DOI https://doi.org/10.30525/978-9934-26-504-4-10

UPDATING THE CONTENT AND METHODS OF TEACHING STUDENTS PYTHON PROGRAMMING

Petrenko S. I., Dehtiarova N. V.

INTRODUCTION

Informatics is one of the school subjects that help students form a modern scientific worldview. This educational discipline is a means of developing a student's intellectual potential. The work in informatics classes is designed to develop students' algorithmic, critical, and logical thinking styles, and it provides opportunities to create conditions for the development of creative abilities and aesthetic taste. Knowledge of informatics enables students to be better prepared for participation in subject Olympiads, not only in informatics, but also in intellectual competitions and discussions. Interest in informatics can inspire future applicants to choose a professional field and prepare them for higher education.

The school course in informatics involves studying a wide range of thematic areas, which expand and deepen throughout the learning process: information processes and systems, network technologies and the Internet, creation and publication of web resources, text data processing, computer presentations, spreadsheet data processing, databases, database management systems, computer graphics, 3D graphics, multimedia object processing, modeling, basics of algorithmization and programming, and fundamentals of information security.

With such a large number of study topics, one of the leading areas is the thematic line "Fundamentals of Algorithmization and Programming." An analysis of all current school informatics programs [1] [2] [3] [4] indicates that up to 40% of the study time in each grade is devoted to this topic, highlighting the importance of this thematic line in the informatics course. The content of teaching algorithmizing and programming in general education institutions, as well as the methodology of teaching these topics, has been extensively researched by scientists and methodologists over time.

[1] Інформатика. 5–9 класи. Програма для загальноосвітніх навчальних закладів.

[2] Програма курсу інформатика 8 – 9 класи загальноосвітніх навчальних закладів з поглибленим вивченням інформатики.

[3] Інформатика. Навчальна програма вибірково-обов'язкового предмету для учнів 10-11 класів загальноосвітніх навчальних закладів (рівень стандарту)

[4] Інформатика для 10-11 класів (профільне навчання)

Bazurin V.M. [5] in his research, studies the key factors influencing the choice of programming environment when teaching informatics in general secondary education institutions. He also analyzes the didactic potential of programming environments such as C, C++, C#, and Java. Certain historical aspects of the formation of the school informatics course from 2003 to 2012 were researched by Donchenko Ya.[6] [7]

Semenikhina O. and Rudenko Yu.[8] explore the challenges of teaching students to program and propose effective solutions to overcome these challenges. Methodological aspects of teaching the basics of algorithmization and/or programming, as well as the content of these topics, are studied by Noyuylnyk T., Kohut U., Zhydyk V. [9], Antonyuk B.[10], Barabolina T.M. [11] and others.

This issue has been researched since the inclusion of algorithmization and programming topics in the school informatics curriculum and requires updating with each content revision. Currently, Python is the most popular programming language for teaching in general secondary education institutions in Ukraine.

The **aim** of this work is to update the content and methods of teaching students programming in Python.

1. Analysis of the content of programming instruction based on textbooks and curricula

The course of informatics in middle school is studied as a mandatory separate subject. In grades 8–9, informatics can be studied in-depth, considering students' cognitive interests and the technical capabilities of the school. In grades 10–11, the informatics course can be studied at both the standard and advanced levels. Currently, a transformation is taking place in

[5] Базурін В. Середовище програмування як засіб навчання учнів основ програмування. *Інформаційні технології і засоби навчання.* 2017. Том 59. № 3. С.13-27. DOI:10.33407/itlt.v59i3.1601

[6] Донченко Я. Чинники розвитку шкільного курсу інформатики в загальноосвітніх школах України (2003-2012). URL: https://www.pulib.sk/web/kniznica/elpub/dokument/Bernatova9/subor/Donchenko.pdf.

[7] Донченко Я. До проблеми впровадження інформаційно-комунікаційних технологій у процес вивчення інформатики в загальноосвітніх школах України. *Педагогіка вищої та середньої школи.* 2014. Вип. 41. С.139-145

[8] Семеніхіна О., Руденко Ю. Проблеми навчання програмувати учнів старших класів та шляхи їх подолання. Інноваційні технології і засоби навчання. 2018. Том 66. № 4. 54-63. DOI:10.33407/itlt.v66i4.2149

[9] Кодильник И., Когут У., Жидик В. Методичні аспекти вивчення основ алгоритмізації та програмування мовою Python у шкільному курсі інформатики в старших класах. *Фізико-математична освіта.* Том 31. № 5. 2021. DOI: https://doi.org/10.31110/2413-1571-2021-031-5-006

[10] Антонюк Б. Основи алгоритмізації та програмування. Курс лекцій. Луцьк: Вежа-друк, 2022. 36 с.

[11] Барболіна Т.М. Шкільний курс інформатики та методика його викладання: навчальний посіб. Полтава, 2008. Ч.2. Часткова методика. 116 с.

the structure of general education. However, this area of research will remain relevant in the future, as it does not depend on the specific names of the components into which general education is divided but is focused on the content aspect of education.

Studying the subject of "Informatics" aims to develop students' ICT competence. The content line "Fundamentals of Algorithmization and Programming" directly influences the achievement of this strategic goal. Successful study of this thematic line is impossible without solving a complex set of interrelated tasks. The primary objectives in studying this topic are:

– Developing analytical skills in analyzing task conditions and the obtained results;

– Fostering algorithmic thinking in building a task model and finding solutions;

– Enhancing logical thinking for selecting problem-solving methods;

– Developing critical thinking when analyzing program code and finding the most optimal solution methods;

– Building theoretical knowledge on the fundamentals of algorithmization and programming;

– Developing the ability to predict the outcomes of program code execution;

– Cultivating creative thinking when choosing ways and methods of solving tasks and designing the appearance of the program code results;

– Developing concentration and attention skills while writing program code;

– Building the ability to identify, analyze, and correct errors;

– Enhancing spatial and schematic thinking;

– Strengthening interdisciplinary connections;

– Cultivating precision during program code writing;

– Developing skills for independent and team work;

– Fostering respect for colleagues during group project work and self-respect;

– Facilitating the choice of future professional activities.

The outlined tasks can only be accomplished through a comprehensive approach, as addressing each individual goal is impossible without simultaneously impacting others. In studying the content line "Fundamentals of Algorithmization and Programming," the curriculum provides for a sequential complication and deepening of the material, with the conditional division into four levels.

The first level (grades 2–4) involves a preparatory introduction to the conceptual framework of the topic and educational-simulation software environments.

