Web of Scholars: Multidimensional Research Journal
ISSN: 2751-7543
Volume 04 Number 03 (2025)
https://innosci.org/wos

Article

Methodology for Organizing and Conducting Practical Training in Teaching The Graphics Module of a Programming Language

Qulliyev Javohirbek G'anijon O'g'li*[1]

1. Teacher of the Department of "Mathematics and Informatics" Bukhara State Pedagogical Institute
* Correspondence: javohr00@gmail.com

Abstract: The development of modern information technologies also requires new approaches to the methodology of teaching programming languages. In particular, graphic modules of programming allow students to create visual projects, see the result of the code in real time, and develop logical thinking. This article analyzes the methodology for organizing and conducting practical classes on teaching graphic modules of programming languages. The main purpose of the article is to develop effective methods for teaching graphic modules in programming languages, organize practical classes, and develop students' ability to work independently.

Keywords: Case Study, Modeling, Creative Work, Method, Graphics Module, Program, Task, Synonymous Codes

1. Introduction

Through high-quality teaching of programming languages in the education system, it is possible to find a solution to the problem of training qualified programmers, which is one of the most urgent problems of today. So, this can be achieved by combining modern and interesting features of teaching and delivering it to listeners in effective methods.

To date, various scientific studies and research have been conducted in Europe, America and East Asian countries on teaching programming languages. The above studies have been carried out, created or proposed, but the imperfection of these methods has been recognized by experts. That is, a general teaching method based on a clear conclusion has not yet been created [1]. However, significant conclusions have been drawn from the research work carried out.

The biggest problem for students is not understanding the main terms, but learning to use them correctly. By clarifying the problems, it is possible to choose the right teaching method, change it or create a new one. Advantages of working with graphics modules [2]. Graphics modules of programming languages provide students with the following advantages: Visual understanding - The ability to directly see the result of the code through a graphical interface.

Creative approach: Students gain freedom in creating their own projects; Practical results: Students learn to create real-life graphic applications.

As a result of studying a number of studies, it is recommended to organize practical classes on graphics modules in the following stages:

Providing theoretical concepts - Explaining the basics of graphic programming, modules and their capabilities.

Citation: O'g'li, Q. J. G. Methodology for Organizing and Conducting Practical Training in Teaching The Graphics Module of A Programming Language. Web of Scholars: Multidimensional Research Journal 2025, 4(3), 96-103.

Received: 26th Mar 2025
Revised: 30th Mar 2025
Accepted: 6th Apr 2025
Published: 12th Apr 2025

Teaching the basics of program coding - Students are given tasks to create simple graphic elements.

Conducting interactive classes - Students are given the opportunity to create visual interfaces and develop their own projects.

Independent work and projects - Students are given tasks to create graphic applications close to real life.

Assessment of results - The level of knowledge of students is assessed through tests, practical assignments, and project defense [3], [4].

2. Materials and Methods

Using educational technologies for teaching programming languages or teaching topics, you can use several interactive methods. For example: “Modeling”, “Creative work”, “Case study” or “Synonym code analysis”, “Work with small groups”. When working with small groups, it is advisable to give students tasks by creating a problem situation, or from the above methods, the form of conducting a practical lesson using the “Case study” method can be organized as follows:

Working with buttons in the Tkinter module

The task given to students: Develop a program for the given tasks using standard components.

Figure 1 describes a simple graphical user interface (GUI) output for a basic programming task using the tkinter library in Python. The window displays a message titled "Hello World" with the text "Hello World!" printed inside the GUI window. This figure serves as a visual example for students to understand the expected outcome when completing an introductory task of creating a test program that displays text on the screen. The figure illustrates the appearance of a basic application window generated by executing the Tk() method and launching it with the mainloop() method. It acts as a reference for the practical task outlined in the document, helping students visualize the final result of the GUI-based application they are required to build [5].

Figure 1. Sample for tasks

Case tasks:

1) Display information about yourself on the screen.

2) Write the program code to create the window shown below:

a. Create a test and display the questions and answers on the screen.
b. Required equipment: Personal computer, teaching aids.
c. Required software application: vscode, spyder or other editor.

Based on the above tasks, the first task is organized as follows:

Steps to complete the task

a. Get acquainted with the necessary information;
b. Get acquainted with the program in a practical way based on the information;
c. Develop an algorithm for building the program and theoretical modeling;
d. Build a program for the given task;
e. Make sure that the finished program is working correctly.

