Volume 2, Issue 12, December 2024 **ISSN (E):** 2938-3757

Web of Technology: Multidimensional Research Journal
webofjournals.com/index.php/4

LEARNING THEORIES AND THEIR APPLICATION IN THE CONTEXT OF PROGRAMMING LANGUAGES

Komilova Zulkhumor Khokimovna
Fergana State University
Department of Information Technologies

Abstract
This article analyzes the theories of learning - behaviorism, cognitivism and constructivism and their application in teaching programming languages. On the basis of these theories, scientific applications of software tools and methods for effective organization of the educational process are considered. It also provides information about the possibilities of developing students' algorithmic thinking skills through programming languages and modern platforms used to automate such a process. Scientific and practical foundations are explored, and theoretical approaches are shown in practice.

Keywords: Learning theories, behaviorism, cognitivism, constructivism, programming languages, algorithmic thinking, educational technologies, interactive educational platforms.

Introduction

The modern education system is increasingly integrated into the digital environment. This process brings qualitatively new approaches to the education system on a global scale. Organization of the educational process with the help of digital technologies plays an important role in improving the quality of education. At the same time, digital tools are pushing to rethink traditional methods in education.

Today, programming languages are becoming an integral part of the educational process. They help students develop logical thinking and problem-solving skills while imparting technical knowledge. This creates wide opportunities not only in the field of software, but also in other fields. Therefore, combining programming with pedagogical approaches is one of the urgent issues of today's education.

For the successful use of programming languages in teaching, it is important to rely on the foundations of modern pedagogical theories, in particular behaviorism, cognitivism and constructivism. These theories offer science-based approaches to effectively organize the educational process. By using these theories in teaching programming, it is possible to deepen students' knowledge and skills.

Programming is not only a technical skill, but is widely used in education as a tool for developing algorithmic and systematic thinking. By forming algorithmic thinking, students will have the opportunity to solve complex problems and master a systematic approach. This

approach helps to develop logical thinking not only in the field of software, but also in everyday life.
The integration of digital technologies and programming languages into the learning process contributes not only to the individual development of students, but also to the improvement of the effectiveness of educational institutions and pedagogical methods. In this process, interactive educational platforms, visual programming environment and problem-based teaching methods take a special place. Therefore, it is important to study learning theories and their implementation in the context of programming languages. This not only improves the quality of education, but also serves to form the digital skills of students and harmonize them with the requirements of the modern world.

Methods and Results

Learning theories are methodological frameworks that help explain the processes of knowledge acquisition, retention, and application. The following basic theories are used in the process of learning programming languages:
Bihiviorism is a theory that the teaching process is formed on the basis of external influences. For example, teaching through a step-by-step approach to programming and a reward system.
Cognitivism is a theory that supports a deep understanding of knowledge and the study of its internal processes. It is used in the study of the interdependence of algorithms and logical structures in programming languages.
Constructivism is a theory that students form their own knowledge based on their own experiences. For example, developing programming skills by creating independent projects.
Programming languages are used in education in the following areas:
Programming languages such as Python, Java, and C++ teach students a systematic approach to problem solving. The power and universality of these languages provide a basis for performing complex tasks at various levels. For example, Python's simplicity and powerful libraries are important for deep learning of algorithmic knowledge. In the process of programming, students learn to break complex problems into small parts and approach each part separately. This process helps them develop analytical thinking. At the same time, understanding the structure of software code improves their ability to solve problems effectively.
Programming can be learned in a simple and interesting way through interactive educational platforms such as Scratch and Code.org. These platforms convey the fundamentals of programming in an understandable and simple way for children and beginners. They allow you to learn programming visually. Young children begin building programs using visual blocks on Scratch and Code.org platforms. In this process, they develop algorithmic thinking skills. Also, the educational process will be interesting and practical with the help of interactive activities.
Designed for beginners to coding, the platform and programs allow students to work in a simplified environment. In the process, they easily learn to understand the basics of programming. For example, platforms like Code.org provide step-by-step courses and guide the student in the right direction. Through the use of interactive platforms, the cooperation between the student and the teacher in the educational process increases. This increases the

efficiency of mastering the educational material. Also, visual and practical exercises allow students to independently check and consolidate knowledge.
Interactive platforms are based on pedagogical theories and offer the following advantages. Below we explain each theory with specific examples.
Based on the theory of bihiviorism, interactive platforms use reward and immediate result display systems. For example, on the Code.org platform, students will have to complete an exercise called "Illuminate the Labyrinth". This exercise requires you to write some simple code to move the robot along a given path. After each correct step, the platform confirms success with immediate feedback such as "Sorry" or "All paths are correct". This method increases students' motivation and encourages them to practice more.
According to the theory of cognitivism, interactive platforms teach complex issues step by step and provide their analysis. For example, through the exercise "Bird Flight" on the Scratch platform, students learn to simulate the movement of birds. First, the concepts of motion and time are explained to them, and then they create an algorithm for bird flight using visual blocks. Such a step-by-step approach helps to master complex issues more easily.
Constructivism theory focuses on students' independent creation of knowledge on interactive platforms. For example, consider developing a Temperature Conversion project in Python. In this project, students create their own code to convert from Celsius to Fahrenheit and vice versa. They acquire new knowledge independently and develop a program that can be used in real life.
The application of theories in the educational process is as follows:
Bixiviorism, in which a scoring system is introduced for students to complete tasks as part of the Fundamentals of Algorithms exercise on the Code.org platform. For example, you can get 10 points for 5 correct steps. Such a system strengthens and encourages the desire of students to achieve high results.
Cognitivism, in which an activity on the topic "Movements of various creatures" is conducted on the Scratch platform. Students use visual blocks to identify animal movements and model their interactions. They learn the process of explaining the consistency of different actions with the help of diagrams.
Constructivism, in which students develop an urban transport simulation project. In this project, they write an algorithm to find the shortest route for passengers. In this process, they form new knowledge based on their experiences and learn to solve real-life problems independently.

Conclusion

The integration of learning theories and programming languages is important in improving educational effectiveness. The principles of behaviorism, cognitivism, and constructivism offer effective methods that can be used to teach programming. With the help of interactive platforms and modern technologies, the possibilities of developing algorithmic thinking skills of students and automating the educational process will expand.

Web of Technology: Multidimensional Research Journal
webofjournals.com/index.php/4

Reverences

1. Chen Zan, Chia Arthur, Bi Xiaofang. Promoting innovative learning in training and adult education-a Singapore Story // Studies in Continuing Education. 2020. Vol.43, Issue 2. – pp. 197-198.
2. Yang By Jin, Yorozu Rika. Building a Learning Society in Japan, the Republic of Korea and Singapore. – Paris: Publication Series on Lifelong Learning Policies and Strategies. UNESCO Institute for Lifelong Learning. 2015. - №2. – p. 29.
3. Narayanan, Krishnan. Micro-coaching as a blend to make e-learning more effective // Dissertations and Theses Collection (Open Access). – Klong Luang, Thailand: 2019. – p. 29.