The second level (grades 5–7) continues the introduction begun in elementary school, covering basic algorithmic structures and creating program codes to solve simple, intuitively understandable tasks. To account

for the age characteristics of students, educational-simulation software environments are used to support the study of the "Algorithms and Programs" section.

The third level (grades 8–9) involves the full formation of a conceptual framework sufficient for using the full-function environment of one of the programming languages.

The fourth level (grades 10–11 (12)) does not include the study of the "Fundamentals of Algorithmization and Programming" content line in the core module of the curriculum for the elective-compulsory subject at the standard level. However, it is possible to study this content line by choosing the "Creative Programming" module or by studying informatics under the curriculum of the specialized level.

The content line "Fundamentals of Algorithmization and Programming" is a system-forming component of the informatics course. This topic is studied throughout almost all years of schooling in a significant volume. Ensuring a comprehensive approach to studying this content line is one of the reasons it is considered one of the most challenging for most students in school[12].

Another factor that changes certain difficulties in learning programming is the content of the study. This challenge is related to the fact that school curricula do not limit teachers in programming the choice of the language of instruction. The choice of programming an educational language differs from several factors, among which the leading ones are: the teacher's knowledge of the language, the availability of textbooks and methodical materials, the traditions of studying programming at school, recommendations of school management and parents, etc. Considering all these necessary elements, we observe that currently various programming languages are being studied in schools Python, Pascal/Lasarus, Basic, C++, C#, Java, JavaScript and others [13,14].

School textbooks, in most cases, are oriented towards one of the programming languages.

The analysis of currently relevant textbooks revealed that the use of the Scratch environment is recommended in most textbooks for grades 5–6 to study algorithmization and programming (tab. 1).

[12] Семеніхіна О., Руденко Ю. Проблеми навчання програмувати учнів старших класів та шляхи їх подолання. Інноваційні технології і засоби навчання. 2018. Том 66. № 4. 54-63. DOI:10.33407/itlt.v66i4.2149

[13] Дегтярьова Н. В., Петренко С. І., Вернидуб Г. О., Тутова Н. О., Мигаль В. О. Реалізація диференційованого підходу при навчанні учнів програмуванню. Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми. Збірник наукових праць. Випуск сімдесят другий. Київ -Вінниця, 2024р.

14 S. Petrenko, N. Dehtiarova, K. Parfylo, O. Shovkoplias, O. Viunenko. The formation of widget development skills in the Python programming language. 47th Convention International Convention on Information, Communication and Electronic Technology, MIPRO 2024

Table 1

Analysis of 5–6 grade textbooks

Textbooks	The authors	Programming language, platform
Informatics: textbook for 5th grade. [15]	Trishchuk I.V.	Scratch
Informatics: textbook for 5th grade.[16]	Kozak L. Z., Vorozhbit A.V.	Scratch
Informatics: textbook for 5th grade. [17]	Ryvkind Y., Lysenko T., Chernikova L, Shakotko V.	Scratch
Informatics: textbook for 5th grade [18]	Hlynskyi Ya., Lysobei L., Chuchuk O., Diachun V.	Scratch
Informatics: textbook for 5th grade.[19]	Dzhon Endriu Bios.	Scratch
Informatics: textbook for 5th grade [20]	Korshunova O., Zavatskyi I.	Scratch
Informatics: textbook for 5th grade. [21]	Morze N.V. Barna O.V.	Scratch
Informatics: textbook for 5th grade [22]	Bondarenko O., Lastovetskyi V., Pylypchuk O., Shestopalov Ye.	Python
Informatics: textbook for 5th grade [23]	Korniienko M., Kramarovska S., Zaretska I.	Scratch
Informatics: textbook for 6th grade [24]	Джон Ендрю Біос.	Scratch
Informatics: textbook for 6th grade.[25].	Morze N.V. Barna O.V.	Scratch
Informatics: textbook for 6th grade [26]	Korshunova O., Zavatskyi I.	Scratch

[15] Тріщук І.В. Інформатика : підручник для 5 кл. закладів загальн. середн. освіти

[16] Козак Л. З. Інформатика : підруч. для 5 кл. закл. загал. серед. освіти

[17] Інформатика : підручник для 5 кл. закладів загальн. середн. освіти / Й. Ривкінд, Т. Лисенко, Л. Чернікова, В. Шакотько.

[18] Інформатика : підручник для 5 класу закладів загальної середньої освіти / Я. М. Глинський, Л. В. Лисобей, О. І. Чучук, В. В. Дячун.

[19] Джон Ендрю Біос. Інформатика : підруч. для 5 кл. закл. загал. серед. освіти

[20] Коршунова О.В. Інформатика : підруч. для 5 кл. закл. загал. серед. освіти / О.В. Коршунова, І.О. Завацький.

[21] Морзе Н.В., Барна О.В. Інформатика : підруч. для 5 кл. закл. загал. серед. освіти / Н.В.Морзе, О.В.Барна

[22] Інформатика.: підручник для 5 класу закладів загальної середньої освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопалов.

[23] Інформатика. Підручник для 5 класу закладів загальної середньої освіти. / Корнієнко М. М., Крамаровська С. М., Зарецька І. Т.

[24] Джон Ендрю Біос. Інформатика : підруч. для 6 кл. закл. загал. серед. освіти

[25] Морзе Н.В., Барна О.В. Інформатика : підруч. для 6 кл. закл. загал. серед. освіти / Н.В. Морзе, О.В Барна.

[26] Коршунова О.В. Інформатика : підруч. для 6 кл. закл. загал. серед. освіти / О.В. Коршунова, І.О. Завацький.

Informatics: textbook for 6th grade [27]	Ryvkind Y., Lysenko T., Chernikova L, Shakotko V.	Scratch
Informatics: textbook for 6th grade. [28]	Trishchuk I.V.	Scratch
Informatics: textbook for 6th grade [29]	Bondarenko O., Lastovetskyi V., Pylypchuk O., Shestopalov Ye.	Python

Accordingly, all tasks are designed to be performed in this environment.

The textbooks for 7th grade offer a transition from the Scratch environment to an introduction to the programming language. The following trend is observed (tab. 2):

− authors Kazantseva O. and Stetsenko I.[30] recommend continuing to use the Scratch environment;

− teams of authors under the leadership of Mozse N.[31] and Ryvkind Y.[32] offer the teacher and students a choice (continue learning using the example of the Scratch environment or, using an analogy with Scratch, move on to get acquainted with the Python programming language).

