IMPROVING THE EFFECTIVENESS OF PROGRAMMING LECTURES BY APPLYING STRUCTURING METHODS FROM THE SCIENCE OF PROGRAMMING PARADIGMS

[1]Sanobar Khakimova, [2]Allamova Shaxlo

[1,2] Urgench Branch of Tashkent University of Information Technologies named after Muhammad al Khwarizmi

https://doi.org/10.5281/zenodo.7855438

Abstract. *In this article, we focus on various technologies currently used to facilitate the teaching and learning of computer programming courses in higher education. Programming is an important subject in computer science and information systems development, and today the demand for it in the computer industry is increasing day by day. New application programs are created and existing applications are recoded on demand. Smartphone and mobile phone sales have a huge audience today and users need relevant software applications for their daily work. In addition, the software development industry needs skilled programmers who know the required programming language. However, there is a shortage of programmers due to various reasons. One of the reasons is that learning programming is a complex process. Universities need to inspire students' interest in programming and integrate innovative technologies into their programming courses like never before to encourage students to become qualified programmers. Today, there are challenges in the use of technology in the educational process, and in the future, effective solutions for the integration of technology into university-level programming courses must be included.*

Keywords: *educational technology, programming course, teaching and learning programming.*

1.INTRODUCTION

The Government of the Republic of Uzbekistan allocates large funds for targeted training of specialists in the field of information technology and computer programming and financing of their start-up projects. These measures, introduced by the government, challenge universities to produce qualified programmers for our society. Today, there are some research works by Western and Russian scientists aimed at developing methodological approaches to teaching programming, including: activity approach [2], systematic approach [1], cognitive approach [6], semiotic approach [4], problem approach [5] . Many researchers in different countries of the world have conducted research on the application of innovations in education. They have collected valuable information about "innovation", "interactive methods", "innovative technologies", but there are not enough specific recommendations to increase the effectiveness of teaching "Programming methods and paradigms" course in technical universities.

The method of logic-graphic diagram implies the goal of achieving high results in a short time without spending too much mental and physical effort, and it is more effective in teaching "Programming methods and paradigms", "Programming" and "Algorithm design" courses taught in technical universities. Viewing objects is an important concept of the logic-graphic diagram approach. It aims at revealing the integrity of the object, identifying various connections in it and bringing them to a unified theoretical view. Effective sequence material created with the help of a logic-graphic diagram was developed in "Object-oriented programming paradigm" section of "Programming methods and paradigms" course, categorized by special tasks, test marketing tools, and in-depth subjects.

The technical university aims at creating automated systems for students, independent studies, and directly in order to increase the efficiency of the educational process. Using the built-in pedagogical software, the student studies the texts of lectures, laboratory work, seminars, independent work and practically determines his work on the topic, checks his knowledge using examples of specially classified test tasks.

A system is a set of elements that interact and communicate with each other, forming a certain unit [8]. In addition, researchers of the systematic approach [9] emphasize that the system is a set of objects, and their interaction causes the emergence of new integrative qualities that are not characteristic of individual components that make up the system. [8].

A structure is a set of stable connections between elements that ensure the unique integrity and uniqueness of an object, that is, the preservation of the main characteristics during various external and internal changes.

It should be noted that the systematic approach to knowledge and change of any object is the leading general scientific approach. The application of this approach in teaching programming is such a component as a system of teaching programming with all its features: integrity, communication, structure and organization, system levels and their hierarchy, management, self-organization of the system; its activity and development are necessary.

In the context of teaching Programming methods and paradigms, the main idea of a systematic approach is to consider educational tasks in close connection with programming tools, methods, and the technological process as a whole. Each of these areas should not be studied separately, but in close connection with each other.

Taking into account the interdisciplinarity of Programming methods and paradigms with other technical university disciplines and scientific knowledge fields, it can be stated that a systematic approach is central to its teaching, and other methodological approaches that can be used to teach programming paradigms should complement and extend its core ideas and principles, and not contradict it.

The activity approach in teaching is based on the "fundamental position that the human psyche is inextricably linked with its activity and its activity is conditioned." The activity category is central to this approach. At the same time, activity means the intentional activity of a person manifested in the process of his interaction with the outside world, and this interaction consists in solving vital tasks that determine the existence and development of a person.

