GALAXY INTERNATIONAL INTERDISCIPLINARY RESEARCH JOURNAL (GIIRJ)
ISSN (E): 2347-6915
Vol. 11, Issue 1, Jan. (2023)

OPPORTUNITIES OF PROBLEMED EDUCATIONAL TECHNOLOGY IN INCREASE EFFICIENCY OF PROGRAMMING LANGUAGE TEACHING

Mirsanov Uralboy Mukhammadievich,
Associate Professor of the Navoi State Pedagogical Institute

ABSTRACT

This article presents the possibilities of problem-based learning technology in increasing the effectiveness of teaching programming languages Delphi, Borland C++ Builder, and suggestions and recommendations for its application.

Keywords: Delphi, Borland C++ Builder, programming, problem-based learning, creative thinking, logical, algorithmic.

Problem-based education is an improved teaching technology, the task of which is to encourage the process of active learning and to form a scientific-research method in thinking. Therefore, it is considered appropriate to use problem-based educational technology to improve the effectiveness of teaching subjects, including programming languages, and to develop students' critical, logical, and algorithmic thinking [1].

The main idea of problem-based education is independent acquisition of knowledge by students in solving problem situations. It serves as an important technology in improving the effectiveness of teaching informatics and information technologies and in developing students' creative thinking about information technologies. Because the development of computer technologies, software, and information technologies creates the need to deepen the content of the computer science course in relation to practical issues and creates difficulties in learning them. The complexity of computer systems, networks, technical objects, their management and technical maintenance makes the problem of training specialists in effective and optimal actions in solving practical problems more urgent [2].

In our opinion, problem-based learning helps informatics and information technology students to increase the level of independence in learning and to develop creative thinking. In this regard, the use of problem-based educational technology is of great importance in increasing students' motivation, creativity, and competences related to informatics and information technologies, including programming languages. Because by giving problematic assignments to students in teaching programming, mental difficulties are created in them. As a result, students have the opportunity to improve their creativity by independently solving programming problems.

Also, teaching students to think creatively and logically about programming leads to acquisition of mental activity methods, formation and development of scientific, critical-analytical, logical thinking skills in them. For example, is it possible to find prime numbers in two intervals in a Delphi environment? in order to find an answer to the problematic question, students conduct the following research: they look for the definition of prime number and its mathematical formula in the literature; if there is no mathematical formula for finding the prime number, then he tries to determine some rule by numerical calculation (for example, subtracting the given numbers from the root ($\sqrt{n}$) and getting the whole part); develops a prime number calculation algorithm; selects prime number-determining operators (operators such as for, if);

In the Delphi environment, selects buttons with the function of calculating and generating prime numbers; determines the ways to interconnect the given button and program codes; The program code for the given problem is written:

```
const  n=100000;
var
 k,m,l, i:integer;   a:array[1..n] of integer;
begin m:=strtoint(edit1.Text); l:=strtoint(edit2.Text);
for i:= 2 to n do  a[i]:=i;    for i:=1 to n do
   for k:=2 to trunc(sqrt(i)) do
if (a[i]/k-trunc(a[i]/k))=0 then a[i]:=0; for i:=m to l do if a[i]<>0
then   Memo1.Lines.add('Tub son='+inttostr(a[i]));  end;
```

verifies the functionality of the program by compiling the written program code. If there is an error in the program code, then the identified deficiencies are corrected; by switching the developed visual environment to active mode, the correct or incorrect operation of the program is checked by giving a value.

Through the above-mentioned problem task, it is possible to teach students not only programming, but also to repeat mathematical knowledge, to improve the skills of working in visual environments, to acquire the skills of analytical, logical and algorithmic thinking, and to increase the creative ability of preparing practical projects. Thus, the success of problem-based learning in teaching programming depends on the following factors: problematization of tasks related to programming; by creating problem situations, to activate the activities of numerical calculations and preparation of practical projects in the memory of students-students of programming languages; harmonizing the programming process with play and work; the ability of the teacher to use problematic methods appropriately and effectively; creating a chain of problematic questions about solving a programming problem and explaining it to students in a logical sequence.