− the author team Bondarenko O., Lastovetskyi V., Pylypchuk O. and Shestopalov E. offer to continue acquaintance with the Python programming language, as in the entire line of their textbooks for grades 5–7 [33].

− the author team of O. Korshunova and I. Zavatskyi.[34] is considering the possibility of starting an acquaintance with the Python programming language.

[27] Інформатика : підручник для 6 кл. закладів загальн. середн. освіти / Й. Ривкінд, Т. Лисенко, Л. Чернікова, В. Шакотько

[28] Тріщук І. В. Інформатика : підручник для 6 кл. закладів загальн. середн. освіти

[29]Інформатика.: підручник для 6 класу закладів загальної середньої освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопалов.

[30] Казанцева О. П. Інформатика: підручник для 7 класу закладів загальної середньої освіти / Казанцева О. П., Стеценко І. В.

[31] Н.В. Морзе Інформатика : підруч. для 7 кл. закл. загал. серед. освіти / Морзе Н.В., Барна О.В.

[32] Інформатика : підручник для 7 кл. закладів загальн. середн. освіти / Й. Ривкінд [та ін].

[33] Інформатика.: підручник для 7 класу закладів загальної середньої освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопалов.

[34] Коршунова О.В. Інформатика : підруч. для 7 кл. закл. загал. серед. освіти / О.В. Коршунова, І.О. Завацький.

Table 2

Analysis of textbooks for the 7th grade

Textbooks	The authors	Programming language, platform
Informatics: textbook for 7th grade	Kazantseva O. Stetsenko I.	Scratch
Informatics: textbook for 7th grade	Morze N.V. Barna O.V.	Scratch or Python
Informatics: textbook for 7th grade	Ryvkind Y.and oth.	Scratch or Python
Informatics: textbook for 7th grade	Bondarenko O., Lastovetskyi V., Pylypchuk O., Shestopalov Ye.	Python
Informatics: textbook for 7th grade	Korshunova O., Zavatskyi I.	Python

Current textbooks for grades 8-9 follow the study of various programming languages. For the 8th grade, the authors' collectives of Kazantseva O.[35], Morse N.[36], Korshunova O.[37], Bondarenko O.[38], and Rudenko offer the Python language with a textbook for classes with in-depth study of computer science[39]. And in the textbook of the author's team under the leadership of Ryvkind Y.[40], the possibility of choosing to study Lazarus/Free Pascal or Python is provided.

Textbooks for the 9th grade follow a different trend. Author collectives Rudenko V.[41],., Ryvkind Y.[42], Bondarenko O.[43] offer to study programming in the Lazarus environment. And when studying programming according to the textbook of the author's team under the leadership of Morse N.[44], it is

[35] Казанцева О. П. Інформатика: підручник для 8 класу закладів загальної середньої освіти / Казанцева О. П., Стеценко І. В.

[36] Н.В. Морзе Інформатика : підруч. для 8 кл. закл. загал. серед. освіти / Морзе Н.В., Барна О.В.

[37] Коршунова О.В. Інформатика : підруч. для 8 кл. закл. загал. серед. освіти / О.В. Коршунова, І.О. Завацький. З.Р. Стасюк

[38] Інформатика.: підручник для 8 класу закладів загальної середньої освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопалов.

[39] Руденко В.Д. Інформатика. Підручник для 8 класу з поглибленим вивченням інформатики закладів загальної середньої освіти. / В.Д. Руденко, Н.В. Речич, В.О. Потієнко.

[40] Інформатика : підручник для 8 кл. закладів загальн. середн. освіти / Й. Ривкінд [та ін].

[41] Руденко В. Д. Інформатика для загальноосвітніх навчальних закладів з поглибленим вивченням інформатики : підруч. для 9 кл. загальноосвіт. навч. закл. / В. Д. Руденко, Н.В. Речич, В.О. Потієнко.

[42] Інформатика : підручник для 9 кл. закладів загальн. середн. освіти / Й. Ривкінд [та ін].

[43] Інформатика.: підручник для 9 класу закладів загальної середньої освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопалов.

[44] Н.В. Морзе Інформатика : підруч. для 9 кл. закл. загал. серед. освіти / Морзе Н.В., Барна О.В., В.П. Вембер

possible to choose the programming language Free Pascal in the Lazarus environment or Python in the PyCharm environment.

Table 3

Analysis of textbooks for grades 8–9

Підручник	Автори	Мова програмування, Середовище
Informatics: textbook for 8th grade	Kazantseva O., Stetsenko I.	Python
Informatics: textbook for 8th grade	Morze N., Barna O.	Python
Informatics: textbook for 8th grade	Korshunova O., Zavatsky I., Stasyuk Z.	Python
Informatics: textbook for 8th grade	Bondarenko O., Lastovetskyi V., Pylypchuk O.	Python
Informatics. Textbook for 8th grade with in-depth study of computer science	Rudenko V., Rechych N., Potienko V.	Python
Informatics: textbook for 8th grade	Ryvkind Y. [and others].	Lazarus / Free Pascal або Python
Informatics. Textbook for 9th grade with in-depth study of computer science	Rudenko V., Rechych N., Potienko V.	Lazarus / Free Pascal
Informatics: textbook for 9th grade	Ryvkind Y. [and others].	Lazarus / ObjectPascal
Informatics: textbook for 9th grade	Bondarenko O., Lastovetskyi V., Pylypchuk O.	Lazarus / ObjectPascal
Informatics: textbook for 9th grade	Morze N., Barna O.	Lazarus / Free Pascal або Python / PyCharm

The textbooks recommended by the Ministry of Science and Education of Ukraine for grades 10-11, which provide for the study of an elective and mandatory computer science course at the standard level, do not have materials for studying the content line of programming, because it is not provided for in the basic module. There is no textbook or manual recommended by the Ministry of Education and Culture of Ukraine for studying the optional module "Creative Programming".

For the study of computer science in grades 10[45]–11[46] at the professional level, the textbooks of the author's team under the leadership of V. D. Rudenko are recommended. Working on these tutorials involves learning the Python language.

[45] Руденко В. Д. Інформатика (профільний рівень): підруч. для 10 кл. закл. Загальн. серед. освіт. /В. Д. Руденко, Н.В. Речич, В.О. Потієнко.

[46] Руденко В. Д. Інформатика (профільний рівень): підруч. для 11 кл. закл. Загальн. серед. освіт. /В. Д. Руденко, Н.В. Речич, В.О. Потієнко.