The problem of selection and systematization of educational material components has been widely discussed by pedagogues, experts and scientists for a long time. Currently, many models represent the logical structure of educational material. Despite the nature of these diverse models, they have successfully passed experiments in a real pedagogical process based on practical methods, approaches, and have given positive results. As a result of the analysis of the essence of the educational process, many authors emphasize that it has a dual character [7, 8, 9, 10].

The content of the structure of the educational material, in particular, the nature of the development of education applied in the form of "problems" in programming science, and the description of the related organizational and didactic structures. Taking into consideration the problematic nature of thinking, the essence of a specific topic or section is as follows, that is, "In the form of a logical sequence of cognitive questions, and the educational process as a chain of educational situations, its questions as the core of its knowledge, and the essence of students is the joint, harmonious work in problem solving using teaching methods and various educational tools".

The concept of an activity approach in teaching Programming methods and paradigms includes:

- forming the student's readiness to actively, adequately and effectively use information and telecommunication technologies in the process of teaching the subject;
- identifying and forming the creative individuality of the student;
- development of future professional views.

Under universal educational efforts, in a broad sense, "the ability to learn, that is, the subject's ability to develop and improve himself through conscious and active assimilation of new social experience" is understood.

Cognitive views take priority in improving the effectiveness of learning programming paradigms. Cognitive universal educational activity constitutes a system of ways of knowing the surrounding world, independent search, research process and a set of operations for processing, systematizing, summarizing and using the received information. In this, general educational activities, symbolic, logical activities, problem setting and problem-solving activities are distinguished.

The formation of universal actions of general education allows students to:

- identify and formulate the problem independently, determine the goal, tasks, find the necessary information, search for a solution to the problem;
- form knowledge;
- conscious use of programming terms in speech in various fields of science;
- choose the most effective way to solve the programming problem depending on the specific conditions.

Significant and symbolic actions play a special role in increasing the effectiveness of teaching programming, which provide concrete methods of changing the educational material, represent modeling actions that perform the function of demonstrating the educational material. These include:

- replacement - transferring important features of the object to the model;
- coding - writing an algorithm in a language understandable to the executor;
- schematization - representation of the results of the previous two stages in a visual, perceptually simple form.
-modeling - the transformation of an object from a sensual form into a model (spatial-graphic or sign-symbolic), where important features of the object are highlighted.

Logical universal actions play an important role in choosing convenient methods for solving problems, in determining optimal algorithms. Their level of development affects students' ability to perform tasks of transferring the notation of a problem-solving algorithm from one symbol system to another. These tasks include:

- analysis of the main and non-essential parts of objects to highlight their features;
- synthesis - assembly of parts into a whole by filling in missing components;
- selection of bases and criteria for classification, comparison, categorization of objects;
- summarizing the concept, publishing results

The diversity of the content of each of the listed educational activities creates the necessary conditions for solving programming problems. The teacher must make sure that the students have all the necessary system of actions that constitute the programming ability. For this, it is necessary to connect the stages of solving programming problems with the structure of action:

1) The subject of action. The subject can be both a material object and words, expressions of the concept;

2) The purpose of the action. In order to successfully master the action, students must be taught to accomplish the goal. The goal is inextricably linked with the motive of action.

3) Motive. It motivates a person to set various goals and to achieve them, to perform appropriate actions. If the student does not see the need to perform certain actions, this will make the educational activity not interesting for him, he will not see any meaning in it.

4) The system of operations. Action includes several operations that must be performed in a certain sequence: action algorithm. In some cases this algorithm is fixed, and in others it can be changed.

5) A guiding basis is a system of conditions on which a person actually relies when performing an action. It may be complete or incomplete, true or false; the effectiveness of problem solving depends on how complete and correct the indicative framework is.

6) Result. The result of an action is always some product (tangible or hypothetical) and may or may not be consistent with the goal. The ratio of the result and the goal is determined.

