стилистическим особенностям, приведенные ссылки на литературу, с которой работал исследователь. Безусловно, существуют на просторах Интернета и достойные материалы, которые представляют интерес в качестве базового исследования. Но выводы, которые школьник должен привести в своей работе, должны соответствовать проведенным исследованиям и заявленным целям.

На очном этапе школьники делают доклад. Сегодняшний день диктует свои условия. Как правило, рассказ сопровождается презентацией. Не все ребята владеют навыками публичного выступления, теряются и начинают дублировать содержимое слайдов. Здесь роль учителя не должна быть сведена на нет. Очень важно научить ребят выступать перед аудиторией, подавляя свое волнение, говорить доступно и понятно, не включать непонятную терминологию в текст доклада. Мультимедийная презентация должна быть в помощь участникам, а не отвлекающим моментом. В нашей области «информационные технологии», как правило, выполненные научно-практические работы поддержаны проектами. На сегодняшний день большую популярность получили проекты с использованием роботизированных механизмов. При транспортировке технические модели могут невольно пострадать, потому неплохо при подготовке презентации включить в нее видеоролики заранее.

До недавнего времени научно-практические конференции школьников предназначались для учащихся преимущественно старшего школьного возраста. Сейчас НПК работает еще в двух номинациях: «Мое первое открытие» для начальной школы [6] и для среднего звена — конкурс исследовательских проектов, который проводится по направлениям научно-техническое, естественнонаучное, гуманитарное, инженерно-технологическое (изобретательский, конструкторский, проект по решению реальной производственной задачи и др.)

В нашей области деятельности бывают представлены замечательные проекты [5]. Творческая работа в области информационных технологий предусматривает создание интеллектуального или материального продукта. Опять же субъективная или объективная новизна выполненной работы поддерживается потенциалом практического применения. Работа может быть выполнена самостоятельно или при участии консультанта. Некоторые проекты уже проходят апробацию в жизни. Например, запомнилась работа шестиклассника, который сделал систе-

му автоматического полива грядок с клубникой. Симбиоз физики, программирования, изобретательских качеств личности помогли создать полезный продукт. Исследование программ на предмет выполнения практических задач также является одним из популярных направлений творческой мысли учащихся. Ученики из Академгородка написали проект, который моделирует кормушки для белок и птиц с последующим выпиливанием деталей на станке.

Для представления проекта автору дается 8–10 минут. За это время он должен обосновать выбор темы и проблему проекта, сформулировать поставленные задачи, описать процесс их решения, четко сформулировать полученные результаты и выводы. Надо отметить, что наиболее естественно ведет себя младшая аудитория. Они, как правило, непосредственны, очень хорошо и эмоционально говорят о своей деятельности, если чрезмерно не «зарепетировались» с наставниками. Жюри оценивает работы на предмет умения выделить и сформулировать проблему, определить цель и задачи проекта, оригинальность идеи, исполнения, результатов проекта, наличие исследовательского компонента в проекте, возможность практического применения проекта в целом или его отдельных результатов. Важна степень самостоятельности автора при выполнении проекта. Кроме того, жюри смотрит на качество оформления презентации, умение ответить на вопросы, общую культуру общения.

Хочется отметить качественное оформление работ, представляемых на конференцию в младшей возрастной группе и среднем звене. Начало введения посвящается обоснованию выбора темы. Описывается актуальность, проблема проекта, постановка цели, задачи. Обосновывается выбор инструментария, использование технологий, методик для реализации целей и задач. Приводится описание планирования, распределение ролей среди участников проекта и т. д. В основной части описывается теоретическое обоснование выбранной темы, описывается выполнение по этапам, включающим поиск информации, оформление работы, подготовка к презентации... В заключении делаются выводы. Описывается решение проблемы, рефлексия по поводу завершения проекта, трудности и успехи в период работы над проектом, достижение конечной цели, область применения, результативность, перспективы дальнейшей работы. В конце приводится список использованной литературы. Бывает, что в приложениях есть фото, видеоотчеты, схемы, графики, рисунки, анкеты соцопроса, расчеты и т. п.