The study of the content line "Fundamentals of algorithmization and programming" in the school computer science course can be conventionally divided into several successive interrelated stages.

The first of them introduces the conceptual base of the topic. Basic concepts include: command, command executor, system of executor commands, object, object properties, algorithm, program. Separation of the functions of any sentence and a persuasive one allows students to understand that the command has its own special, readable lexical structure, in which there is an unambiguous command to perform an action. Such an action can be performed by the performer. Understanding the content of basic concepts lays the foundation for successful study of this topic and ensures a consistent transition from spoken language to the language of commands and algorithms and the translation of algorithms into one of the programming languages.

The next important step is to get acquainted with the ways of presenting algorithms. Currently relevant textbooks, in most cases, recommend a consistent transition from a verbal way of presenting algorithms to a graphic one (a system of points connected by arrows), and from a graphic one to block diagrams.

At the first stages of consolidating the skills of presenting algorithms, you can start with everyday examples (brushing your teeth, the order of physical exercise, planting a tree, etc.). When moving to a more formal stage of creating algorithms, it is advisable to use examples of logical problems.

For example:

As having a capacity of 2 l. and 5 liters draw 1 liter from the water tap. water Make an algorithm for solving this problem. How many ways can you solve this problem?

It is also advisable to create algorithms for creating a winning strategy in a logical game. Such a classic example can be the game of Bashe, or the three-on-three Tic Tac Toe game.

Using the example of block diagrams, it is more expedient to demonstrate the basic algorithmic structures. With the help of block diagrams, the difference between a linear structure, a cycle and a branch is clearly visible.

A very important step in studying the content line "Fundamentals of Algorithmization and Programming" is the transition from algorithmization to the implementation of algorithms in the programming environment. Since students of the 5th grade have a visual way of thinking, it is advisable to make this transition with the help of one of the visual programming environments. The most common visual programming environments are: Scratch, KTurtle, Blockly. Each of these environments has its own

advantages and disadvantages. But most of the authors of actual textbooks for grades 5-6 stop at the Scratch environment.

The first algorithms that are implemented usually have a linear structure. Examples of such tasks can be the program (Fig. 1.), which offers a tour of the rooms in the apartment.

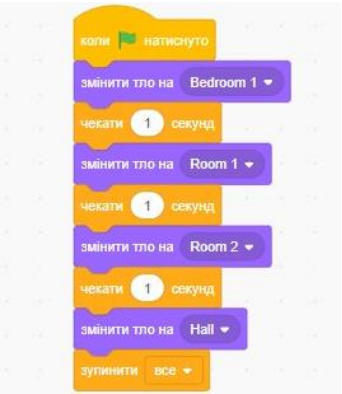

Fig. 1. The code of the program that changes the color of the room in the apartment

Such a code has a significant didactic potential, because on its basis, students can be offered several tasks:

- change the order of visiting rooms;
- change the time of stay in the rooms;
- set different stay times in different rooms;
- add music for a tour of the rooms;
- add your music design for each room;
- by analogy, make a project for changing the seasons and others.

Additional tasks can be presented as mini-projects for work in pairs or small groups.

It is advisable to familiarize yourself with cycles using examples of creating animation effects. For example, demonstrate creating an animation for a swimming shark (Fig. 2).

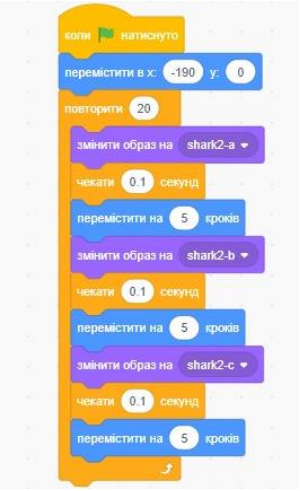

Fig. 2. Animation for creating a shark figure

In this project, ask students to change some parameters;

– increase and decrease the number of repetitions;
– change the waiting time;
– number of steps per movement;

Draw conclusions after each change of parameters and after changing a group of parameters.

Similar tasks can be developed to create animations with other sprites (a person exercising or dancing and others).

A cycle with a parameter is the easiest for students of grades 5–6 to understand, but it is not possible to solve all problems with such structures. Theoretically, there are also cycles with a precondition and a postcondition. Therefore, at the stages of familiarization with repetition structures in the Scratch environment, there is a structure "repeat until...". Using it, you can create cycles with a precondition and a postcondition.

To demonstrate the operation of such a structure, we suggest considering the example of the "Red Cat" hunting for a mouse (Fig. 3).

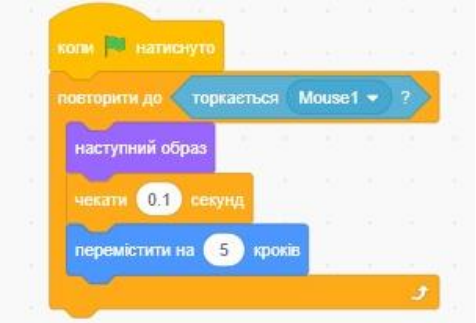

Fig. 3. Hunting of the "Red Cat" for a mouse

Acquaintance of students with cyclic structures involves simultaneous acquaintance with groups of commands "Operators", "Sensors" and "Variables". These groups of commands will allow students to expand the capabilities of all algorithmic structures and move from elementary projects with one or two sprites to more complex ones that involve the interaction of several active objects. And also at this stage, in math classes, you can start programming for solving math problems.

As a creative task, after studying linear structures and cycles, students can be offered to create a comic book based on one of the Ukrainian folk tales "Kurochka Ryaba", "Rukavychka", "Kolobok" and others. Creating a comic can be organized as a project task for working in groups or pairs.

The greatest difficulties arise when studying branching structures. These difficulties are due to the fact that 5th grade students have not yet become familiar with logical operations and it is difficult for them to understand the meaning and result of such operations. Practical experience shows that it is more appropriate to start acquaintance with an incomplete branching structure. The block diagram of the branching structure is somewhat similar to a loop and so the analogy works.

It is advisable to demonstrate the first tasks using sprite management examples (Fig. 4).

After the implementation of such a project, it is advisable to invite students to experiment with the choice of specific keys for movement (arrows left, right, up, down, and others). After such experiments, the students have a need to combine object management in one structure. This makes it possible to move on to familiarity with the full if-then-else branching structure and the if-then-else choice structure.

The practical use of different types of branching structures involves getting to know the simplest logical operations "and", "or", "not". Understanding the content and result of performing these operations makes a

significant contribution to the transition from visual thinking to logical, conceptual.