Theoretical information and key program card: Tk() is the main window that tkinter offers the method “Tk(screenName=None, baseName=None, className='Tk', useTk=1)”. To change the name of the window, you can change the className to whatever you want. The main window of the application can be created as follows:

There is a method known as mainloop(), which is used when your application is ready to run. mainloop() is an infinite loop that is used to start the application, wait for an event to occur, and restart it if the window is not closed [6].

Figure 2 describes the appearance of the main window generated by the tkinter library in Python before any widgets, such as buttons, have been added. This window serves as the parent or base window, upon which additional interactive elements can be placed. The figure demonstrates the default empty state of the GUI window created using the Tk() class. In the context of the accompanying explanation, this window is where buttons and other widgets will later be inserted using the Button() widget. This serves as a foundational example to help learners understand the structure and behavior of the main application window when building interactive GUI applications with Python's tkinter.

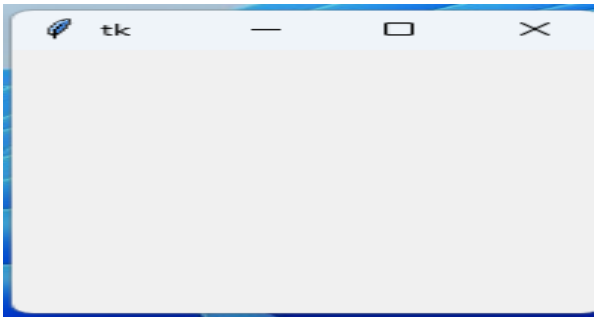

Figure 2. Main window.

Button button. Buttons are used to perform actions that are expected to occur when a button is clicked (for example, calculations or operations to be performed). This widget is used to add a button to your application. General syntax:

w=button(master, option=value)

Master is a parameter used to display the parent window. There are a number of options that can be used to change the format of the buttons. The number of options can be passed as comma-separated parameters.

Figure 3 describes the graphical user interface (GUI) state after the creation and activation of a button labeled "Stop" within a tkinter application. The window now displays the button centrally, allowing users to interact with it. The figure visually demonstrates the result of successfully coding and placing a button inside the main application window. When this "Stop" button is pressed, it triggers a predefined action — in this example, it executes the r.destroy() command, which terminates the application. This figure serves as a practical example for learners to understand how to create interactive components within the GUI and how button events are linked to specific program functions, such as closing the window or performing other operations [7], [8].

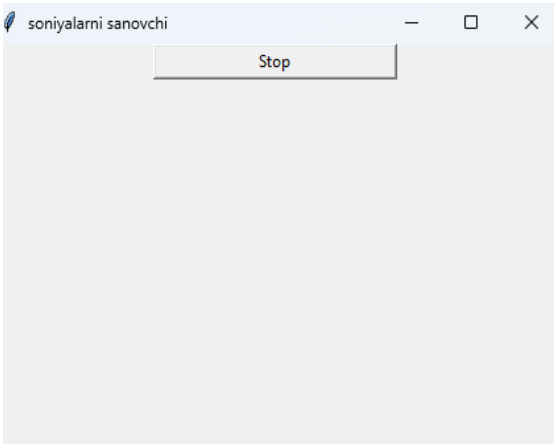

Figure 3. The state after pressing the button.

We can use the above view not only to perform actions on the stop button, but also on other buttons, as well as to give various decorations to the programming area. So, it serves to perform the processes that should be expected as a result of pressing the button. Let's look at several examples of this button.

Create a program that executes the exit command from the program.

```
import tkinter as tk
r = tk.Tk()
r.title('soniyalarni sanovchi')
button = tk.Button(r, text='Stop', width=25, command=r.destroy)
button.pack()
r.mainloop()
```

Writing the code for the above task.

Working with the Label component: The Label component serves not only to place data on the screen, but can also be used to display the results of the program [9]. To do this, the program points to a screen window where you can put any text or image that can be updated at any time by the code. General syntax:

w=Label(master, option=value)

master is the parameter used to display the main window.

If there is more than one, it will be displayed as “Label2” and so on until “LabelN”.

Let's look at how to place data on the screen and output the program's output using the Label button.from tkinter import *

```
root = Tk()
w = Label(root, text='o_zim aqimda malumot !')
w.pack()
root.mainloop()
```

Figure 4 describes the state of the graphical user interface after placing and configuring a label component within a tkinter application. The figure shows a simple window containing the text "o'zim haqida ma'lumot!", which is displayed using the Label2 component. This result is achieved after successfully creating a label and associating it with a button action. The process involved creating both a label and a button, then programming the button to display the specified text inside the label when clicked. The figure illustrates how labels are used to output information within the GUI window, and it demonstrates the interaction between button controls and label components to update

or present user-defined messages dynamically. This example helps learners understand the role of labels in GUI design for displaying static or dynamically updated text.