НПК «Мое первое открытие» проходит по регламенту 7 минут на доклад и 3 минуты на обсуждение и вопросы. Выступления участников, поддержанные мультимедийной презентацией, отражают суть проделанной работы. Большинство детей используют презентацию как дополнение к устному докладу. Демонстрационная (практическая) часть выполненной работы самостоятельно показывается детьми после устного доклада. В представленных работах прослеживается роль учителя (руководителя), который позаботился о стиле изложения и подачи выполненной исследовательской работы.

Основная защита работы проходит во время выступления и демонстрации. Дополнительное время, выделенное на вопросы, направлено на развитие творческой мысли всех участников, чтобы они учились дискуссии в научном сообществе. Большинство представленных работ с хорошо структурированной теоретической и с выполненной практической частью. Дети свободно рассказывают об используемых в работе технологиях, самостоятельно демонстрируют свои устройства, фильмы. Большинство выступающих рассказывают о представленной работе с осознанием дела, компетентны в тематике, оценивают адекватно свой вклад как в исследовательскую, так и в практическую часть.

Информатика относится к числу фундаментальных наук, которая включает в сферу своих интересов как теоретическую информатику (например, дискретная математика, теория алгоритмов, теорию игр и т. д.), так и практическая (программирование, вычислительная техника и информационные технологии). Работа, относящаяся к мультимедийным технологиям (компьютерная анимация), робототехника, первые веб-странички с подобранным материалом

соответствует возрастным особенностям учащихся начальной школы и вызывает искренний интерес.

К достоинству организаторов конференции надо отнести тот факт, что каждый из участников секции получает диплом лауреата и памятную медаль. Дружественный подход к юным исследователям, присутствующий на научно-практической конференции, должная оценка, компетентное напутствие и предложенное направление, в котором можно продолжить развитие исследований — залог успешного начала осознанного занятия исследовательской деятельностью в будущем.

В НГУ проводится школьная секция Международной научно-практической конференции школьников. Научные сотрудники ИСИ СО РАН являются членами жюри секции «Информационные технологии». В этом году несколько работ старших школьников весьма достойно выглядели на МНСК.

Одна из работ учащегося лицея НГТУ была не только с реализованной практической частью собственного языка программирования, но и с подробным анализом парадигм программирования, начиная с процедурного, объектно-ориентированного до функционального. Школьник 11-го класса весьма уверенно и грамотно анализировал достоинства и недостатки различных языков программирования, на основе этого анализа создал свой.

Другое направление было представлено учащимися ВКИ НГУ, которые работают как программисты и инженеры в научных институтах СО РАН. Они создают не только программы, поддерживающие эксперименты, но и разрабатывают технологически интересные приборы, поддерживающие научные эксперименты.

Нейронные сети являются на сегодняшний день одним из популярнейших видов деятельности в различных областях. Было несколько работ в этом направлении. Хочется отметить работу, направленную на обучение алгоритмов в современной действительности. Поддержка людей с ограниченными возможностями в связи с плохим зрением была решена группой молодых исследователей, которые основывались на обучении готовых алгоритмов для мобильных устройств. Другая работа была посвящена более тщательному анализу существующих алгоритмов и реализации собственной модели искусственного нейрона для решения задачи предсказания бинарной классификации.

Традиционно институт систем информатики им. А. П. Ершова работает в направлении раннего вовлечения школьников в профессию программиста. Летняя школа юных программистов (ЛШЮП) [7, 8, 9], созданная в 1976 г. Андреем Петровичем Ершовым, бережно относится к традициям и старается сохранить эффективные механизмы и методики работы с будущими кадрами российской науки.

Традиционное разнообразие тематик мастерских на ЛШЮП предоставляет возможность в каждой из мастерских выполнить практические реализации проектов по заявленным темам. Только некоторые темы мастерских, получивших интересный результат: «Обработка больших данных», «Умный дом», «Квантовые вычисления», «Генетические алгоритмы и нейросети», «Создание языков программирования», «Работа с кластером, суперкомпьютеры» и др. говорят о том, что ЛШЮП всегда остается тем звеном, которое связывает школу с современными направлениями в науке.

Основными задачами ЛШЮП является отбор талантливых старшеклассников, заинтересованных в овладении профессиональным программированием, обучение учеников среднего звена навыкам коллективной работы с применением современных информационных технологий и содействие развитию способностей к практическому программированию учащихся младших классов, а также поддержка педагогов, успешно преподающих информатику и программирование в общеобразовательной системе.