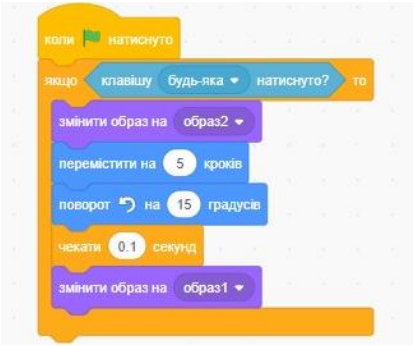

Fig. 4. Movement of the sprite with a keyboard key

In order to generalize the study of algorithmic structures, interested students can be offered, as an additional task, to develop small computer games and demonstrate their work in class.

Learning programming

When transitioning from the stage of getting to know the process of algorithmization and implementation of the simplest algorithms in simulation environments of programming languages to the process of learning a programming language, the question of language choice arises. The curriculum leaves this choice to the teacher. The analysis of currently relevant textbooks shows that the opinion of textbook authors on this issue is divided. It is suggested to study one of two languages: Pascal or Python.

Analysis of the latest research on the use of programming languages by Ukrainian specialists gives the following results:

– in 2017, the five most used programming languages included: Java – 23%, JavaScript – 16%, C# – 15%, PHP – 13%, Python – 7% (none of the Pascal variants are in the ranking, which includes 25 languages)[47];

– in 2024 the research made it possible to include the following five most used programming languages: JavaScript – 15.4%, TypeScript – 15.1%, Python – 14.7%, Java and C# each 12.7%[48];

– among programmers who know more than one programming language, the most popular are JavaScript and Python, which are found in the majority of cases [49]

[47] Рейтинг языков программирования № 8: Java по-прежнему доминирует, но доля рынка начала сжиматься

[48] Рейтинг мов програмування 2024.

[49] Скільки мов використовують українські розробники та як їх комбінують

From this analysis, we can conclude that in order for a student to be able to compete in the IT services market in the future, he must learn a programming language from school that will be useful in the future. The most advanced language on this list is TypeScript. JavaScript and Python should be highlighted among the most popular. At the same time, the popularity of Python is currently growing quite rapidly, and the popularity of JavaScript is slightly but decreasing.

If we take into account the presented analytics and the content of currently relevant textbooks, then the choice is clear – Python is the best programming language to start learning. As an educational language, Python has several advantages over other languages: the syntax of the language is simple and the program code is easy to read, the language is cross-platform, easy to install, and has standard libraries that are installed together with Python and contain ready-made tools for working with the operating system system, web pages, various data formats, for building the graphical interface of programs, etc.

Learning the Python language in most cases traditionally begins with displaying the proposed expression on the screen. For example:

>>>print (Hello world)

Hello world

During the next steps of working in the Python programming language environment, it is advisable to offer students to display the results of the simplest mathematical calculations (addition, subtraction, multiplication) on the screen. Performing such tasks requires familiarity with some types of data and functions that allow them to be entered. Whole and real numbers and string values and ways of declaring them are sufficient for work in the first lessons. Simultaneously with the consideration of the types of values, it is necessary to consider the concept of a variable and understand the possible names of variables.

The Python language is interesting as well as educational because a significant number of standard libraries are available with its standard installation. For the educational process, at the stage of the first acquaintance, it is advisable to use the tkinter library of graphic elements. Using this library for learning has several important points:

First, students create graphic objects and visually see the results of their work.

Secondly, the skills of writing program code using comments are acquired.

Thirdly, the "vocabulary" of official words and functions and properties increases.

Fourthly, the meaning of the concept of "object" is deepened and its properties and events that occur when using this object are distinguished.

Fifth, the skills of importing libraries or library elements are formed.

The first code when using the tkinter library will be the window creation code:

*from tkinter import * # importing graphics library*
window = Tk() # creating the main window
window.title ('Графічна програма') # creating the name main window
window.geometry('500x500'+20+20) # the size of the main window and the location point

After the creation and implementation of this code, students should be invited to conduct experiments with this code to remove each line of code or some property and draw conclusions.

The results of such an experiment will allow the teacher to introduce the concept of "method" along with the concept of function. Separate them and show the difference in use with examples.

Example: Tk() – function, title (), *geometry()* – methods.

When creating labels, frames, single-line and multi-line text fields, the main attention should be paid to the properties of these elements and their values (parameters),

On the elementary example of creating a label with the text "Hello world!"[50] it is expedient to offer students to conduct experiments on adding properties and their values to the program code (tab. 4). Such work will allow students to more thoroughly understand the syntax of the Label() function and learn to manage its parameters.

*from tkinter import ** # import graphics library
window=Tk() # creating the main window
window.title ('Мітка') # creating the name of the main window
widget = Label(window, text='Hello world!') # creating a label with the name widget, which will display the text on the screen
widget.pack() # placing a label on the screen using the packer

Label widget options[51]

Property	Value
width	Label width
height	Label height
text	Text on the label
bg	Label background
fg	Text color
bd	Label border width
relief	Label relief (FLAT, GROOVE, RIDGE, SUNKEN, RAISE)
font	Type of font on the label
textvariable	The name of the variable in which the text on the label will be stored

[50] Петренко С.І., Дегтярьова Н.В. Створення віджетів мовою програмування Python. Методичні рекомендації с. 12.
[51] Python – просто! Віджети. Label

Lessons on creating frames, one-line and multi-line text fields, a scale and a list can be conducted using a similar algorithm, which will allow you to develop stable skills in using properties and their values.

Learning the process of installing a button differs from previous widgets in that an event occurs when the button is clicked. Therefore, a button is an object that has its own parameters and performs a certain action.

The action itself is created by its own function. The syntax of own functions and examples of their use are very convenient to consider when studying the creation of buttons, because these two processes are organically intertwined and complement previously acquired knowledge and skills.

To set the action that will occur after clicking the button, you can use the command method. This method allows you to add an action that is declared using a user-created function.

Example: *You need to set an action on the button that changes the text on the button and its color.*[52]

```
from tkinter import * # import graphics library
window=Tk() # creating the main window
window.title ('кнопка') # creating the name of the main window
butt = Button(text=" Press the button ", # setting the text on the button
width=20, # width
height=2, # height
bd=5, # button border width
fg='gray') # the color of a button that has not yet been pressed

def press(): # creating an action function for the button
butt['text'] = "Молодець" # new text on the button
butt['bg'] = 'red' # new color on the button

butt.config(command=press) # sets the action on the button
butt.pack() # places a button in a window
```

To create other widgets, such as a radio button and a switch, you need to consider the syntax features of branching and repeating structures.

