Дегтярьова Н., Петренко С., Удовиченко О. Робота з графічними віджетами при вивченні мови програмування Python в закладах загальної середньої освіти. *Освіта. Інноватика. Практика,* 2023. Том 11, № 4. С. 26-34. DOI: 10.31110/2616-650X-vol11i4-004

Dehtiarova N., Petrenko S., Udovychenko O. Robota z hrafichnymy vidzhetamy pry vyvchenni movy prohramuvannia Python v zakladakh zahalnoi serednoi osvity [Working with graphic widgets when learning the Python programming language in institutions of general secondary education]. *Osvita. Innovatyka. Praktyka – Education. Innovation. Practice,* 2023. Vol. 11, No 4. S. 26-34. DOI: 10.31110/2616-650X-vol11i4-004

DOI: 10.31110/2616-650X-vol11i4-004

Неля ДЕГТЯРЬОВА
Сумський державний педагогічний університет імені А.С. Макаренка, Україна
https://orcid.org/0000-0001-9590-4915
degtyarevanv@fizmatsspu.sumy.ua
Сергій ПЕТРЕНКО
Сумський державний педагогічний університет імені А.С. Макаренка, Україна
https://orcid.org/0000-0002-3089-6499
s.petrenko@fizmatsspu.sumy.ua
Ольга УДОВИЧЕНКО
Сумський державний педагогічний університет імені А.С. Макаренка, Україна
https://orcid.org/0000-0002-3401-3251
udovich_olga@fizmatsspu.sumy.ua

РОБОТА З ГРАФІЧНИМИ ВІДЖЕТАМИ ПРИ ВИВЧЕННІ МОВИ ПРОГРАМУВАННЯ PYTHON В ЗАКЛАДАХ ЗАГАЛЬНОЇ СЕРЕДНЬОЇ ОСВІТИ

Анотація. *Робота присвячена методичним аспектам навчання учнів програмуванню мовою Python. Наведено результати опитування вчителів щодо особливостей навчання програмуванню в закладах загальної середньої освіти, де з'ясовано, що проблеми мотивації та зацікавленості учнів програмуванням актуальні і на даний час, більшість вчителів не використовує віджети або роботу з графічними об'єктами при навчанні програмуванню, вчителі відмічають складнощі в опануванні програмування учнями, в школах програмування вивчається в різних середовищах та за допомогою різних мов програмування, проте переважає в 8 та 9 класах вивчення саме мови Python.*

Мова Python має велику кількість бібілотек та вбудованих функцій, проте вивчення в школах часто обмежується вивченням саме математичних функцій і поза увагою залишаються ті бібліотеки, які сприятимуть зацікавленості учнів та підвищенню їх мотивації у вивченні програмування. Робота з віджетами дає змогу учню отримати реальний продукт як результат своєї роботи, демонструє прикладне застосування мови програмування.

У роботі наведено окремі функції роботи з графічними об'єктами та запропоновано приклади розробки елементів віджетів. Авторами створені методичні рекомендації для студентів педагогічних спеціальностей, де укладено теоретичний матеріал та розроблені авторські та наведені існуючі приклади для формування практичних навичок програмування. Зокрема, приділено увагу бібліотеці графічних елементів tkinter.

Ключові слова: *віджети; мова програмування Python; програмування в школі; методика навчання інформатики; бібліотека tkinter.*

Nelia DEHTIAROVA
Sumy State Pedagogical University named after A.S. Makarenko, Ukraine
https://orcid.org/0000-0001-9590-4915
degtyarevanv@fizmatsspu.sumy.ua
Sergii PETRENKO
Sumy State Pedagogical University named after A.S. Makarenko, Ukraine
https://orcid.org/0000-0002-3089-6499
s.petrenko@fizmatsspu.sumy.ua
Olga UDOVYCHENKO
Sumy State Pedagogical University named after A.S. Makarenko, Ukraine
https://orcid.org/0000-0002-3401-3251
udovich_olga@fizmatsspu.sumy.ua

WORKING WITH GRAPHIC WIDGETS WHEN LEARNING THE PYTHON PROGRAMMING LANGUAGE IN INSTITUTIONS OF GENERAL SECONDARY EDUCATION

Abstract. *The work is devoted to the methodical aspects of teaching students programming in the Python language. The results of a survey of teachers regarding the peculiarities of teaching programming in institutions of general secondary education are presented. It was found that the problems of students' motivation and interest in programming are still relevant at the present time, most teachers do not use widgets or work with graphic objects when teaching programming, teachers note difficulties in mastering programming by students. Programming in schools is studied in different environments and with the help of different programming languages, but Python is the predominant language in the 8th and 9th grades.*

The Python language has a large number of libraries and built-in functions, but the study in schools is often limited to the study of mathematical functions and those libraries that will contribute to the interest of students and increase their motivation in learning programming are neglected. Working with widgets allows the student to get a real product as a result of his work, demonstrates the applied application of the programming language.