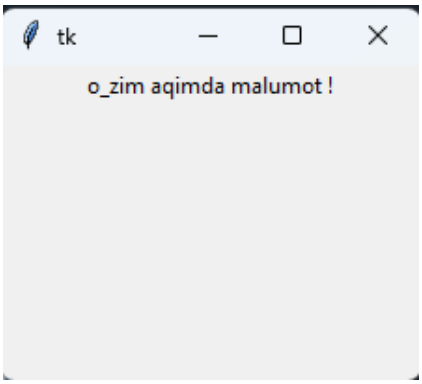

Figure 4. The state after placing the button.

1. We create the “Label2” component by selecting the same path for the second time.
2. We create the “Button1” button and enter the sentences.
3. We double-click on the Button1 (calculation) item and enter the sentence “Label2.:= ‘ ” in the programming field.
4. After entering the above sequence of actions, we run the program and get the following result. As a result, the following view is created.

Figure 5 describes the state of the application window after a button is pressed to display additional user information using the tkinter GUI framework. The figure shows a fully structured window with formatted text and a button. The window contains static information about a person, including their name, age, profession, and address, which are displayed using the Label component. Additionally, there is a button labeled "Qo'shimcha ma'lumot" (Additional Information) that, when clicked, triggers a messagebox.showinfo() function, which is designed to open a pop-up message box containing more information. This figure illustrates how multiple GUI elements such as labels, buttons, and pop-up messages can be integrated into a simple Python GUI application to create an interactive and user-friendly interface.

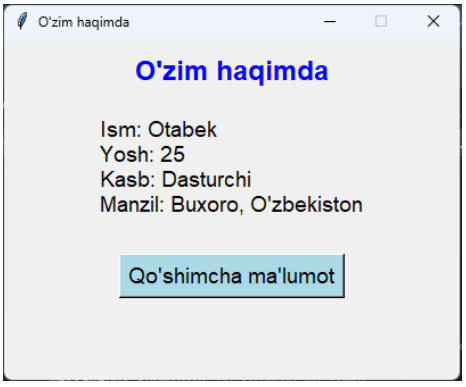

Figure 5. The state after pressing the button.

Code explanation:

1. tk.Tk() – Creates the main window.
2. Label – Used to display text.
3. Button – Displays additional information when the user clicks the button.
4. messagebox.showinfo – Opens a separate window and displays a message.

Using the program:

1. Save your code, for example, with the name o_zim_haqimda.py.

2. Execute the following command in the Python environment:

python o_zim_haqimda.py

Result:

1. A graphical interface called “O_zim” opens in the window.
2. It contains information about name, age, profession and address.
3. When you click the “Qo‘shimcha ma’lumotlar” button, a new small window opens and additional information is displayed.

3. Results

The results of the study showed that lessons organized on the basis of graphic modules of programming led to the following results:

a. The level of students' mastery increased by 30-40%;
b. The ability to work independently increased by 45%;
c. Students' motivation and interest increased by 50%.

Organizing practical exercises based on an interactive approach in teaching graphic modules of programming languages gives effective results. When students are given the opportunity to create visual and interactive projects, their interest increases and a solid assimilation of knowledge is achieved [10].

In the future, it is planned to conduct research on further improving this methodology and integrating it with various disciplines.

Organizing practical exercises using graphic modules not only increases students' interest in programming, but also serves to develop their independent thinking and problem-solving skills. Further improvement of this methodology and its integration into educational programs may be an important factor in improving the quality of programming education in the future.

4. Discussion

Interactive technologies of teaching are formed primarily on the basis of the following criteria:

Mastering the methods and techniques of obtaining, processing and using information relevant to their professional activities;

Information literacy, which consists of computer literacy and information search skills, the use and evaluation of information, possession of computer communication technologies, and the acquisition and use of the capabilities of information technologies in all areas of pedagogical activity;

When conducting lessons based on innovative technologies, the teacher can use various technical means (projector, electronic board, computer, tablet, etc.). Each teacher can introduce innovations in education individually [11], [12].

Exchange of experience and increasing creativity are important motivating factors for students and the shortest way to gain new experience. From this it can be concluded that the use of methods that allow the exchange of experience ensures the assimilation of new knowledge in programming. One of these methods is the Synonym Codes Method or the methods discussed above [13], [14]. The main goal of this method is to develop the skills of reading, understanding, analyzing and simplifying a program by replacing similar parts of the program with each other, identifying single-valued program blocks and selecting the optimal one from them. This method can also be used to implement reflection.