The structure of branching and loop structures involves the execution of logical operations that students have already become familiar with in grades 5–7, but Python has its own peculiarities, so it is necessary to draw analogies with previous knowledge and highlight special points.

[52] Мова програмування Python: Основи програмування. Навчальний посібник для студентів спеціальності 014 Середня освіта (Інформатика) / С. Петренко, Н. Дегтярьова. С. 38.

Python uses the following operators to compare objects (variables of different types).:

– > – the mathematical sign is greater;
– < – the mathematical sign is less;
– >= – mathematical sign is greater than or equal to (not less than);
– <= – mathematical sign is less than or equal to (not more than);
– == – the mathematical sign is equal to;
– != – the mathematical sign is not equal to (not equal to).

It is also necessary to draw the attention of students that quite often there is a need to build complex logical expressions that contain several simple logical statements, between which logical operators must be used: and, or, not. Students are already familiar with these operators from previous classes.

It is expedient to consider branching structures by analogy with structures from Scratch (if – then – otherwise)

Syntax of the if-else statement**:**

if boolean_expression:
command_block _1
else:
command_block _2

The execution of the if-else statement involves the calculation of the value of a logical expression. If the value of the logical expression is true, block_command_1 is executed, otherwise block_command_2 is executed.

Using the example of the task of determining the sign of a non-zero number, the program code is implemented as follows:

x = int(input("Enter a non-zero number:"))
if x > 0:
print("The number is positive.")
else:
print("The number is negative.")

A branch statement may be incomplete if it lacks an else branch with a corresponding statement block.

In the process of textual design of the code, the entry of the service word if with a logical expression and the service word else is placed in the code as the main instruction, and nested command blocks are shifted to the right.

If the condition of entering a non-zero number is removed from the condition of the problem, it becomes necessary to determine whether the number is positive, whether the number is negative, or whether the number is zero. There is a need to supplement the program with the additional need to check the number for equality of zero and display the corresponding message.

This can be done by using the incomplete form of the branching operator three times:

```
x = int(input("Enter a number:"))
if x > 0:
print("The number is positive.") 55
if x < 0:
print("The number is negative.")
if x == 0:
print("The number is zero.")
```

In some tasks, the implementation of branching in this way is inconvenient or poorly readable in the program code. In this case, Python provides a simplified notation for performing a large number of checks. For their implementation, the if-elif-else operator is used, which has the following form:

```
if boolean_expression _1:
command_block _1
elif boolean_expression _2:
command_block _2
elif boolean_expression _3:
command_block _3
...
[else:
command_block _N]
```

To implement our task, the code will look like this:

```
x = int(input("Enter a number:"))
if x > 0:
print("The number is positive.")
elif x < 0:
print("The number is negative.")
else:
print("The number is zero.")
```

This structure is sometimes called a selection operator.

The study of cycles also takes place by analogy with Scratch structures. First, it is advisable to study the loop with a prerequisite, which is also called the While loop (until), since it is with this keyword that it begins. The While loop is used in cases where it is not possible to determine the value of the number of times the loop is used.

The syntax of the while loop statement:

```
while logical_expression:
Command block of the loop body
```

In the process of textual design of the code, the record of the while function word with a logical expression is placed in the code as the main

instruction, and the nested command blocks of the loop body are shifted to the right.

Приклад. Calculate the factorial of a number n.

```
print('Enter a natural number n=')
n=int(input()) # entering a number from the keyboard
f=1 # the initial value of the factorial
i=2 # the initial value of the counter
while i<=n: # cycle start (condition check)
f=f*i # multiplication
i=i+1 # the next counter value
print('Factorial=',f) # result
```

Learning a loop with a parameter, which is analogous to a loop with a predetermined number of steps in Scratch, but has the ability to iterate not only numbers, but any elements from a certain set. It is called for. For a better understanding of its work, it is advisable to introduce students to data of the range type (range).

Syntax of the loop operator for:

```
for loop_variable in variable sequence:
command_block
```

In the first step, the loop variable is assigned the value of the first element of the sequence, then a block of commands from the loop body is executed. Next, the loop variable is given the next value from the sequence – a block of commands from the loop body is executed, etc. until the loop variable has been assigned all values of the sequence.

Example. For example, we suggest solving the problem of calculating the factorial of a number given from the keyboard and comparing program codes using loops with a parameter and a prerequisite.

```
print('Enter a natural number n=')
n=int(input())
f=1
for i in range (2, n+1): # cycle start (number of iterations)
f=f*i
print('Factorial=',f)
```

2. Studying widgets as an example of a differentiated approach in teaching programming

Studying the processes of creating widgets allows students to familiarize themselves with the use of basic algorithmic structures and functions and move on to a more thorough study of the Python programming language.

Also, the study of different structures provides an opportunity to apply a differentiated approach.

In December 2022, a survey of teachers in the city of Sumy and the Sumy region was conducted. The regional context of educational programming practices was studied. The regional aspect demonstrates the level of expected learning outcomes and the perspective of the region for enterprises or companies in a certain industry. A total of 26 educational institutions were covered. Every computer science teacher has the right to make a reasonable choice of platforms or software tools for teaching students. The survey showed that the majority of teachers take into account the modern requirements of the labor market and it was found that the Python programming language is gradually being implemented by more and more educational institutions.

The authors created a survey and one of the questions was "Which programming language should be studied at school?". More than 82% of respondents indicated the Python programming language.

We were also interested in the answers to the question to teachers "What are the problems in learning a programming language, in your opinion?". The answers were as follows:

algorithmic thinking is not developed in students (65.4%);

the student does not know how to adapt the ready-made algorithm for the task with small changes in the condition (46.2%);

the student does not understand the structure of the program (38.5%);

the student does not know how to test programs (15.4%)

and others.

And to the question "What, in your opinion, is most important for students' interest in programming?" Rank from most important to least important. You can add your own option in the free line" the following responses were received:

– own desire (92.3%);

– material support (access to the network, availability of a powerful PC, etc.);

– awards at competitions, receiving scholarships from donors (42.3%);

– and others.

Quite often, the answer from the respondents was similar to the following: "If the student wants, he will find an opportunity and he does not need external influencing factors."

The question "Do you use a differentiated approach when teaching programming?" was important for the content of the research. The answers were distributed as follows:

no, in any year – 15.4%,

rarely – 30.8%;

depends on the topic – 30.8%;

often – 15.4%;

yes, in each lesson – 7.6%.