We will consider the method of using structural logic in lectures on the topic "Object-oriented programming paradigm" in "Programming methods and paradigms" course. The teacher is provided with a technological map of the subject and a presentation on the topic. The student is given the text of the lecture: plans, main concepts, theoretical part, control questions, literature on the topic, documentary form of the lecture, audio and video lecture format, examples on the topic, tests and basic concepts. The learning objectives of the course are to introduce students to the Object Oriented Programming Paradigm. A programming paradigm is a mathematical theory or consistent set of principles for a computer programming approach. Each paradigm supports a set of concepts that are used to program a particular problem. For example, using object-oriented programming is the preferred method for problem solving.

The educational aim of the lesson is to provide students with in-depth knowledge about their professional activities, to form a sense of duty and responsibility towards society.

The objectives of the course are to form students' ability to apply knowledge, develop logical thinking and independent work skills. Methods of using logic-graphic diagram, consists of a 7-stage teaching methodology of the lecture class on the topic "Object-oriented programming paradigm" in the subject "Programming styles and paradigms".

THE FIRST STAGE. In preparing for the lesson, preparing students for the lesson, knowing the theory of the subject, using information and communication technologies; to know certain information; checking and analyzing homework. In addition, at this stage, it is necessary to activate the theoretical knowledge necessary for students in lectures. In order to do this, information is displayed simultaneously to students on all parallel displays.

The teacher launches files on the topic of the lecture. Students watch the activity of the teacher with concentration. The main goal is to focus students' attention, and it can be limited to 5-10 minutes.

THE SECOND STAGE. The teacher announces the topic, introduces the students to the goals and tasks of the subject, the sequence of the topic plan, the main concepts, the content of the lecture, and the results expected from the lecture. Explains the relationship between current and past topics. Students listen to the information given by the teacher, and the teacher should motivate the students based on the information given. This stage lasts 10 minutes.

THE THIRD STAGE. At this stage, the teacher explains the topic to the students using a logic- graphic diagram. To increase training efficiency it is appropriate:

- abandoning the uniformity of methods used in training;
- use of adequate elements of the program in the educational process;
- ensure rational use of time;
- effective use of problematic educational elements;
- use of video and multimedia tools in the course of the lesson

During the lesson, the speech of the teacher should be emotional, simple and understandable. The focus should be on explaining key phrases and new concepts rather than using unnecessary words. The main goal of this stage is to systematically provide students with basic expressions and scientific information, to develop social cooperation and creative interest in them. The duration of this stage is 25 minutes.

THE FOURTH STAGE. At the analysis stage, the teacher creates conditions for intensive analysis of ideas presented by students. Students analyze some aspects of the data and determine the most appropriate solution through intensive analysis of ideas and thoughts. At this stage, students are required to boldly express their personal opinion by increasing academic activity during the lesson, developing free, logical thinking. Students use every opportunity to think broadly and understand the essence of the problem. The duration of the stage is 15 minutes. THE FIFTH STAGE. During the supervision phase, students are observed with the main concepts of the lecture, program algorithms and methods of acquiring new knowledge. In this case, the teacher uses tests and examples divided into paper (handouts) or electronic (via computers) options according to their complexity. Students learn algorithms on the subject and develop a sense of error correction, activity, creativity and a desire to learn more about them. The duration of this stage is 15 minutes.

THE SIXTH STAGE. At the assignment stage, students are shown examples of creating programs of problems, as well as they are given the task of independently creating programs of relevant problems. In addition, it is necessary to collect the necessary materials from the Internet, analyze them, find useful sites on the topic, create logical graphic design solutions and use them. At this stage, students should understand the tasks and feel responsible for their completion. The duration of this stage is 5 minutes.

THE SEVENTH STAGE. At the last stage, the teacher provides information about the basic concepts of the topic and concludes the lesson. At this stage of the lecture, the main attention is paid to the analysis of the work done. In this process, the student evaluates his success, that is, in the electronic version; the teacher can send his suggestions and requests or answer the questionnaire, give methodological instructions for the topic of the next lecture.

The learning objectives of the course are to introduce students to the Object Oriented Programming Paradigm, basic flow charts, key concepts, and critical problem solving.