The listeners will have to make as many substitutions as possible. The winner will be the one who made the most substitutions. Depending on the size of the program, a time limit may be set. At the end of the time, the listener who made the most substitutions will explain his answers and make a presentation [15], [16]. After that, the remaining listeners will provide additional answer options. In order to allow for more substitutions, the following can be used in the program.

In general, the use of innovative technologies in teaching programming languages gives good results. It interests students in programming languages and increases their self-confidence. By conducting lessons using innovative technologies, it is possible to ensure that students and teachers work together, and the student acquires scientific knowledge, develops knowledge, skills and abilities.

5. Conclusion

Today, teaching "Programming Languages" in educational institutions is one of the most urgent problems. To solve it, this article reveals the methodology for teaching programming languages. The main form and content of teaching a programming language are shown. Reasoned recommendations for teaching programming languages are developed;

As a general conclusion, it can be said that using the recommendations for teaching programming technologies given in the graduation thesis would help in teaching programming languages. Organizing practical exercises based on an interactive approach in teaching graphics modules of programming languages gives effective results. When students are given the opportunity to create visual and interactive projects, their interest increases and a solid assimilation of knowledge is achieved.

REFERENCES

[1] M. A. Bobojonova, *Python dasturlash tili: o‘quv qo‘llanma*. Buxoro: Sadriddin Salim Buxoriy, 2023, 108 b.

[2] Sh. A. Nazirov, R. V. Qobulov, M. M. Musayev, and A. N. Nematov, *Delphi tilida dasturlash asoslari*. Toshkent: G’afur G’ulom nomidagi nashriyot-matbaa ijodiy uyi, 2007, 280 b.

[3] N. N. Zaripov, "Maktab o‘quvchilariga dasturlash muhitini o‘qitishda innovatsion texnologiyalar yordamida ta’lim samaradorligini oshirish yo‘llari," 443, 124 b.

[4] S. A. Normatov, *Oliy ta’lim tizimida zamonaviy dasturlash tillarini o‘qitish usullari*.

[5] K. N. Tog‘ayeva, *Ta’lim metodlari va vositalari unumli foydalanish yo‘llari*. Toshkent: Alisher Navoiy nomidagi Toshkent davlat o‘zbek tili va adabiyoti universiteti.

[6] N. D. Paxlavonovich, "The Role of Artificial Intelligence Systems in Teaching Informatics," *Int. Conf. on Adaptive Learning Technologies*, vol. 15, pp. 50–52, Mar. 2025.

[7] D. B. Hasanova, "Kasb-hunar maktablarida axborot-kommunikatsion texnologiyalari fanini o‘qitishning o‘ziga xos metodlari," *Fan, ta’lim va innovatsiyalar: nazariya, amaliyot va natijalar*, 2024.

[8] N. D. Paxlavonovich, "The Problem of Innovative Activity in the Organization of Education on the Base of Information Technologies in General Education Schools," *Multidisciplinary Journal of Science and Technology*, vol. 4, no. 6, pp. 634–636, 2024.

[9] A. M. Ahmedov, "Qo‘qon davlat pedagogika instituti o‘quvchilarda dasturlash ko‘nikmalarini shakllantirish omillari," *Ekonomika i Sotsium*, no. 2(81), ch. 2, 2021.

[10] D. P. Nabiyev, "Dars jarayonida ma’lumotlar bazasidan foydalanish metodikasi," *Inter Education & Global Study*, no. 2, pp. 355–362, 2025.

[11] "https://bestpublication.org/index.php/iq/article/view/420", [Accessed: Apr. 11, 2025].

[12] Q. Javohirbek G‘anijon o‘g‘li, "Dars jarayonida axborot texnologiyalaridan foydalanishning samaradorligi," *PEDAGOG*, vol. 8, no. 3, pp. 99–102, 2025.

[13] "https://eduproger.uz/blog/dasturlash-asoslari-python", [Accessed: Apr. 11, 2025].

[14] Q. Javohirbek G‘anijon o‘g‘li, "Python dasturlash tilining turtle modulidan foydalanib geometrik shakllarni chizish," *PEDAGOG*, vol. 7, no. 10, pp. 33–39.

[15] H. D. Z. Bo‘ronovna and A. Amonova, "Bulutli texnologiyalar," *Formation of Psychology and Pedagogy as Interdisciplinary Sciences*, vol. 3, no. 30, pp. 101–106, 2024.

[16] Q. Javohirbek G‘anijon o‘g‘li, "Dars jarayonida raqamli texnologiyalardan foydalanish metodikasi," *O‘zbekistonda Fanlararo Innovatsiyalar va Ilmiy Tadqiqotlar Jurnali*, vol. 4, no. 38, pp. 48–50, 2025.