Одна из целей Летней школы юных программистов — дать почувствовать учащимся технологию разработки. Опыт по обучению программистов технологии небогат в столь юном возрасте, взгляды на подходы значительно разнятся. Потому в мастерских значительное время уделяется ретроспективе, осмыслению проделанной работы. Содержание этапов проектирования, макетирование, отладка, тестирование и составление документации показывает тесное взаимодействие в коллективе всех участников, выполняющих проект. Благодаря совместному анализу участников мастерской и преподавателей, неудачи превращаются в достижения.

Заключение

Безусловно, научно-практические конференции школьников, в которые вовлечены ребята как можно более младшего возраста, формируют особое отношение к любой профессии. Любознательность, данная человеку от природы, должна быть поддержана мотивацией и руководством к результативному проведению исследований, воспитывать творческое отношение к любой профессии, дать в руки инструменты для накопления научных знаний, сформировать активную жизненную позицию.

Список литературы

1. Марчук А. Г., Тихонова Т. И. Традиции в системе подготовки творческой молодежи // Компьютерные инструменты в школе. № 2. СПб, 2008. С. 3–11.
2. Выготский Л. С. Умственное развитие детей в процессе обучения. М. : Госучеб педагог, 1935.
3. Коменский Я. А. Избранные педагогические произведения : в 2 т. Т. 2. М., 1982.
4. Кулагина И. Ю. Возрастная психология: развитие ребенка от рождения до 17 лет. 5-е изд. М. : Изд-во УРАО, 1999.
5. Тихонова Т. И. Отчет о работе новосибирской Летней школе юных программистов 2014 // Системная информатика. 2015. № 5. С. 75–104.
6. Сапрыкина Г. А., Тихонова Т. И. Конференции школьников как способ выявления специальных одаренностей // Сибирский учитель. № 4 (113). Новосибирск, 2017. С. 55–58.
7. Тихонова Т. И. Становление личностных качеств программиста : сб. докладов и тезисов 9-й Ершовской конференции по информатике. Рабочий семинар «Информатика образования» (PSI14). 2014. С. 70–72.
8. Марчук А. Г., Тихонова Т. И. Летняя школа юных программистов — этап становления школьной информатики : сб. науч. трудов Междунар. науч.-практич. конф. «От информатики в школе к техносфере образования». М., 2016. С. 243–249.

UDK 37.018.536''322: 004.9, 004.438, 37.047

**SCIENTIFIC-PRACTICAL CONFERENCE
OF SCHOOLCHILDREN TO MOTIVATE RESEARCH ACTIVITIES**

T.I. Tikhonova

Researcher, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentiev pr., 6,
Senior Lecturer, Novosibirsk State University,
6300906, Novosibirsk, Russia, Pirogova St., 2, e-mail: tanja@iis.nsk.su

Abstract. The development of a research approach to any activity is important to start at an early age. Inquisitiveness is a natural quality of a person. The article discusses the work with students aimed at developing research capacity of children. Engaging in creative work by performing research with the practical implementation of the project, to make a report at the scientific-practical conference forms a proactive stance, socializes personality and lays the qualities necessary for professional work.

Keywords: information technology, project activities, education and creativity

References

1. *Marchuk A. G., Tikhonova T. I.* Tradicii v sisteme podgotovki tvorcheskoj molodezhi // Komp'yuternye instrumenty v shkole. № 2. SPb, 2008. P. 3–11. (in Russian)
2. *Vygotskij L. S.* Umstvennoe razvitie detej v processe obuchenija. M. : Gosucheb pedagog, 1935. (in Russian)
3. *Komenskij Ja. A.* Izbrannye pedagogicheskie proizvedenija : v 2 t. T. 2. M. : Pedagogika, 1982. (in Russian)
4. *Kulagina I. Ju.* Vozrastnaja psihologija: razvitie rebenka ot rozhdenija do 17 let. 5-e izd. M. : Izd-vo URAO, 1999. (in Russian)
5. *Tikhonova T. I.* Otchet o rabote novosibirskoj Letnej shkole junyh programmistov 2014 // Sistemnaja informatika. 2015. № 5. P. 75–104. (in Russian)
6. *Saprykina G. A., Tikhonova T. I.* Konferencii shkol'nikov kak sposob vyjavlenija special'nyh odarenostej // Sibirskij uchitel'. № 4 (113). Novosibirsk, 2017. P. 55–58 (in Russian)
7. *Tikhonova T. I.* Stanovlenie lichnostnyh kachestv programmista : sb. dokladov i tezisov obrazovanija» (PSI14). 2014. P. 70–72. (in Russian)
8. *Marchuk A. G., Tikhonova T. I.* Letnjaja shkola junyh programmistov — jetap stanovlenija shkol'noj informatiki : sb. nauch. trudov Mezhdunar. nauch.-praktich. konf. «Ot informatiki v shkole k tehnosfere obrazovanija». M., 2016. P. 243–249. (in Russian)

