European Science Methodical Journal
ISSN (E): 2938-3641
Volume 2, Issue 6, June - 2024

SHAPE STUDENTS ' KNOWLEDGE AND SKILLS BY CREATING APPLICATIONS IN THE PYTHON PROGRAMMING LANGUAGE

Mukhiddin Uljayevich Ashurov
Tashkent State Pedagogical University named after Nizomi
Senior Teacher of the Department of Informatics and its Teaching Methodology

Abstract:

This article discusses the problem of lack of time for teaching high school students object-oriented programming within the discipline “Computer Science” and its solution by developing an elective course aimed at in-depth study of the Python programming language through the creation of game applications. The course offers the development of two games: arcade and logic using the capabilities of the Pygame and Arcade libraries.

Keywords: object-oriented programming; Python; elective course; a game; Informatics.

Introduction

According to the definition of an authority in the field of object-oriented methods of program development, Gradi Bucha: “Object-oriented programming (OOP) is a programming methodology that is based on representing a program as a collection of objects, each of which is an implementation of a certain class (a special kind of type), and classes form a hierarchy on the principles of heritability.”

Currently, in the academic subject “Informatics” there is a tendency to reduce the importance of teaching the topic “The concept of an algorithm. Programming". The number of lessons devoted to studying this topic in high school has decreased. Most of the time is devoted to teaching topics in the “Information and Communication Technologies” cycle. At the same time, the requirements for the level of mastery of knowledge and skills in this section of the computer science program have not changed, since it remains the basis of fundamental knowledge in the subject.

An analysis of current teaching materials shows that the emphasis in teaching schoolchildren programming languages at school is still on procedural languages (Pascal, Basic). However, in recent years, the Python programming language has become increasingly popular, which is used to teach and create a wide variety of programs: from text dialogues to serious web applications and 3D games.

In-depth study of the topic “The concept of an algorithm. Programming" is always accompanied by the teaching of electives that contribute to achieving a sufficient theoretical and practical level of knowledge acquisition. In an educational environment, elective courses in computer science solve problems of in-depth study of individual topics and sections, which makes it possible to develop the profile level of the subject, and also helps to improve the level of students' preparation for passing the Unified State Exam. Also, elective courses contribute to the profile self-determination of students, increase the effectiveness of developing a specialist's preparedness to be able to apply the acquired knowledge in their professional activities.

Thus, on the one hand, there is a need for in-depth study of programming languages in the school computer science course, on the other hand, it can be stated that there is a shortage of hours allocated for this and insufficient development of methodological material for teaching the Python programming language, which determines the relevance of the research.

We have developed an elective course "Game Programming in Python", aimed at students of 10-11 specialized classes who are already familiar with the basics of programming in Python, and designed for 34 hours.

The purpose of creating this elective course is to consolidate skills in working with the basic constructs of the Python programming language, as well as to deepen and expand practical and theoretical knowledge on the topic of teaching "Algorithmization and Programming." In the process of studying, along with practical mastery of the programming language, students are expected to become familiar with logical and arcade types of computer games, their implementation and study of auxiliary libraries. The course is aimed at developing students' logical, algorithmic and systems thinking during the preparation of programs in a programming language, as well as developing creative abilities.