The educational aim of the lesson is to provide students with knowledge about the professional activity of programming, to form their sense of duty and responsibility before society.

The objectives of the course are to form students' ability to apply knowledge, to create a program independently, and to develop their logical thinking skills.

Expected learning outcomes.

Teacher: managing the lesson process, instilling new knowledge in students, not getting bored, constantly monitoring, achieving a lot in a short time, increasing students' logical thinkingand heuristic activity, and fair assessment.

Student: learn to systematize acquired knowledge, increase the ability to acquire, consolidate and evaluate new knowledge, develop independent and logical thinking, and the ability to work in a team.

Form of the lesson: lecture (80 minutes).

Method of the lesson: Working on a computer.

Course: Existing programming paradigms, with closely related language classes and methodologies, are widely accepted as a basis for differentiation.

For example, the object-oriented paradigm.

1) Organizational part. Motivation: interest in creating efficient algorithms and task programs based on programming styles and paradigms;

2) Activation of knowledge: all students are given a creative task.

Equipment: electronic version of the topic, computer, presentations, projector, simple typical examples, tests and answer sheets (on paper).

Course:

Stage 1. Preparation for the lesson (5 minutes).

Educational activities:

1) Launches an electronic version prepared on the topic.

2) Begins to describe the topic using presentations.

Student activity: observes and summarizes the activity of the teacher.

Expected results: student concentration.

Stage 2. Generalization of knowledge on the previous topic and introduction to a new topic (10 minutes).

Teacher activity: briefly repeats the main parts of the previous topic, explains the statistical analysis of the test results. Announces a new topic and provides information about its relevance, introduces students to the goals and tasks of the subject, the main concepts of the thematic plans, the content of the lecture, and the expected results. In the presented slides, the teacher explains the purpose and tasks of the lesson, its connection with other subjects and the importance of the topic.

Students' activities:

Object-oriented programming defines the following basic concepts related to the object: class, object, property and method content, essence, goals and tasks, algorithm and the role of programming in the development of our society.

Expected results: formation of strong motivation in students to model, algorithmize and program practical tasks.

Stage 3. The topic is explained and students are activated (45 minutes).

Teaching: Students will be taught that object-oriented programming is a key paradigm in modern application development and is characterized by the representation of the object domain in the form of objects that combine data and behavior. Gives an understanding of classes. Object Oriented Programming is explained in Table 1 below.

Table 1.

Solving problems with object-oriented programming concepts.

№	Problem setting	Problem solving	Key words
1	Create a class named "myClass".		Class
2	Create an accessible integer attribute "myNum" and an inaccessible string attribute "myString" from outside the class.		public private
3	Create functions in the class "myMethod1" that return the value of the attribute "myNum" and "myMethod2" in the class that does not return a value and outputs the word "Hello" to the console.		int myMethod1() void myMethod2()
4	Create an object named "myObj" and access the attributes		myObj.myNum
5	Assign values to the attributes in the class through the constructor		myClass(){…}
6	Let the Car class (child) inherit attributes and methods from the Vehicle class (parent).		class Car**:** public Vehicle

Feedback incudes opinions from listeners, and controls their activity. Students are asked questions such as "program", "algorithmization", "object", "class", "program code", "integration". Several students answer the questions. The teacher summarizes the answers and explains with examples.

Student activity: students express their thoughts and opinions, acquire new knowledge.

Expected results: increase students' ability to successfully apply logical concepts in practice, algorithm and programming of repetitive calculation processes, development of creative interest, logical thinking, teamwork, and social cooperation.

The teacher asks students such questions as "class", "constructor", "creation of objects", "inheritance", "polymorphism", "encapsulation" (Fig. 1)

Table 2.

Analyzing concepts

By Lassi Ceska analyzing concepts	Private express solutions

Teacher: What programming paradigms do you know?

Student 1: functional paradigm, declarative paradigm, imperative paradigm, logical paradigm, structural paradigm, object-oriented paradigm.

Student 2: In the object-oriented paradigm, a program is presented in the form of interacting objects. Procedural programming is writing procedures or functions that perform operations on data, while object-oriented programming is creating objects that contain data and functions.