ОСОБЕННОСТИ ПРЕПОДАВАНИЯ ОСНОВ ВЫЧИСЛЕНИЙ С ПЛАВАЮЩЕЙ ТОЧКОЙ

Б. Л. Файфель

доц., Саратовский государственный технический университет им. Ю. А. Гагарина,
410054, Саратов, ул. Политехническая, 77, e-mail: catstail@ya.ru

Аннотация. В работе представлен опыт преподавания основ плавающей арифметики на примере ряда типовых задач и ситуаций. Рассмотрены примеры потери значимости при суммировании рядов и сформулирован ряд практических рекомендаций. В частности, показано, что даже абсолютно сходящийся ряд не всегда может быть успешно просуммирован с использованием плавающей арифметики, но вполне поддается суммированию при использовании арифметики рациональной.

Ключевые слова: плавающая точка

Введение

Современная наука и инженерия использует вычисления с плавающей точкой весьма активно. Однако действующие в настоящее время учебные курсы (как в высших, так и в средне-специальных учебных заведениях) не уделяют работе плавающей точкой должного внимания. Результатом сложившейся ситуации является то обстоятельство, что выпускники вузов часто не знают элементарных вещей, связанных с плавающей арифметикой, и допускают непростительные ошибки. Предлагаемая работа отражает опыт автора.

1. Постановка задачи

Как хорошо известно, арифметика с плавающей точкой имеет ряд существенных отличий от арифметики целых чисел. Главное отличие состоит в том, что числа с плавающей точкой обрабатываются с некоторой погрешностью. Это кажущееся незначительным отличие приводит к весьма серьезным последствиям (вплоть до нарушения закона коммутативности сложения). Учащиеся, впервые столкнувшиеся с подобными эффектами, оказываются деморализованными и обвиняют компьютер («машина работает неправильно»).

Для предотвращения подобных ситуаций целесообразно как можно раньше объяснить обучаемым особенности арифметики чисел с плавающей точкой и дать необходимые рекомендации. Для объяснения лучше всего использовать модельные ситуации, как это сделано в [1].

Начать, вероятно, следует с правильной терминологии — объяснить некорректность понятия «число десятичных знаков после запятой» и ввести в обиход корректный термин «число значащих цифр».

© Б. Л. Файфель, 2019

Автор выражает искреннюю благодарность Н. В. Шилову и Т. И. Тихоновой за поддержку и интерес к работе.

Далее следует показать вычислительным экспериментом, что целые числа компьютер обрабатывает в полном соответствии с законами алгебры. Для этого можно программно просуммировать диапазон целых $[1, 10000]$ двумя способами: от больших к меньшим и от меньших к большим. Результат, естественно, получится один и тот же.

Если же вычислить сумму обратных величин $1+1/2+1/3+\dots+1/10000$ (используя данные типа float) также двумя способами (от меньших к большим и от больших к меньшим), то результаты уже не будут идентичными друг другу:

$$\begin{aligned} s1 &= 9,787613 \\ s2 &= 9,787604 \end{aligned}$$

Как видим, вычислительный эксперимент показывает расхождение результатов (которые должны бы совпадать в соответствии с законами алгебры). В этом месте необходимо подчеркнуть, что мы в данном случае не можем решить, какой из этих результатов верен (возможно, неверны оба, но один может быть ближе к правильному значению). Уместно также сказать, что проигнорировать расхождение в двух последних цифрах нельзя: в других случаях расхождение может оказаться очень большим. Так, если вычислить эти суммы от 1 до $1/10000000$, то результат окажется следующим:

$$\begin{aligned} s1 &= 15,403683 \\ s2 &= 16,686031 \end{aligned}$$

Здесь верной оказалась только одна значащая цифра. Коварство этой ситуации в том, что для коротких отрезков суммируемого ряда результаты совпадают. Для того чтобы хотя бы понять, какой из результатов ближе к истине, следует вычислить на компьютере сумму, значение которой известно точно, взяв какую-либо специфическую последовательность с точно вычисляемой суммой (как это сделал автор работы [1]). Сумма следующего вида:

$$1/(1*2) + 1/(2*3) + 1/(3*4) + \dots + 1/(999999*1000000)$$

с помощью несложного преобразования может быть вычислена точно. Ее значение равно $1-1/1000000=0,999999$. Если же вычислить эту сумму без преобразования двумя способами (от больших к меньшим членам и от меньших к большим), то получится следующий результат:

$$\begin{aligned} s1 &= 0,999853 \\ s2 &= 0,999999 \end{aligned}$$

Как видим, сложение, проведенное от меньших чисел к большим, дало точный результат, а сложение в другом порядке обеспечило лишь три верные значащие цифры.

Теперь следует объяснить причину такого поведения. Для этого можно углубиться в детали представления чисел с плавающей точкой (стандарт IEEE 574), однако для начинающих этот путь представляется достаточно сложным, так как он изобилует многочисленными второстепенными деталями.

2. Упрощенный взгляд на представление чисел с плавающей точкой

Будем считать, что компьютер работает не в двоичной, а в десятичной системе счисления (на основные выводы это не повлияет, но повысит наглядность). Обучаемым следует разъяснить форму представления числа с плавающей точкой в компьютере. Эта форма имеет вид:

$$\pm 0.\text{mmmmmm} \pm \text{pp},$$

где 0.mmmmmm — мантисса с соответствующим знаком, а pp — десятичный порядок. Число, представленное в такой форме, на самом деле равно:

$$\pm 0.\text{mmmmmm} * 10^{\pm \text{pp}}.$$

Здесь уместно сразу же объяснить, что, поскольку количество разрядов мантиссы конечно, целесообразно выбрать такое представление числа, при котором первый разряд мантиссы ненулевой (такая форма называется нормализованной). Нормализованная форма позволяет максимально использовать разрядную сетку.

После объяснения представления чисел с плавающей точкой можно объяснить причину погрешности, возникающей при сложении чисел. Рассмотрим процесс сложения двух чисел:

$$X = 0,12345E-3$$

$$Y = 0,54321E+2$$

Оба числа представлены в нормализованной форме, однако мы не можем просто сложить их мантиссы — сначала нужно выполнить денормализацию («записать разряд под разрядом») и только потом складывать. Будем считать, что наш модельный компьютер содержит два регистра с плавающей точкой F1 и F2. На рис. 1 показано, как располагаются числа X и Y в этих регистрах:



Рис. 1. Представление чисел в регистрах модельной машины

Процесс денормализации показан ниже на рис. 2.

Из рисунка отчетливо видно, что от слагаемого, подвергающегося денормализации, остаются всего две значащие цифры. Таким образом, обучаемые подводятся к главному выводу: при сложении чисел с плавающей точкой при разных порядках слагаемых обязательно происходит некоторая потеря значимости результата. Именно этим и объясняются результаты вычислительных экспериментов, приведенные выше.

Этот подход позволяет достаточно легко объяснить обучаемым резкое возрастание погрешности, возникающее при вычитании близких чисел. Для этого можно рассмотреть вычитание двух чисел $0,1234567 - 0,1234568$. Представление этих чисел в регистрах модельной машины, а также результат вычитания, представлен на рис. 3. Видно, что в результате вычитания у результата остается единственная верная значащая цифра.