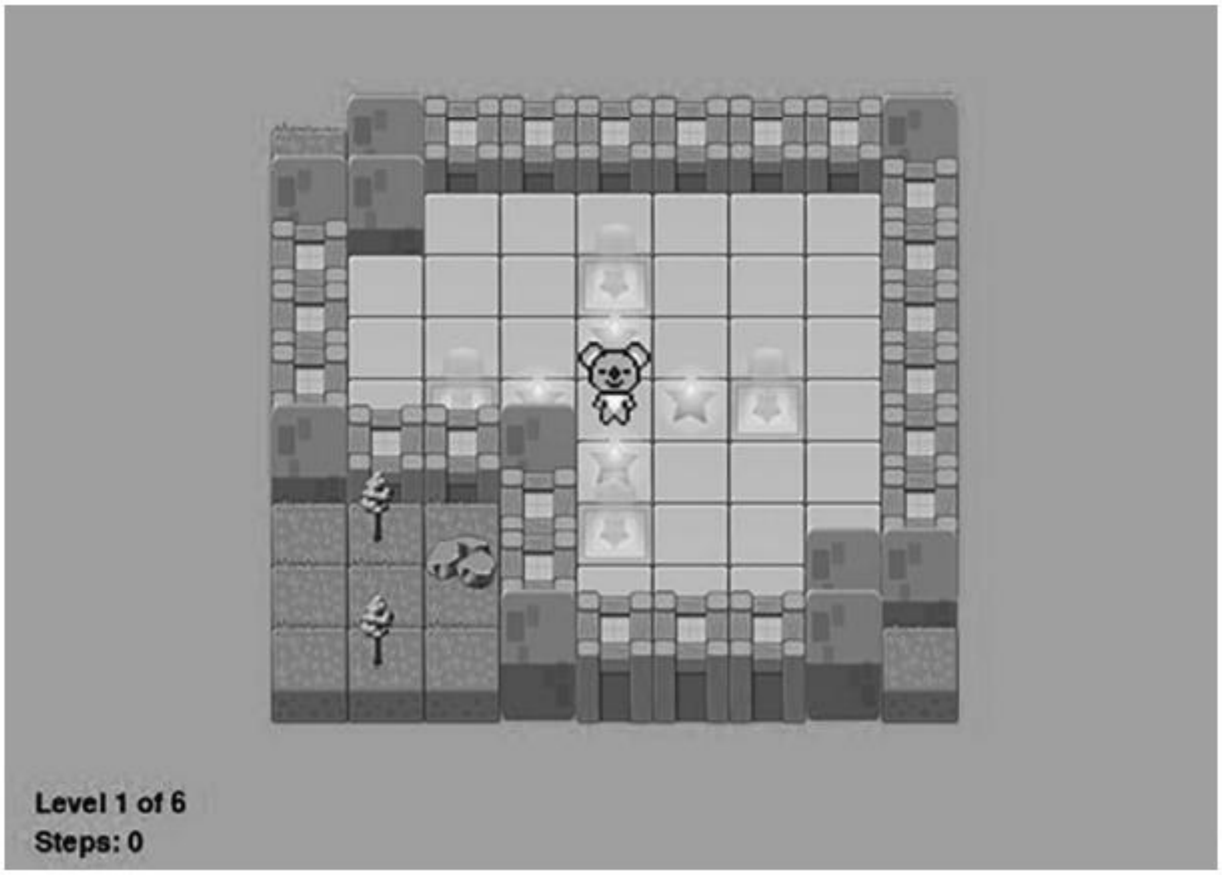
The main characteristic forms in the implementation of this program are combined classes, consisting of theoretical and practical parts, with the practical part taking up more time.

The elective course begins with a theoretical lesson about the capabilities and differences of the two libraries that will be used to create games, and also outlines the path for studying the course. At the beginning of most classes, a theory is given for the part of the code that will be developed in the lesson. Gradually, students perform practical work, creating two individual games: the first using the Pygame library (as it is the most popular and widespread today), and the second using the Arcade library (in order to show students the possibility of developing games by using more modern, convenient and easy-to-learn library, not inferior to others in its functionality). In the process of creating game applications, extracurricular time is allocated for searching and processing images for the graphic design of the game, music, as well as for developing levels and implementing other own ideas. Thus, after developing each game, students will have to defend it in front of the class, demonstrating the functionality and uniqueness of the game.

Interim control of knowledge acquisition will be carried out by checking students' practical work (developed code fragment). Final control will be carried out by protecting projects (developed games) after completion of work on each game.