Student 3: Based on the logic programming paradigm, a program is created using formal logic rules, and in the functional programming paradigm, the process of creating a program is expressed based on a sequence of calculating values for mathematical functions.

Teacher: What are classes and objects in object-oriented programming?

Student 4: A class is a template for objects, and an object is an instance of a class. When individual objects are created, they inherit all variables and functions from the class. For example: in real life, a car is an object. The vehicle has attributes such as weight and color and methods such as drive and brakes.

Student 5: Attributes and methods are basically variables and functions that belong to a class. They are often referred to as "class members". A class is a user-defined data type that we can use in our program, and it acts as an object constructor or "blueprint" for creating objects.

Teacher: Using slides in the form of 12 structures and with detailed examples in the form of codes, the teacher explains that object-oriented programming is considered as the main paradigm in the development of modern application programs, and that its feature is the representation of the subject area in the form of objects that combine data and behavior.

Teacher: explains the main structures of algorithmic and programming of computational processes (Fig. 2) and their essence with the help of the electronic version of the lecture and presentations (Fig. 3).

Figure 3. Classification of the organization

In a specific example, we consider the creation of various structures of algorithms and the programming of simple integrator ion calculation processes.

Example:

The specification of the semantic graph by topic is presented in Table 3, and the structure of the semantic graph is presented in Table 4.

Table 3.

The specification of the semantic graph by topics

Subject	Nodes count	Training elements. Basic concepts
1. Processes of using classes, objects, constructors and inheritance in object-oriented programming	1.1	Access specifier: public
	1.2	Attribute (int variable): int myNum;
	1.3	Attribute (string variable): string myString;
	2	Object
	2.1	Create an object of MyClass: MyClass myObj**;**
	2.2	Access attributes and set values: myObj**.**myNum = 15; myObj**.**myString = "Some text"
	3	Methods
	3.1	Method/function defined inside the class: void myMethod() { cout << "Hello World!"; }
	4	Constructor
	4.1	Constructor parameters: Car(string x,string y, int z)
	4.1	Constructor: MyClass() {cout << "Hello World!"; }
	4.2	Create an object of MyClass (this will call the constructor): MyClass myObj;
	4.3	~ExampleClass()
	5	The meaning of Encapsulation, is to make sure that "sensitive" data is hidden from users
	5.1	Increased security of data
	6	Inheritance:class Car: public Vehicle
	7	Polymorphism uses those methods to perform different tasks.

Didactic materials (text, graphic, animation, multimedia) are prepared for each educational element, logically completed concepts explaining the essence of cooperation.

Table 4.

The structure of the semantic graph.

Program	// Calculation of the student's average score
	#include <string>
	class Students
	{ public:
	// Setting the student's name
	void set_name(std::string student_name)
	{ name = student_name; }
	// Getting a student's name
	std::string get_name()
	{ return name; }
	// Setting the student's last name

```
void set_last_name(std::string student_last_name)
{ last_name = student_last_name; }
            // Obtaining a student's last name
std::string get_last_name()
{ return last_name; }
           // Setting intermediate grades
void set_scores(int student_scores[])
{ for (int i = 0; i < 5; ++i)
{ scores[i] = student_scores[i]; } }
              // Setting the average score
void set_average_ball(float ball) { average_ball = ball; }
              // Getting an average score
float get_average_ball()
{ return average_ball;  }
private:
               // Interim estimates
int scores[5];
               // Average score
float average_ball;
               // Name
std::string name;
               // Surname
std::string last_name; };

#include <iostream>
int main()
{
                   // Creative object class Student
Students student;
std::string name;
std::string last_name;
                   // Entering a Name Using the Keyboard
std::cout << "Name: ";
getline(std::cin, name);
                    // Entering a last name
std::cout << "Last name: ";
getline(std::cin, last_name);
                    // Storing the first and last name in an object of class Students
student.set_name(name);
student.set_last_name(last_name);
                    // Ratings
 int scores[5];
                    // Sum of all grades
int sum = 0;
```

```
                    // Entering intermediate grades
    for (int i = 0; i < 5; ++i)
    { std::cout << "Score " << i+1 << ": "; std::cin >> scores[i];
                    // summation
    sum += scores[i]; }
                    // We save intermediate grades in an object of the Student class
    student.set_scores(scores);
                    // Calculate the average score
    float average_ball = sum / 5.0;
                    // We store the average score in an object of class Students
    student.set_average_ball(average_ball);
                    // Retrieving student data
    std::cout << "Average ball for " << student.get_name() << " " << student.get_last_name() << " is " << student.get_average_ball() << std::endl;
    return 0; }
```

The class is created. Calculations are performed using the attributes and functions defined in it.