ЗН	п	1	2	3	4	5	6	7		ЗН	1	2	3
+	0	1	2	3	4	5	0	0		-	0	0	3

ЗН	п	1	2	3	4	5	6	7		ЗН	1	2	3
+	0	0	1	2	3	4	5	0		-	0	0	2

ЗН	п	1	2	3	4	5	6	7		ЗН	1	2	3
+	0	0	0	1	2	3	4	5		-	0	0	1

ЗН	п	1	2	3	4	5	6	7		ЗН	1	2	3
+	0	0	0	0	1	2	3	4	5	+	0	0	0

ЗН	п	1	2	3	4	5	6	7		ЗН	1	2	3
+	0	0	0	0	0	1	2	3	4	+	0	0	1

ЗН	п	1	2	3	4	5	6	7		ЗН	1	2	3
+	0	0	0	0	0	0	1	2	3	+	0	0	2

Рис. 2. Процесс денормализации поэтапно

ЗН	п	1	2	3	4	5	6	7		ЗН	1	2	3
+	0	1	2	3	4	5	6	7		+	0	0	0

ЗН	п	1	2	3	4	5	6	7		ЗН	1	2	3
-	0	1	2	3	4	5	6	8		+	0	0	0

ЗН	п	1	2	3	4	5	6	7		ЗН	1	2	3
-	0	0	0	0	0	0	0	1		+	0	0	0

Рис. 3. Вычитание близких чисел

3. Проблемы, возникающие при суммировании рядов

Как хорошо известно, косинус вещественного x представляется следующим рядом Тэйлора:

$$\cos(x) = 1 - (x^2/2!) + (x^4/4!) - (x^6/6!) + \dots$$

Этот ряд сходится абсолютно при любом x . Студентам часто дают задачу вычислить косинус (или синус) суммированием ряда, выполняемом до тех пор, пока модуль очередного вычисляемого члена ряда не станет меньше заданной заранее точности. Представляет методический интерес сравнить результаты, полученные суммированием ряда (Cos) с результатами работы библиотечной функции (cos) [2]. Сравнение представлено на рис. 4.

x	cos(x)	Cos(x)	Расхождение
0.000e+000	1.000000e+000	1.000000e+000	0.000000e+000
5.000e-002	9.987503e-001	9.987502e-001	1.000000e-008
1.000e-001	9.950042e-001	9.950042e-001	0.000000e+000
1.500e-001	9.887711e-001	9.887711e-001	0.000000e+000
2.000e-001	9.800666e-001	9.800666e-001	0.000000e+000
1.850e+001	9.395249e-001	1.144768e+000	2.052426e-001
1.900e+001	9.887046e-001	9.913036e-001	2.599001e-003
1.950e+001	7.958150e-001	6.106047e-001	1.852103e-001
2.900e+001	-7.480575e-001	1.315880e+004	1.315954e+004
2.950e+001	-3.383192e-001	3.822034e+003	3.822372e+003
3.000e+001	1.542515e-001	-2.368533e+003	2.368688e+003

Рис. 4. Вычисление косинуса прямым разложением в ряд

Видно, что при малых значениях аргумента результат совпадает с результатом вызова библиотечной функции. Однако при возрастании аргумента погрешность начинает расти и результат становится просто абсурдным. Причина этой ситуации состоит в том, что при значениях аргумента порядка 30 и выше члены ряда быстро возрастают до величин порядка 10^{11} и только потом начинают убывать (когда факториал «наберет силу»). Получается, что результирующее значение косинуса есть разность двух сумм P (сумма положительных членов) и N (сумма отрицательных). При этом обе эти суммы достигают величин, порядка 10^{12} , а их разность должна быть меньше единицы. Чтобы результат вычитания сохранил хотя бы одну верную значащую цифру, мантисса числа с плавающей точкой должна быть длиной не менее 11 десятичных цифр. Тип данных *float* располагает мантиссой (в десятичном эквиваленте) в 7-8 десятичных цифр. Именно этим и объясняется невозможность прямого суммирования ряда для косинуса при больших значениях аргумента и при короткой мантиссе.

В традиционных курсах математики вопросы вычислений обычно не рассматриваются, и получается, что «программирование опровергает математику». Эта проблематика редко затрагивается при обучении, что впоследствии порождает непонимание и ошибки.