An important feature of teaching programming is that the acquisition of relevant competencies depends not only on the ability to simply repeat existing knowledge, but should not be limited to the ability to apply template solutions. In fact, any real problem to be solved by a programmer requires non-standard thinking and non-standard actions. The methods of implementation of the problem project are selected depending on the specific content of the educational material. A problematic situation is created by clearly relying on the existing knowledge of some students. Based on them, students are taken contrary to the facts. Therefore, knowledge is not enough and additional information is needed to overcome the conflict. This version of the problem situation always has a high load on students, so cognitive efficiency is high.

A number of teachers and psychologists V. Okon V., I.Ya. Lerner, M.I. Makhmutov, T.V. Kudryavtsev [3] proposes a problem-based learning method to develop logical thinking. Problem-based methods are based on the active cognitive activity of students, which consists in creating problem situations, updating knowledge, analyzing, finding and solving complex problems that require the ability to see phenomena and patterns behind particular facts.

The teacher creates a problem situation, guides students to solve it, organizes the search for a solution. Thus, in the process of finding a solution, the child acquires new knowledge, he learns

new ways of acting. The problem situation is specially created by the teacher using a number of special methodological methods.

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

1. The level of non-independent activity - students' perception of the teacher's explanation, mastering the model of mental action in a problem situation, the student's independent work, exercises of the nature of reproduction, oral repetition;

In learning programming, non-independent activity is manifested in the study of ready-made texts of programs. For example, the following program code and project window are provided in the Borland C++ Builder programming environment to develop a program to generate a matrix A(N,N) whose diagonal elements are 1 and the rest are zero (N is a positive integer):

```
{
StringGrid1->ColCount=StrToInt(Edit1->Text);
StringGrid1->RowCount=StrToInt(Edit1->Text);
int i,j,n=StrToInt(Edit1->Text);
for(i=0;i<=n;i++)    for(j=0;j<=n;j++)
if ((i==j)||(i+j==n-1)) StringGrid1->Cells[i][j]=1; else
StringGrid1->Cells[i][j]=0;
}
```

2. The level of semi-independent activity is characterized by the participation of students in applying previous knowledge in a new situation and finding a way to solve the problem set by the teacher. This level is characterized by learning new material under the guidance of a teacher. As such tips, you can use program templates (For example, a program to create a matrix A(N,N) with diagonal elements of 1 and remaining elements of zero in the Borland C++ Builder programming environment (N is a positive integer)):

```
{
StringGrid1->ColCount=StrToInt(Edit1->Text);
StringGrid1->RowCount=StrToInt(Edit1->Text);
int i,j,n=StrToInt(Edit1->Text);
for(i=0;i<=n;i++)    for(j=0;j<=n;j++)
...
}
```

In doing so, students will create the main application code and the project window until they find:

```
if ((i==j)||(i+j==n-1)) StringGrid1->Cells[i][j]=1; else
StringGrid1->Cells[i][j]=0;
```

3. The level of independent activity is the performance of reproductive-research-type tasks, in which the student-student himself solves according to the text of the textbook, applies previous knowledge in a new situation, formulates and solves problems of an average complexity level and proves hypotheses with little help from the teacher, etc.;

For this level, it is necessary to put a task in front of students-students that they should solve without the help of the teacher. For example, a program in the Borland C++ Builder

programming environment to generate and transpose a matrix X(N,M) for given positive integers N and M.

```
{
int i,j,
n=CSpinEdit1->Value;  m=CSpinEdit2->Value;
for(i=0;i<n;i++)   for(j=0;j<m;j++)
StringGrid1->Cells[i][j]=random(10);
for(j=0;j<m;j++)    for(i=0;i<n;i++)
StringGrid2->Cells[j][i]=StringGrid1->Cells[i][j];
}
```

4. The level of creative activity is creative imagination, logical analysis and assumptions, finding a new way to solve an educational problem, performing independent work that requires proof, conclusions and generalizations based on one's own ideas.