The remaining 11 class structure options are explained in detail using the presentation, the differences between the structures are explained, analyzed, and the advantages of object-oriented programming are discussed.

Teacher: What are the advantages and disadvantages of the structure used by the if command in organizing simple calculation processes?

Student 6: The advantage of this structure is that the value of the composite parameter iteration is expressed as needed during the iteration process. “Goto m” is used instead of the conditional transition operator to return to the iteration, this command can be a source of error in many cases. Programmers are encouraged to avoid using conditional transition operators whenever possible.

Teacher: What is the purpose of encapsulation?

Student 7: It is a good practice to declare class attributes private (as often as possible). Encapsulation gives you better control over your data because you can change one part of your code without affecting other parts. As a result, data security is increased.

Teacher: What are the differences in input specifications?

Student 8: In the C++ programming language, public class members can be accessed outside the class; and private access specification is used when members can only be accessed inside the class.

Teacher: With the increase in the use of mobile and smart phones, the demand for mobile application development and related programming skills is increasing. Education, business, medical, government, transportation, tourism and other industries are in high demand for related software applications and hence need more programmers who know latest programming platforms and languages today. It is explained that nowadays, algorithms and programming play an important role in the effective organization of the information process of society, in solving the technical and economic problems that arise in various production enterprises and organizations, and it is emphasized that the use of modern technologies in every field, and the development of their software is one of the urgent issues.

Teacher: In the future, automation of control will become one of the most important issues in almost all enterprises, organizations and offices, and its foundations are explained with real examples of algorithms and object-oriented programming. It states that the aim of object-oriented programming is to learn in detail how to create classes, objects, inheritance, and constructors.

Stage 4. Analysis (10 minutes)

The teacher answers the students' questions. Explains technical concepts with as many real examples as possible. At the same time, the students themselves answer the tasks, the teacher concludes.

Stage 5. Control (10 minutes)

The teacher distributes the tests, students mark their answers on the control sheet and the answer sheet.

The tests are based on the core concepts of Algorithm and Programming Fundamentals.

1. What is the output of the following program?

```
#include <iostream>
#include <string>
using namespace std;
class MyClass {
  public:
    int myNum;
    string myString;  };
int main() {
  MyClass myObj;    myObj.myNum = 15;
  myObj.myString = "Some text";
    cout << myObj.myNum <<" “;
  cout << myObj.myString;
  return 0; }
```

A)Some text   15   B) Some text   C) 15 Some text   D) 15

2. “How to use value 15 for myClass” “myNum” attribute?

```
class MyClass
{ public:       int myNum;};
int main () {  MyClass.myObj ;
  ___________.__________  ____  _____;
 cout < < myObj.myNum;
 return 0;
}
```

A)   myNum. myObj=15 B) myObj.myNum=15 C) myObj=15 D) myClass.myNum=15

3. Fill in the blanks on how to create a myMethod function in Class and call myMethod using myObj.

```
Class MyClass
{public:
_______() {cout<<”Hello”;} };
int main() { MyClass myObj;
```

__________.____________();

return 0;}

A)myMethod, myObj, myMethod B) myObj,myMethod, myMethod

C)myMethod, myObj, myObj D)myMethod, myMethod, myObj

4. Encapsulation is a mechanism for combining givens and their …

A) processing code B) sorting code C) code D) processing variable.

5. Which keyword is used to create a class?

A) create B) public C) clas D) class

6. The C++ language supports object-oriented programming principles. Which of the following are these principles?

A) Encapsulation B) Inheritance C) Polymorphism D) All are correct

7. … – is a variety of forms.