Тем не менее просуммировать этот ряд напрямую можно, если отказаться от арифметики с плавающей точкой и перейти к арифметике рациональных чисел. Ниже приводится код на языке Лисп (в котором поддерживается арифметика рациональных чисел):

```
(defun my-cos (x &optional (eps 1e-8))
  (let ((a 1) (s 1) (k 0))
    (loop
      (when (<= (abs a) eps) (return s))
      (setq a (- (/ (* a x x) (+ k 1) (+ k 2)))
            s (+ s a)
            k (+ k 2))))))
```

Вызов этой функции с аргументом $x = 50$ возвращает следующий результат (для наглядности числитель и знаменатель этой дроби показаны с разбивкой):

```
243706131654541132675603386082219549825542813852030945546703580040720392481
21649326796191979218353411428243256901695353743984506265611950655237792210831033
740166338199817232878060581913569126766599
/ 25255410493873184332225648114958816946608988211936130235611855567635907889
66318443878980153006888502210533710469572846996825946020610949081573617550435820
2660509266505949702813572299506856327202849
```

Если привести эту дробь к форме с плавающей точкой двойной точности, то получится результат 0,964966028620532. Библиотечная функция возвращает результат 0,964966028492113. Легко убедиться, что результаты совпадают в пределах восьми значащих цифр (что полностью соответствует значению параметра `eps`, заданного при вызове Лисп-функции).

Если же требуется повысить точность вычислений с плавающей точкой, то можно либо воспользоваться готовыми решениями (класс `BigDecimal` в языке Java), либо реализовать операции с плавающей точкой и мантисой любого размера самостоятельно. Вторым путем, по мнению автора настоящей работы, представляется методически более полезным.

В Лиспе реализовать такую арифметику несложно. Число с плавающей точкой можно моделировать списком из двух целых: первый элемент — мантисса (без целой части), второй элемент — порядок. Оба элемента могут иметь знаки. Ниже приведена модельная функция вычисления косинуса прямым разложением в ряд:

```
(defun my-cosf (x &optional (n 100))
  (let ((a '(1 1)) ;; a=1
        (s '(1 1)) ;; s=1
        (k '(0 0))) ;; k=0
    (dotimes (i n s) ;; повторяем n раз
      (print 'a)=(sfaPrint a)(terpri) ;; печать a
      (print 's)=(sfaPrint s)(terpri) ;; печать s
      (print 'k)=(sfaPrint k)(terpri)(terpri) ;; печать k
      (setq a (sfaMul (sfaMul a x) x)) ;; a=a*x2
      (setq a (sfaDiv a (sfaAdd k '(1 1)))) ;; a=a/(k+1)
      (setq a (sfaDiv a (sfaAdd k '(2 1)))) ;; a=a/(k+2)
      (setq a (sfaNeg a)) ;; a=-a
      (setq s (sfaAdd a s)) ;; s=s+a
      (setq k (sfaAdd k '(2 1)))) ;; k=k+2
```

Здесь `sfaPrint`, `sfaMul`, `sfaDiv`, `sfaNeg` и `sfaAdd` — функции печати, умножения, деления, изменения знака и сложения чисел с плавающей точкой при описываемом подходе. Результат вычисления косинуса этой программой при сороказначной мантиссе и $x = 50$ дает такой результат:

```
a=-0,4186150934836135941204740284155915144198e-22
s= 0,9649660284921132741064407483993783117076e0
k= 0,1780000000000000000000000000000000000000000000000000000e3    k=178
```

Библиотечная функция дает результат 0,964966028492113. Легко убедиться, что все цифры результата совпадают с цифрами, полученными выполнением Лисп-кода.

Заключение

Рассмотрение вопросов, затронутых в настоящей работе, должно сформировать у обучаемых ряд навыков и понятий, которыми следует пользоваться при необходимости проводить вычисления с плавающей точкой:

- главной характеристикой арифметики с плавающей точкой является длина мантиссы, поэтому при прочих равных условиях стандартный тип *double* предпочтительнее типа *float*;
- при вычислениях сумм следует начинать суммирование от меньших членов к большим (это снижает погрешность вычисления, связанную с конечностью мантиссы);
- избегать вычитания близких чисел.

Список литературы

1. Филличев С. В. Занимательный Basic. М. : Эконом, 1997. 191 с.
2. Мак-Кракен Д., Дорн У. Численные методы и программирование на Фортране. М. : Мир, 1977, 580 с.

UDK 519.622.2

SPECIFICITIES OF TEACHING FLOATING POINT COMPUTATION

B. L. Fayfel

Associate professor, Saratov State Technical University of Yu. A. Gagarin
410054, Saratov, Politehnicheskaya st., 77, e-mail: catstail@ya.ru

Abstract: The paper contains experience of teaching fundamentals of floating arithmetics on the example of number standard tasks. We have considered examples of loss of significance when summing up the series. A series of practical recommendations for solving such problems are formulated. The author showed that even an absolutely convergent series cannot always be successfully summed on the using floating arithmetic, but it is quite amenable to summation on the using rational arithmetic.

