Bazayeva Zh.B., Nurahmetova A.E.
Kazakh State Women's Teacher Training University, Kazakh-American University, Almaty, Kazakhstan

USING OF WEB-VISUALIZERS IN THE SCHOOL PROGRAMMING COURSE

The article describes a method to improve the quality of mastering the material, which is based on the use in teaching programming tutorials, simulators, namely web-visualizers (program-simulators, which are web-applications and demonstrates the process of the algorithms). Offers training in the basics of programming, in particular cyclic structures, using problem-based learning methodology.

Keywords: *programming, simulator, web-visualizer, problem-based learning, methods of teaching.*

В статье рассматриваются обучающие программы-тренажеры при обучении программированию учеников в школе, а именно о применении web-визуализаторов. Также предлагается обучение циклических конструкций применяя методику проблемного обучения.

Ключевые слова: *программирование, тренажер, web-визуализатор, проблемное обучение, методика обучения.*

An important feature of learning programming in school is that the possession of the relevant competency is not limited to the ability of simple reproduction of existing knowledge and is not limited to the ability to apply template solutions. In fact, any real problem solved by the programmer requires unconventional thinking and nonstandard actions.

It is obvious that the teaching of programming in school begins with a discussion of the elementary structures. After learning the basics of programming on any algorithmic language should go to typical problems that are the basis of algorithmic culture and serve as a reference point for further study of programming. Selection of tasks is based on the standard program of a school course. Students must master the original programming skills in high level language, which includes the ability to develop linear structure algorithms, use branching statements, loops, organization subprograms, including recursive programs. The student should be able to use simple and compound data types: integer, real, symbols, arrays, records. Most important in teaching programming are the following algorithms: linear performance of the algorithm; conditional statement; while loop; repeat until loop; for loop; summation of array elements; find minimum and maximum array elements; simple array sorting; substring search; representation of the set in the computer; organization of recursion; input, output to a file.

For profile informatics course oriented to in-depth study of programming, you can expand the substantive content of the following themes: trees, tree traversal, search algorithms; matrix, work with numbers and matrices, strings, lists; invariants, inductive evidence; generation of pseudo-random sequences.

In this case, a list of considered algorithms is necessary to expand by the following algorithms: matrix multiplication; advanced array sorting; external sorting, sorting the file; work of stack and queue.

This list is not exhaustive.

The choice of these tasks justified goals and objectives of teaching programming at school. However, be aware that it is unacceptable to link the student with some specific

templates, it is important to develop the ability to act creatively.

When teaching programming in demand are not only special knowledge and algorithmic thinking, developing in the process of algorithmization, but also formed logical thinking.

Formation of logical thinking, usually starts in elementary school in studying the foundations of mathematics and logic. Logic ability of students can be developed only when they are actively involved in the acquisition of new knowledge. One of the most effective methods of self-development of logical thinking is a problem-based learning, as it is closest to the creative work of the scientist, which is characterized by using the hypotheses, evidence, experiment.

Techniques for creating problem situations are chosen depending on the specific content of the training material. In some cases the problem situation is created based on existing knowledge of students. Based on them, students make a conclusion, which is in conflict with the facts. It means that the knowledge is inadequate and need more information to resolve the controversy. This variant of a problem situation always cause keen interest in students, hence the cognitive efficiency is high.

A number of teachers and psychologists (VOkon' V., I. YA. Lerner, M. I. Makhmutov, T. V. Kudryavtsev) offer problem-based learning methodology for the development of logical thinking. Problematic methods are based on the creation of problematic situations, active learning of students, consisting in searching and solving complex issues that require actualization of knowledge, analysis, ability to see beyond the individual facts only phenomena and laws.

The teacher creates a problematic situation, directs students to solve it, organizes the search for solutions. Thus, students in the search process acquire new knowledge, takes possession of the new modes of action. The problem situation is created specifically by a teacher, using a series of special instructional techniques.

Depending on the nature of the interaction between teachers and students, there are four levels of problem-based learning:

1. The level of non-independent activity - the perception of students the teacher's explanations, the absorption of the sample of mental action in a problematic situation, the implementation of the independent work, exercises, oral reproduction.