A) Public B) Encapsulation C) Void D) Polymorphism

8. … C++ provides the ability to perform different actions when a function with the same name is used by different objects.

A) Polymorphism B) Encapsulation C) Public D) Private

9. In order to access and modify class attributes and methods from outside the class, the … keyword is used

A)private B) protected C) public D) inherit

10. To prevent access and modification of class attributes and methods from outside the class, the ... keyword is used

A)private B) virtual C) public D) inherit

Use the class keyword to create a class:

Complex engineering problems particularly arise with the help of telematic artificial thinking systems in the form of organizational "human-computer" systems, "robotic" systems and "human" productions and technologies that combine the latest communication tools based on programming algorithms and methods.

Test results on the topic "Object-oriented programming paradigm".

Name._______________________ ___ group ______________

Questions	1	2	3	4	5	6	7	8	9	10
The answers										

The teacher collects the answer sheet, analyzes it and shows the wrong answers. The correct answers are shown on the slide. Students revise their answers.

Criteria for evaluating test answers: 6-7 tests - "satisfactory"; 8 tests - "good"; 9-10 test - "excellent".

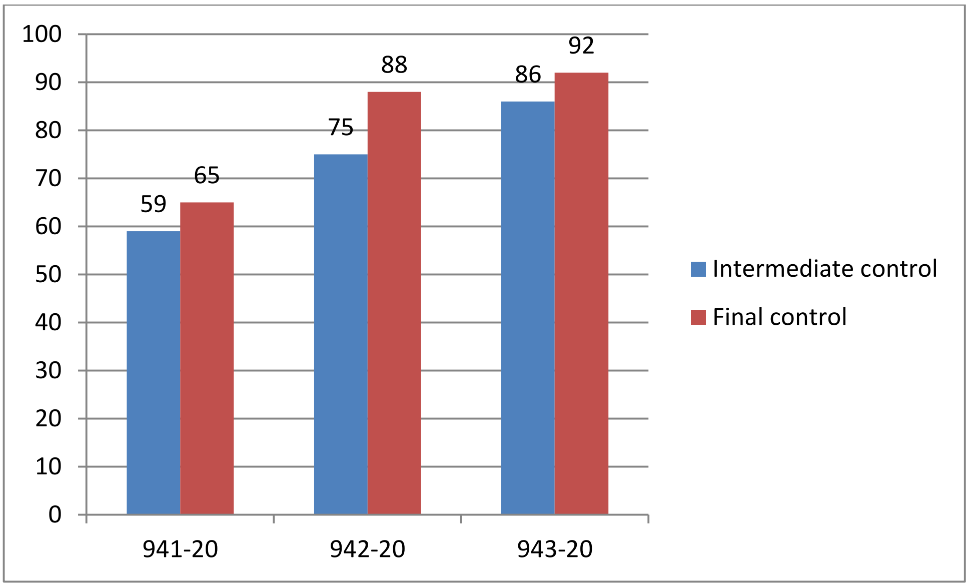
At this stage, the basic concepts of the course are mastered by the students, the feelings of activity and creativity are formed in the students, the desire to correct their mistakes and get better knowledge increases.

Stage 6. Assignment (3 minutes). Students are provided with a brief explanation of the homework assignment on the Moodle platform with the assignment address.

Stage 7. Wrap up (5 minutes). The topic is summarized before the end of the lesson. At the end of the lesson, the teacher instructs the students to write their thoughts and opinions on the questionnaire. Students fill out a specially prepared questionnaire and the lesson is evaluated in written form. The teacher collects the questionnaires and finishes the lesson.

ANALYSIS OF THE RESULTS OF THE PEDAGOGICAL EXPERIMENT.

To evaluate the effectiveness of the above methods in teaching “Programming paradigms” course in higher education, 23 students from Group 941-20, 24 students from group 942-20, and 13 students from Group 943-20 were selected. In this case, classes in "Programming methods and paradigms" course was conducted in the form of a simple lecture in Group 941-20, and in the rest of the groups, classes were conducted using innovative technologies. Based on the results of the mid-term and final evaluation, the following indicators were recorded in the groups.