Keywords: floating point

References

1. Filichev S. V. Zanimatelnyi Basic. M. : Ekonom, 1997. 191 p.
2. Mak-Kraken D., Dorn U. Numerical methods and Fortran programming. [In Rus.] M. : Mir, 1977. 584 p.

УДК 004

ОТЕЧЕСТВЕННАЯ ТЕОРИЯ ПРОГРАММИРОВАНИЯ. ПРОШЛОЕ И НАСТОЯЩЕЕ

Е. М. Лаврищева

д-р физ.-мат. наук, гл. науч. сотрудник ИСП РАН,
109004, Москва, ул. А. Солженицына, д. 25, e-mail: lavr@ispras.ru, lavrischeva@gmail.com

Аннотация. Сформулированы научные основы программ и систем с учетом теоретических положений академика А.П. Ершова. Приведены первые парадигмы программирования и показан путь их развития в период информатизации. Дано определение программных понятий – модуль, сборка и интерфейс на период 1970–1980-х гг. и подходов к доказательству правильности систем из модулей и компонентов повторного использования (КПИ), reuses и т. п.

Приводится определение парадигм программирования и дана формальная характеристика объектной, компонентной, сервисной и других парадигм. Основу технологии программирования составляет метод сборки в конфигурационную структуру для решения некоторой прикладной задачи с обеспечением качества. Обучению парадигмам программирования сложных систем посвящен сайт <http://7draguns.ru/>.

Ключевые слова: наука, теория, абстракция, метод, математические операции, граф, модель, верификация, конфигурация, тестирование, качество

UDK 004

NATIONAL PROGRAMMING THEORY: THE PAST AND THE PRESENT

E. M. Lavrischeva

Dr. of ph. and math. sc., Chief researcher, Ivannikov Institute for System Programming RAS,
109004, Moscow, Alexander Solzhenitsyn st., 25, e-mail: lavr@ispras.ru, lavrischeva@gmail.com

Abstract. The author has defined the scientific foundations of programs and systems taking into account the theories developed by Academician A. P. Ershov. The first programming paradigms are presented and their development during the computerization period is shown. The paper defines some software concepts — module, assembly and interface — as they were understood in the 1970s–1980s and approaches to proving the correctness of systems made of reusable modules and components. Definitions of programming paradigms and the formal characteristic of the object-oriented, component-oriented and service-oriented paradigms are given. The basis of the programming technology is the method of assembling a configuration structure designed to solve application problems at a good quality level. To learn the programming paradigms of complicated systems, you can go to <http://7draguns.ru/>.

Key words: science, theory, abstraction, method, mathematical operations, graph, model, configuration, testing, quality

Научное издание

ПСИ'19

**12-Я МЕЖДУНАРОДНАЯ ЕРШОВСКАЯ КОНФЕРЕНЦИЯ
ПО ИНФОРМАТИКЕ**

ИНФОРМАТИКА ОБРАЗОВАНИЯ

Труды семинара

2–3 июля 2019 г.

Материалы конференции публикуются в авторской редакции.

Корректор *Я. О. Козлова*
Верстка *А. С. Терешкиной*
Дизайн обложки *Т. Бульонковой*
Подготовка обложки к печати *Е. В. Неклюдовой*
Ответственный за выпуск *Т. И. Тихонова*

Подписано в печать 27.06.2019 г.
Формат 60 × 84 1/8. Уч.-изд. л. 12,5. Усл. печ. л. 11,6.
Тираж 80 экз. Заказ № 157.
Издательско-полиграфический центр НГУ.
630090, Новосибирск, ул. Пирогова, 2

Организаторы:



Институт систем информатики им. А. П. Ершова

N* Новосибирский
государственный
университет
***НАСТОЯЩАЯ НАУКА**

Спонсоры:

N* Novosibirsk
State
University
***THE REAL SCIENCE**



Microsoft Research

ARQA
TECHNOLOGIES

SIBERS[®]



STI · INTERNATIONAL



STI · INNSBRUCK

ISBN 978-5-4437-0917-8



9 785443 709178