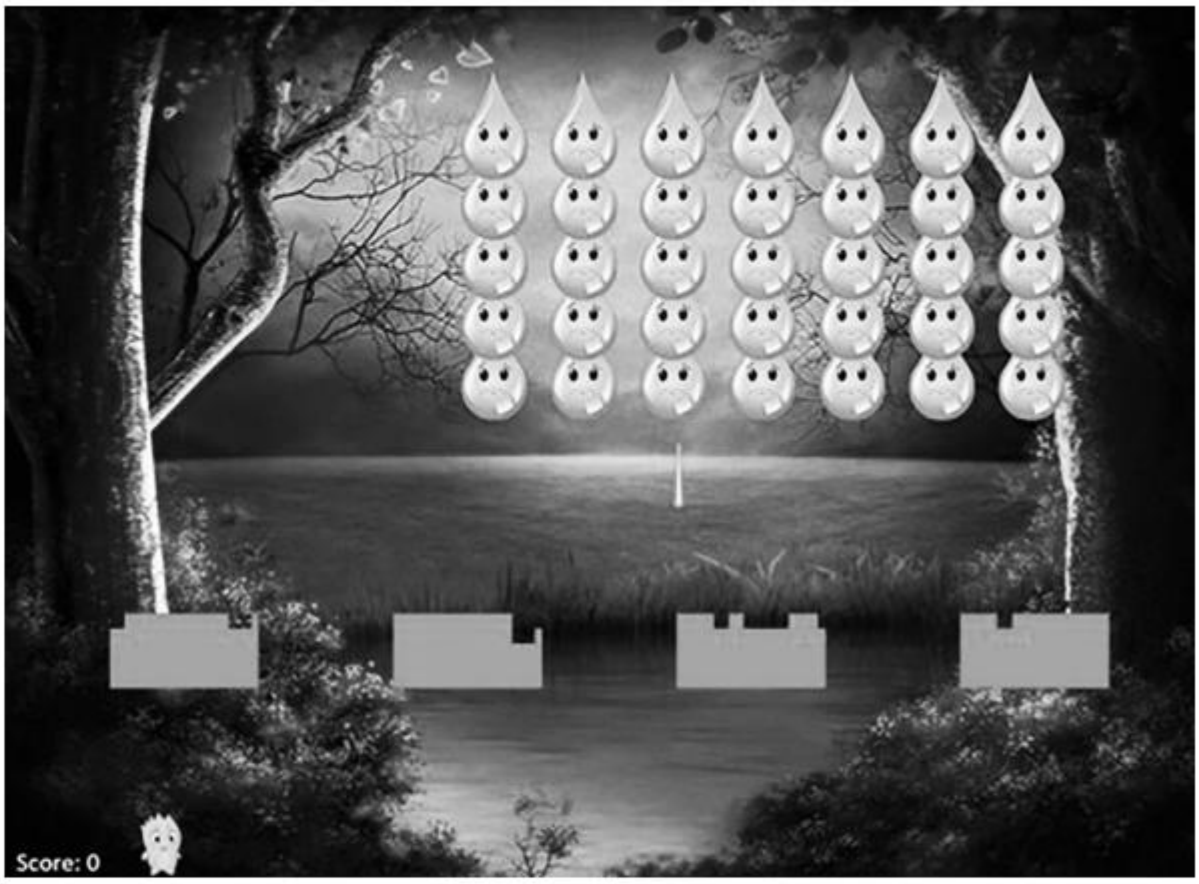
The elective course examines the development process of the logic game "Stars", implemented using the Pygame library, as well as the arcade game "Ogonyok", developed based on the Arcade library.

The essence of the game “Stars” is that on the playing field there are stars and the places where they need to be deployed. The movement of stars across the field is carried out with the help of a game character, whose movement occurs via the keyboard (Fig. 1).



Rice. 1. Example of the design of the game “Stars”

The essence of the game “Ogonyok” is that on the playing field there are the main character - Ogonyok, his enemies - a block of drops and blocks that protect the player from enemy bullets, but are destroyed by them. The character's movement is carried out by moving the mouse. The movement of enemies is determined by the program code. The goal of the game is to destroy all enemies (Fig. 2).



Rice. 2. An example of the design of the game "Ogonyok"

This elective course will help students gain additional programming skills. After completing the course, high school students will be able to create 2D games at a professional level. The knowledge gained will contribute to the formation of a worldview that corresponds to the modern level of development of science and technology, and will also help to make an informed choice of the field of future profession and opportunities for realizing one's own life plans.
One of the most interesting issues that require special attention in the process of studying the subject "Informatics and ICT" at school is increasing the level of student programming. Currently, the emphasis in teaching students programming languages at school is still on procedural languages (Pascal, Basic). Meanwhile, in recent years, the Python programming language has become increasingly popular, which is widely used throughout the world both for teaching and for creating a wide variety of programs: from text dialogues to serious web applications and 3D games [1]. Throughout the history of school computer science, the study of programming elements has been the basis of the course. This is due to the fact that the study of algorithmization and programming in high school is not only a goal, but also a means of shaping students' thinking, contributes to the development of their mathematical abilities, and also lays the foundation for continuing their education in senior specialized classes and at a

university [2-5] . Thus, the topic of the article looks particularly relevant, since it is aimed at developing a methodology for teaching schoolchildren to program in Python using the developed electronic educational tool (ESUN).

Purpose of the study. Electronic educational resources for educational purposes today are an integral part of the educational process and can be used to: achieve the goals of teaching computer science according to the Federal State Educational Standard; helping the teacher organize lessons that encourage students to acquire programming skills with interest. New tools for teaching programming languages are especially important. The purpose of our research was to develop the content of the elective course “Programming in the Python Language” and create the corresponding ESUN “Python Learning”. Material and research methods. An analysis of current school textbooks on computer science by various authors for grades 9-11 showed that they give preference to the imperative procedural and object-oriented programming paradigm (Pascal, Basic) [6-8]. An analysis of existing tools for developing training systems was also carried out, their advantages and disadvantages were considered.