Picture 1. Science mastery indicators of group students.

As it can be seen from the experiment, teaching the course in the form of a traditional lecture recorded a 65% indicator in the final evaluation, while the groups trained on the basis of innovative technologies achieved 87 and 91% indicators. . Furthermore, the students of Urganch branch of TUIT named after Muhammad al-Khwarizmi participated in ICPC 2022-23 International Algorithmic Programming Olympiad in Kazakhstan, and 2 teams actively participated in the semi-finals of the Olympiad. In addition, students are taking pride of place in Informatics and information technologies at the Republican Olympiads. We have come to the conclusion that the use of new innovative methods to achieve high efficiency in higher education in teaching programming paradigms course is highly effective.

2. CONCLUSIONS

Thus, object-oriented programming is the main paradigm in modern application development. Its feature is the representation of the object domain in the form of objects that combine data and behavior. Based on the developed logical structure of the subject (discipline, department, module, block), programming according to the methodology of teaching students in the programming courses of the university helps to implement the modern concept of education in the field of information technologies, to develop modern teaching methods. The modern electronic resource base, created on the basis of the logical structure of the course, not only increases the interest of university students in the studied subjects, but also allows the teachers themselves to preserve the achievements in the arsenal of educational subjects.

An example topic in our structured method, the "Object Oriented Programming Paradigm" section of "Algorithms and Programming Fundamentals" course, along with experimental verification, showed educational effectiveness in teaching these information technologies and improving the systematic knowledge of university students.The selection and composition of the content of modern educational science and the topic "Object-oriented programming paradigm" (a collection of terms and concepts, facts, types of educational and research activities) provides interaction between the subject and methods of teaching, science methodology and the field of knowledge. Orientation of educational materials to the formation of an integrated system of professional and educational skills increases the quality of higher professional education.

REFERENCES

1. Beshenkov S.A., Lyskova V.Yu., Rakitina E.A. “Information and Information Processes”. Informatics and Education, No. 6-8, 1998. P.10-19.
2. Odintsov I. “Professional programming. systematic approach"
3. Publisher: SPb.: BKhV-Peterburg: 2002 Pages: 512, Pages 19-27,121-134.
4. Marcelo, C., Yot-Domínguez, From chalk to keyboard in higher education classrooms: changes and coherence when integrating technological knowledge into pedagogical content knowledge, Journal of Further and Higher Education, 2019, Pages 975-988.
5. Grebneva D. “Review of Methodological Approaches to Teaching Programming at School”. Scientific review. Pedagogical Sciences. 2016. No. 3. P. 13-27.
6. Karelin A. O., Pavlova D. V. Comparative evaluation of the flow of information through traditional and modern sources in student learning. - News of SPbSTU “LETI”. - SPb., 2016, No. 6. - p.30-35.
7. Alexandrova V.V., Zaitseva A. A. 3-D technology and cognitive programming / Information-measuring and control systems. No. 5, vol. 10, 2012. P.61-64.
8. Butakov S.A. Structuring of educational material in accordance with the principle of ascent from abstract to concrete / Abstract. Diss. candidate ped. sciences. - Magnitagorsk, 2001.
9. Ermakov A.V. Multidimensional structuring of educational material as an example of innovative teaching physics in a university / A.V. Ermakov // Physical education in high schools - M.: Publishing House. MFO, 2007. - T13, No. 4. - S. 05-112.
10. Yusupov F. Improving the effectiveness of teaching the course "programming basics" based on the structuring of the discipline. [Text] / F.Yusupov, I.I. Shamuratova, N.Kh.Sapaeva // Young scientist. - 2016. - No. 9.5 (113.5). - S. 73-75.
11. Yusupov F. Structurization of the theme “A square function analysis ” based on the logical graph of the scheme [Text] / F. Yusupov, II Shamuratova // Young scientist. - 201 5 . - No. 9 . - S. 1229-1233.
12. Yusupov F. On the organization of students' independent work in a technical university in the context of ICT [Text] / F.Yusupov, U. Sapaev // Young scientist. - 2016. - No. 9.5 (113.5). - S. 83-85.