After the questionnaire, the authors suggested that the final lessons on the topic be conducted in the form of practical work with differentiated tasks. In the schools of the city of Sumy and the Sumy region, it was suggested to hold classes according to the methodical recommendations developed by the authors of this study. It emphasized the use of a differentiated approach in the final lessons. Students were divided into 2 groups: experimental (group A) and control (group B). In group A, recommendations and examples were implemented, in group B, training continued without changes. A total of 52 students were involved.

An example of the proposed works will be demonstrated in the task: "Write the program code for calculating the recommended mass of a man and a woman according to Paul Brock's formula."

Option 1. The program is executed using a linear structure.

Option 2. The program contains a condition.

Option 3. The student creates a widget and uses Python programming language libraries.

The criteria by which the works will be evaluated were also announced. The maximum number of points is 12. An example of distribution is given in Table 5.

Table 5

Distribution of points when implementing a differentiated approach in teaching programming

Difficulty level	Number of points	Distribution of points
Option 1. Linear structure program	until 7	The program works without errors – 4 points The title "Calculate recommended weight" is displayed – 1 point An explanation of the request "Enter your height in cm=" is provided – 1 point Output of the explanation of the result "Recommended weight" – 1 point
Option 2. Program with branching	until 9	The program works without errors – 5 points The title "Calculate recommended weight" is displayed – 1 point An explanation of the request "Enter your height in cm=" is provided – 1 point An explanation of the request "Enter gender (h-male, f-female)" is provided – 1 point Output of the explanation of the result "Recommended weight" – 1 point
Option 3. Using widgets	until 12	The program works without errors – 7 points An explanation of the requests and output of the result is provided – 1 point Using the radio button – 1 point The calculated start button is provided – 1 point Convenient placement of elements in the window – 2 points

Announcing clear criteria is important because:

– the student focuses on specific requirements, registration rules, etc.;

– the student learns and practices the habit of predicting the output of messages for users;

– before completing the task, the student sees the number of necessary steps, estimates the time required for completion, evaluates his own abilities;

– the student also sees that a higher score is achievable and for this it is necessary to fulfill specific requirements that are clearly explained and realistic to fulfill;

– the student can self-assess according to the following criteria.

There was also a reflection among students about their attitude to differentiation and programming. The results confirmed the expectations – the attitude towards programming has changed. Interest in programming in class has increased. Students made their own choices. On different days, they chose a higher or lower level. And although some of them increased their level of knowledge and success, not every one of these students said that they would choose programming as their future activity. Students decided for themselves whether they are ready to develop programming skills in the future. In this way, students understood their preferences.

CONCLUSION

Theoretical analysis of publications on the topic of research, results of questionnaires and experimental research Make it possible to draw the following conclusions.

1. The Python programming language is popular, chosen by the majority of teachers and interested in students. The research conducted by the authors of the publication and a comparison of similar studies from 2017 and 2023 demonstrate the growing demand for specialists who know and are ready to work with the Python language.

2. In Ukraine, a teacher can choose a platform and a programming language for students of a certain class. Avhonomy in this matter leads to different options for learning platforms. However, it is the teacher's responsibility to choose a modern programming language and/or platform.3. The task of teaching programming and algorithmization can only be solved comprehensively, since the solution of each individual goal is impossible without simultaneous influence on others. When studying the "Fundamentals of Algorithmization and Programming" content line, the curriculum provides for successive complication and deepening of the material with a conditional division into four levels.

The first level (grades 2–4) involves a preparatory introduction to the conceptual apparatus of the topic and educational simulation software environments.

The second level (grades 5–7) continues the introduction begun in elementary school, covering basic algorithmic structures and creating programming codes to solve simple, intuitive problems. To take into account the age characteristics of students, educational and simulation software environments are used to ensure the study of the "Algorithms and programs" section.

The third level (grades 8–9) involves the complete formation of a conceptual basis sufficient to use a fully functional environment of one of the programming languages.

The fourth level (grades 10–11 (12)) does not provide for the study of the content line "Fundamentals of algorithmization and programming" of the main module of the curriculum from the elective subject of the standard level. However, you can study this content line by choosing the "Creative programming" module or by studying computer science according to the profile level curriculum.

The content line "Fundamentals of algorithmization and programming" is a system-forming component of the computer science course. This topic is studied practically during all years of study in a significant amount. Providing a comprehensive approach to the study of this content line is one of the reasons why it is considered one of the most difficult for most students in school

4. Teachers do not apply the differentiated approach enough when studying programming. The ability to choose tasks for students turned out to be important. Pupils who are interested in programming chose the most difficult options for solving problems. Students who are not interested in programming positively evaluated the possibility of choosing simpler ways to solve problems. It is important to give the student freedom of choice. The teacher should not force interest in a certain topic. This form of work is saturated with an educational element as well. Students understand themselves better. Students allow themselves to get an average grade. They are aware of topics (disciplines) that they like and those that they do not want to study. And this means that they form a professional orientation.

BIBLIOGRAPHY

1. Бондаренко О., Ластовецький В., Пилипчук О., Шестопалов Є. Інформатика.: підручник для 5 класу закладів загальної середньої освіти. Харків, «Ранок», 2022. 211 с.

2. Бондаренко О., Ластовецький В., Пилипчук О., Шестопалов Є. Інформатика.: підручник для 6 класу закладів загальної середньої освіти. Харків, «Ранок», 2022. 271 с.

3. Бондаренко О.О., Ластовецький В.В., Пилипчук О.П., Шестопалов Є.А. Інформатика.: підручник для 8 класу закладів загальної середньої освіти Харків, «Ранок», 2021. 240 с.

4. Глинський Я, Лисобей Л., Чучук О., Дячун В. Інформатика : підручник для 5 класу закладів загальної середньої освіти. Тернопіль : Видавництво «Астон», 2022. 208 с.

5. Дегтярьова Н. В., Петренко С. І., Вернидуб Г. О., Тутова Н. О., Мигаль В. О. Реалізація диференційованого підходу при навчанні учнів програмуванню. Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми. Збірник наукових праць. Випуск сімдесят другий. Київ - Вінниця, 2024. С. 53–60.

6. Джон Ендрю Біос. Інформатика : підруч. для 5 кл. закл. загал. серед. Освіти. К.: Видавництво «Лінгвіст», 2022. 176 с.

7. Джон Ендрю Біос. Інформатика : підруч. для 6 кл. закл. загал. серед. освіти. К.: Видавництво «Лінгвіст», 2023. 176 с.