Research results and discussion. In our opinion, the Python language is better than traditional school programming languages and is optimally suited both for initial programming training and for building specialized courses. The following advantages of the Python language can be noted [9]:

1. The language is more understandable than Pascal and Basic. Simple programs are written in several lines; instructions that are not directly related to the algorithm (for example, int main()) are not needed.
2. Simple and concise syntax. As a rule, a program in Python is written shorter than in C++, Pascal and Basic.
3. Free implementation.
4. Modernity of the language, the presence of high-level data structures (lists, sets, associative arrays, long arithmetic).
5. Availability of object-oriented programming (OOP) tools.
6. Availability of a library that allows you to easily develop graphical applications, web applications, etc.

The developed elective course “Programming in the Python language” is an elective subject for students in grades 10-11 of a senior specialized school. The course consists of both theoretical material, including information about the basic concepts of OOP, and practical tasks designed to consolidate the theoretical material received and instill programming skills. The course volume is 34 hours for the entry level (first module) and 24 hours for the advanced level (second module). Let us outline the goals of the course: to introduce students to the principles and basics of programming in the Python programming language; form a theoretical and practical basis for programming in Python; develop thinking based on computer logic and algorithmization. Each lesson includes both theoretical material and practical tasks, examples of their implementation. The theoretical part describes the syntax of the language, describes the main elements and structures, and talks about where they can be applied. In the practical part of the course, students are given tasks to complete independently in order to consolidate the theoretical material studied. The course is divided into two modules. The first module - the basics of the programming language, data types, basic algorithmic structures (following, branching, loops), structured programming are studied. The first module is based on working with simple data types; only at the end of the module is an introduction to structured types (arrays are studied). The second module is entirely devoted to the study of functions and OOP.

Theoretical part. First module

1. First acquaintance with Python (acquaintance with a brief history of the language, its capabilities and scope of application).
2. Data types in programming. Definition of a variable (data types, their use depending on the tasks).
3. Logical expressions (basic operations of Boolean algebra, logical operations in Python, features of their application).
4. Data entry from the keyboard.

Theoretical part. Second module.

1. Introduction to functions in Python (definition of functions in Python, syntax and features of their use).
2. OOP in Python (OOP terminology, basic concepts and definitions).
3. Inheritance in OOP in Python (the concept of “inheritance” within the Python programming language, its application and implementation).
4. Polymorphism and method overriding in OOP in Python (the concept of “polymorphism” and its implementation in Python, method overriding).

Practical part. First module.

1. “Hello world!” The first program in Python.

Students will be asked to write a program using various tools (console, IDLE).

2. Logical Expressions

Using practical examples, students will become familiar with Boolean algebra: logical addition, multiplication, and combinations of various types of operations.

3. Conditional statement if

Using examples, students will learn to use the conditional if statement, and will also reinforce the material on the topic “Logical Expressions.”

4. While loop

During the practical lesson, students will learn to write cyclic algorithms and become familiar with various options for their implementation.

5. Multiple branching

Students will become familiar with multiple branching and the scope of the if-elsif-else statement.

6. Data entry from the keyboard.

Acquisition of skills by students in input and output of data from the keyboard.
Practical part. Second module.
1. Parameters and arguments of functions. Local and global variables.
Students will be asked to implement functions in programs with and without parameters in order to understand the differences and identify their applications.
2. Creation of objects and classes.
Once students have an understanding of what OOP is, they can apply what they have learned in this lab by learning to describe classes and create objects.
3. Constructors of the __init__ class.
The lesson provides an expanded understanding of the description of a class and introduces the concept of a special method (constructor) of a class.
4. Compositional approach in object-oriented programming.
When solving geometric problems, students will gain skills in using composition.
5. Modules and their import.
Practical training for students in using modules in programs.
6. Python source code docstrings.
Students will master the mechanism of documenting code.

7. Operator overloading in OOP.

In practice, students will become familiar with the concept of operator overloading and work through several examples to reinforce practical skills in using this approach in organizing programs.

8. File system.

Practical reinforcement of material on working with files.
Throughout the course, students' work is assessed: source code for correct syntax, presence of comments, effective implementation of algorithms, clarity of variable names, functions, classes.
The composition and structure of the developed ESUN for the course “Programming in Python” includes the following components:
1. Training component (theoretical material to be studied and structured on educational topics; practical material containing the formulation of tasks and methodological recommendations for their implementation).

2. Control component (questions and test tasks, divided into two levels of difficulty: basic and advanced).

Let us dwell in more detail on the ESUN "PythonLearning" we developed - a program that can be divided into 3 stages: design of a software product, preparation of materials, aggregation of materials into a software product [6]. The application is intended for use as a didactic tool in computer science lessons in general education institutions, in independent work of students to study and repeat course materials, as well as for distance learning.