At this level, the task must be set by the teacher, in solving it the pupil-student not only works independently, but also takes a creative approach to it. You can encourage creativity by using the project method element. An M-order square matrix is given as an example. The program to replace the elements lying in the lower part of its main diagonal with 0 (zero):

```
void __fastcall TForm1::CSpinEdit1Change(TObject *Sender)
{
 StringGrid1->RowCount=CSpinEdit1->Value;
StringGrid1->ColCount=CSpinEdit1->Value;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button3Click(TObject *Sender)
{
 int i,j,a=1,
n=CSpinEdit1->Value,m=n;
for(i=0;i<n;i++)  {
for(j=0;j<m;j++)
StringGrid1->Cells[j][i]=0;  m--; }}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button2Click(TObject *Sender)
{  int i,j,a=1,
n=CSpinEdit1->Value,m=n;
for(i=0;i<n;i++) { for(j=m-1;j<n;j++)
StringGrid1->Cells[j][i]=0; m--; } }
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{  int i,j,
n=CSpinEdit1->Value;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
StringGrid1->Cells[j][i]=random(10);
```

```
}
//---------------------------------------------------------------------------
```

At the same time, as an example of the level of creative activity, it is possible to cite the task of developing a project to calculate the determinant of order A(N,N) (where n=2,3,4) in the Delphi environment. Students try to prepare a project based on their ideas while completing these tasks. In this case, he develops a certain algorithm by repeatedly repeating the methods of calculating the determinant. At the next stage, the operator selects the buttons and enters the appropriate program codes into the buttons.

```
procedure Per(k,n:integer;var a:Tmatrix; var p:integer);
var z:Real;j,i:integer;
begin z:=abs(a[k,k]);i:=k;p:=0;
for j:=k+1 to n-1 do  begin
 if abs(a[j,k])>z then begin  z:=abs(a[j,k]); i:=j;  end; end;
if i>k then begin  p:=p+1; for j:=k to n-1 do begin
 z:=a[i,j];  a[i,j]:=a[k,j]; a[k,j]:=z; end; end; end;
function Znak(p:integer):integer;
begin if p mod 2=0 then result:=1 else result:=-1; end;
procedure Opr(n:integer;var a:tmatrix;var det:real);
var
k,i,j,p:integer; r:real;
begin det:=1.0;
for k:=0 to n-1 do begin   if a[k,k]=0 then Per(k,n,a,p);
 det:=znak(p)*det*a[k,k];
 for j:=k+1 to n-1 do  begin  r:=a[j,k]/a[k,k];
 for i:=k to n-1 do  a[j,i]:=a[j,i]-r*a[k,i];
 end;  end; end;
var k,j,i:integer;  a:Tmatrix;  det:real;
begin n:=strtoint(edit1.Text);
SetLength(a,n,n);
 for k:=0 to n-1 do   for j:=0 to n-1 do
 a[k,j]:=strtofloat(StringGrid1.Cells[j,k]);
Opr(n,a,det);
Edit2.Text:=FloatToStrF(det,ffFixed,5,0); end;
```

From the example proposed for the level of creative activity, it can be said that in order to solve the given problem, students first need to have mathematical knowledge, to master mathematical modeling and algorithm effectively, and to have experience in using the task buttons of the visual environment. At the same time, it requires independent research and creative thinking. As a result, the student's programming competence can be developed.

Thus, problem-based learning can be used in programming language classes using the above methods. *With the help of such tasks, students will have the opportunity to develop logical, algorithmic thinking about programming, learn to independently program problems and prepare complex practical projects.*

REFERENCES

1. Tolipova J.O. Biologiyani o‘qitishda pedagogik texnologiyalar // Oliy o'quv yurtlari talabalari uchun darslik. – Тошкент, 2011. – 160 b.
2. Серебреникова, В. В. Технология деятельностного подхода на уроках информатики при изучении темы «Программирование решений» // Молодой ученый. – 2019. – № 12 (250). – С. 288-290.
3. Баженова И. В. Визуализация знания как метод когнитивного подхода к обучению программированию // Решетневские чтения. 2014. – С.281- 285.