8. Інформатика. Навчальна програма вибірково-обов'язкового предмету для учнів 10-11 класів загальноосвітніх навчальних закладів (рівень стандарту). URL: https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-dlya-10-11-klasiv

9. Інформатика. 5–9 класи. Програма для загальноосвітніх навчальних закладів. URL: https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-5-9-klas

10. Інформатика для 10-11 класів (профільне навчання). URL: https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-dlya-10-11-klasiv

11. Казанцева О. П. Інформатика: підручник для 7 класу закладів загальної середньої освіти. Тернопіль: Навчальна книга. Богдан, 2020. 176 с.

12. Казанцева О. П. Інформатика: підручник для 8 класу закладів загальної середньої освіти. Тернопіль: Навчальна книга, 2021. 256 с.

13. Козак Л. З. Інформатика : підруч. для 5 кл. закл. загал. серед. освіти. Київ : Літера ЛТД, 2022. 176 с

14. Корнієнко М. М., Крамаровська С. М., Зарецька І. Т. Інформатика. Підручник для 5 класу закладів загальної середньої освіти. Харків, «Ранок», 2022. 176 с.

15. Коршунова О.В., Завацький І.О. Інформатика : підруч. для 5 кл. закл. загал. серед. Освіти. К.: Видавничій дім «Освіта». 2023. 176 с.

16. Коршунова О.В., Завацький І.О. Інформатика : підруч. для 6 кл. закл. загал. серед. освіти. К.: Видавничій дім «Освіта». 2023. 208 с.

17. Коршунова О.В Інформатика : підруч. для 7 кл. закл. загал. серед. освіти. К.: Видавничій дім «Освіта». 2020. 144 с.

18. Коршунова О.В. Інформатика : підруч. для 8 кл. закл. загал. серед. освіти. К.: Видавничій дім «Освіта». 2021. 256 с.
19. Морзе Н.В., Барна О.В. Інформатика : підруч. для 5 кл. закл. загал. серед. освіти. Київ: УОВЦ «Оріон», 2023. 256 с.
20. Морзе Н.В., Барна О.В. Інформатика : підруч. для 6 кл. закл. загал. серед. освіти. Київ: УОВЦ «Оріон», 2023. 256 с.
21. Морзе Н.В. Інформатика : підруч. для 7 кл. закл. загал. серед. освіти. Київ: УОВЦ «Оріон», 2020. 176 с.
22. Морзе Н.В. Інформатика : підруч. для 8 кл. закл. загал. серед. освіти. Київ: УОВЦ «Оріон», 2021. 224 с.
23. Морзе Н.В. Інформатика : підруч. для 9 кл. закл. загал. серед. освіти. Київ: УОВЦ «Оріон», 2017. 208 с.
24. Петренко С.І., Дегтярьова. Н.В. Створення віджетів мовою програмування Python. Методичні рекомендації. Суми: СумДПУ ім. А.С. Макаренка. 2023. 32 с.
25. Програма курсу інформатика 8–9 класи загальноосвітніх навчальних закладів з поглибленим вивченням інформатики. URL: https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-5-9-klas/informatika.pdf
26. Python – просто! Віджети. Label. URL:https://sites.google.com/comp-sc.if.ua/python-easy/tkinter/%D0%B2%D1%96%D0%B4%D0%B6%D0%B5%D1%82%D0%B8/label?authuser=0
27. Ривкінд Й., Лисенко Т., Чернікова Л., Шакотько В. Інформатика : підручник для 5 кл. закладів загальн. середн. освіти. Київ: «Генеза». 2022. 313 с.
28. Ривкінд Й., Лисенко Т., Чернікова Л., Шакотько В. Інформатика : підручник для 6 кл. закладів загальн. середн. Освіти. Київ: Генеза. 2023. 279 с.
29. Ривкінд Й. та ін. Інформатика : підручник для 7 кл. закладів загальн. середн. освіти. Київ: «Генеза». 2020. 176 с.
30. Рейтинг мов програмування 2017. URL: https://dou.ua/lenta/articles/language– prating-jan-2017/
31. Рейтинг мов програмування 2023. URL: https://dou.ua/lenta/articles/language-rating-2024/
32. Руденко В.Д. Інформатика. Підручник для 8 класу з поглибленим вивченням інформатики закладів загальної середньої освіти. Харків, видавництво «Ранок», 2021. 256 с.
33. Руденко В. Д. Інформатика для загальноосвітніх навчальних закладів з поглибленим вивченням інформатики : підруч. для 9 кл. загальноосвіт. навч. закл. Харків : Вид-во «Ранок», 2017. 240 с.

34. Руденко В. Д. Інформатика (профільний рівень): підруч. для 10 кл. закл. загальн. серед. освіт. Харків : Вид-во «Ранок», 2019. 256 с.
35. Руденко В. Д. Інформатика (профільний рівень): підруч. для 11 кл. закл. Загальн. серед. освіт. Харків : Вид-во «Ранок», 2019. 256 с.
36. Скільки мов використовують українські розробники та як їх комбінують. Аналітика. URL**:**https://dou.ua/lenta/articles/how-many-languages-do-ukrainian-devs-use/
37. Тріщук І.В. Інформатика : підручник для 5 кл. закладів загальн. середн. освіти. Тернопіль : Навчальна книга. Богдан, 2023. 248 с.
38. Тріщук І. В. Інформатика : підручник для 6 кл. закладів загальн. середн. освіти. Тернопіль : Навчальна книга Богдан, 2023. 256 с.
39. Мова програмування Python: Основи програмування. Навчальний посібник для студентів спеціальності 014 Середня освіта (Інформатика). Суми: СумДПУ імені А.С.Макаренка, 2023. 101 с.
40. S. Petrenko, N. Dehtiarova, K. Parfylo, O. Shovkoplias, O. Viunenko. The formation of widget development skills in the Python programming language. 47th Convention International Convention on Information, Communication and Electronic Technology, MIPRO 2024.

Information about the authors:
Petrenko Serhii Ivanovych,
Candidate of Pedagogical Sciences,
Associate Professor at the Computer Sciences Department
Sumy State Pedagogical University named after A.S. Makarenko
87, Romenska St, Sumy, 40002, Ukraine

Dehtiarova Nelia,
Candidate of Pedagogical Sciences,
Head of the department of informatics,
Associate Professor at the Computer Sciences Department
Sumy State Pedagogical University named after A.S. Makarenko
87, Romenska St, Sumy, 40002, Ukraine