In the study of programming non-independent activity is shown in the study of ready text programs. For example, the study of cyclic structures can start by looking at an example of the ready algorithm solving the problem of calculating the sum of elements in the array using for loop. You can prepare handouts containing program implementing the algorithm, followed by comments:

```
program sum1;
var a:array[1..10] of integer;
    s:longint;
    i:integer;
begin
   writeln('enter 10 elements of the array');
   s:=0;
   for i:=1 to 10 do begin
          readln(a[i]);
          s:=s+a[i];
          end;
   writeln('Sum of array elements = ', s );
end.
```

2. The level of self-activity is characterized by the use of previous knowledge in a new

situation and participation of students in finding ways to solve the problems of teacher.

This level is characterized by the study of new material on the basis of teacher prompts. As these tips you can use templates programs. For example, when studying loop can repeat student program template to compute the sum of the integers from 1 to N using a loop repeat:

```
program sum2; var
        i, s, n: integer;
begin

        writeln('Enter a number from 1 to n '); readln(n);

        i:=1; s:=0; repeat
                .....
        until
                .......
        writeln('s=',s);

end.
```

They should be able to:

```
repeat
        s:=s+i;
inc(i) until i<=n;
```

3. The level of self-activity - perform reproductively-type search, when a student decides on the text of the textbook, uses prior knowledge to new situations, designs, solves the problems of the average level complexity proves the hypothesis with little help of a teacher, and etc. For this we need to raise the level of pupil problem to be solved without the help of a teacher. For example, the solution of the problem of calculating the sum of integers from 1 to N by a loop for:

```
program sum3; var
        i, s, n: integer;
begin

        writeln('Enter a number from 1 to n'); readln(n);

        for i:=1 to n do s:=s+i; writeln('s=',s);

end.
```

As a self task - the task to calculate the factorial (product of natural numbers from 1 to N).

4. The level of creative activity - performance of independent work requiring Creator of-sky imagination, logical analysis and speculation, opening new ways of solving educational problems of self-evidence; independent conclusions and generalizations of the invention, the writing of art works.

At this level, you need to put the task on pupil of which there will not be easy to work independently, but also to approach this creatively. Stimulate creative activity can be using the element method of projects. The student should offer yourself to formalize and solve practical-oriented problems using cyclic structures. For example, you can instruct the student to write a program that would allow to calculate the average number of copybooks in the portfolios of students:

```
program notebook; var i, k, s:integer;
        p:real;
```

```
begin
        writeln('Enter the number of pupil); readln(n);
        s:=0;
for i:=1 to n do begin
writeln('Enter the number of copybooks y , i, 'pupil ');
        read(k);
        s:=s+k;
end;
p:=s/k;
writeln('the average number of copybooks', p); end.
```

Disciple of solving the problem, independently form the mathematical model. The first step will be made count of the total number of notebooks by summing the number of notebooks, tells each student. Upon completion of the count value summation is divided by the number of students.

Similar techniques can be used throughout the entire course of study.

Significant difficulties implementing this approach is the need to work with individual students. Indeed, the effectiveness of the methodology for each student will be achieved if he will fill the gaps and carry out tasks itself. Unfortunately, the teacher is difficult to give the necessary attention to each student. One solution to this problem is to use visualizers (software simulators), showing the internal logic of the algorithm and controlling the activities of the students. The important point is the possibility of placing simulators on the Internet. This technology makes it possible to provide universal access funds created (technically, this leads to the need for the implementation of simulators as a web-based application). Program-simulators, which are web-applications and demonstrate the process of working algo rhythms can be called web-visualizers [4]. These products can be widely applied in the traditional class-lesson system and to work independently.

References:

1. Informacionnaja sistema «Edinoe okno dostupa k obrazovatel'nym resursam»: baza dannyh soderzhit primernye programmy osnovnogo obshhego obrazovanija po informatike i IT. URL: http://window.edu.ru/window_catalog/pdf2txt?p_id=14196.

2. Informacionnaja sistema «Edinoe okno dostupa k obrazovatel'nym resursam»: baza dannyh soderzhit primernuju programmu profil'nogo urovnja. URL: http://window.edu.ru/window_catalog/pdf2txt?p_id=14196.

3. Verbickij A. Psihologo-pedagogicheskie osnovy obrazovanija vzroslyh: kontekstnyj podhod. URL: http://www.znanie.org/jornal/n2_01/ psih_podhod.html.

4. Jakimenko O.V., Stas' A.N. Primenenie obuchajushhih programm-trenazherov v obuchenii programmirovaniju // Vestnik TGPU. 2009. Vy-pusk 1 (79). Ser. «Pedagogika: teorija i praktika». S. 54–56.