The development and debugging of the program was carried out using Microsoft Visual Studio, using a user interface in the form of a Windows Form. The program "PythonLearning.exe" is portable and is openly available to students (participants in the testing of the elective course program based on the Municipal Budgetary Educational Institution "Secondary School No. 12 named after V.V. Tarasov" in Penza). After starting the program, a splash screen will appear, after which the initial "Main" form of the program appears. This form will allow you to identify a new user, select the desired difficulty level and begin training, while if the user already exists, training will continue from the lesson where he left off last time.

The main work with lessons is carried out in the form (drawing). Moreover, the entire process of learning and testing knowledge is carried out in this form. This is achieved by embedding an API (web browser application programming interface) into a Windows Form and implementing each lesson as HTML pages.

REFERENCES

1. Abdumajitova, M. I. (2024). Zamonaviy ta'lim yondashuvlari asoslarida talabalarning kreativ fikrlashlarini takomillashtirish metodikasi. Ilm-fan, ta'lim-tarbiya, innovatsiya: zamonaviy vazifalari va istiqbollari, 1(1), 130-133.
2. Orasta, K. (2023). VOCABULARY TEACHING STRATEGIES USING IN ESP CLASSES. Gospodarka i Innowacje., 33, 134-137.
3. Abdumajitova, M. I. Q., & Yigitaliyeva, O. F. Q. (2024). Boshlang'ich sinf o'quvchilarining o'qish savodxonligini aniqlashda pirls tadqiqotining o'rni. Academic research in educational sciences, 5(NUU Conference 1), 480-485.
4. Hamraqulova, O. (2023). Sentence Structure in English Grammar Study in National Groups. Texas Journal of Philology, Culture and History, 17, 25-29.
5. Abdumajitova, M. I. (2024). Talabalarning kreativ fikrlashi: shakl va usullar, texnologiyalar. Sifatli ta'limni tashkil etishda shaxs kreativ potensialini oshirishning dolzarb masalalari, 1(2), 234-238.
6. Мустафаева, Л. В. (2023). Эсхатологические мотивы в романе «Generation P» Виктора Пелевина. Ta'lim jarayonida raqamli texnologiyalarni joriy etish, 1(1), 79-85.
7. Abdumajitova, M. I. (2024). Zamonaviy ta'lim yondashuvlari asoslarida talabalarning kreativ fikrlashlarini takomillashtirish metodikasi. Ilm-fan, ta'lim-tarbiya, innovatsiya: zamonaviy vazifalari va istiqbollari, 1(1), 130-133.

8. Мустафаева, Л. В. (2023). Лингвокультурологический потенциал рассказов а. П. Чехова в преподавании русского языка как иностранного. Проблемы формирования лингвокультурологической компетенции, 1(1), 75-78.
9. Abdumajitova, M. I. (2024). Boshlang'ich sinflarda dars jarayoni davomida neyroo'yin (neyrobika) o'tkazish maqsadi va vazifalari. Ilm-fan, ta'lim-tarbiya, innovatsiya: zamonaviy vazifalari va istiqbollari, 1(1), 127-129.
10. Abdumajitova, M. I. (2023). Bo'lajak boshlang'ich sinf o'qituvchilarida autopedagogik kompetentlikni rivojlantrish tamoyillari. O'zbekistonda fanlararo innovatsiyalar va ilmiy tadqiqotlar, 1(23), 192-198.
11. Abdumajidova, M. I. (2023). Primary Education Administration Standards. Diversity Research: Journal of Analysis and Trends, 1(3), 71-74.
12. Abdumajidova, M. I. Q. (2023). Boshlang'ich sinf o 'quvchilarini ta'limning keyingi turlariga tayyorlash omillari. Academic research in educational sciences, 4(6), 272-275.
13. Abdumajitova, M. I. (2023). Theoretical and legal basis of pre-investigation institute before the investigation. International Bulletin of Applied Science and Technology, 3(6), 502-507.
14. Abdumajitova, M. (2022). Use of educational technologies in mathematics lessons in primary school. Science and Innovation, 1(8), 722-725.
15. Abdumajitova, M. I. (2022). Increasing the efficiency of learning algebra elements on the basis of educational technologies in primary school mathematics. Экономика и социум, (2-2 (93)), 10-12.
16. Abdumajitova, M. (2022). Boshlang 'ich sinf matematika darslarida ta'lim texnologiyalaridan foydalanish. Science and innovation, 1(B8), 722-725.