The work presents individual functions of working with graphic objects and offers examples of widget element development. The authors have created methodological recommendations for students of pedagogical specialties, which include theoretical material and developed original and existing examples for the formation of practical programming skills. In particular, attention was paid to the library of tkinter graphic elements.

Keywords: *widgets; Python programming language; programming at school; computer science teaching method; tkinter library.*

Постановка проблеми. Популярність мови програмування Python в середовищі системних програмістів та затребуваність відповідних фахівців на ринку праці, посилили актуальність її вивчення в закладах загальної середньої та вищої освіти. Затребуваність мови Python забезпечується сукупністю характерних можливостей, які не у повній мірі можуть бути реалізовані в середовищах інших мов:

– багатоцільова направленість мови програмування, призначена для розв'язування широкого спектру задач;

– програми створені на Python можна запускати на різних операційних системах;

– програмний код добре читається;

– мінімальна кількість розділових знаків;

– неявна строга динамічна типізація даних (при оголошенні змінної її тип не вказується);

– наявність в структурі мови великої кількості бібліотек та внутрішніх функцій.

Сукупність цих характерних особливостей дає змогу спростити не тільки сам процес програмування, а й полегшити вивчення мови.

На даний час мова Python вивчається у значній кількості шкіл України. А це потребує відповідний рівень підготовки учителів з програмування. Таким чином, введення дисциплін з вивчення мови програмування Python у освітній процес педагогічних закладів вищої освіти стає органічним.

Проте ще реалізується в багатьох школах застарілий підхід вивчення мови програмування виключно на прикладах математичних виразів. Так, дійсно, за їх допомогою вчитель має велику кількість прикладів для відображення різних структур програм: лінійних, з розгалуженням, з циклами. Проте учням стає нецікавим таке наповнення. Тому і розробляється велика кількість середовищ, де відходять від розгляду виключно математичних прикладів. Тим більше, що сучасні мови програмування мають величезну кількість бібліотек, модулів, які надають можливість продемонструвати інші аспекти програмування учням, зокрема, роботу з графічними об'єктами і, тим самим, значно підвищити їх мотивацію.

Аналіз останніх досліджень та публікацій. Навчання програмуванню в школі було і залишається дискусійною темою для багатьох вчителів, науковців та методистів. Обґрунтування щодо вибору навчальної мови програмування саме вчителем знаходимо в роботі Лапінського В. [6]. Критеріям добору веб-орієнтованих технологій навчання основ програмування присвячено роботу Спіріна О.М. та Вакалюк Т.А. [8]. Питання вивчення основ програмування мовою Python в закладах вищої освіти України розглядалося в дослідженнях науковців, серед яких: Анісімов А., Дорошенко А., Жуковський С., Козуб Г., Семенов Н. [4], Костюченко А. [5], Матвійчук С., Погорілий С., Яковенко А. [9] та інші.

Зазначені автори пропонують традиційну методику отримання навичок програмування на прикладах розв'язання математичних задач, запропоновану А. Єршовим в 80-х роках минулого століття. Такий підхід передбачає певні труднощі в вивченні основ програмування у здобувачів з недостатньою математичною підготовкою. У даній роботі пропонується розпочати вивчення програмування з побудови елементів графічних віджетів. Такий підхід дозволяє зробити процес отримання первинних навичок програмування досить наглядним та інформативним, і спонукає до самостійного заглиблення в середовище мови.

Мета дослідження: обґрунтувати актуальність вивчення мов програмування в процесі роботи з графічними віджетами на прикладі мови програмування Python.

Досягнення мети вбачаємо реалізацією таких задач:

– оприлюднити результати опитування вчителів щодо особливостей навчання програмуванню в закладах загальної середньої освіти;

– описати основні відомості для роботи з віджетами мовою програмування Python;

– запропонувати авторські та розглянути оприлюдненні іншими дослідниками приклади розробки елементів віджетів.

Методи дослідження. Теоретичні методи дослідження: аналіз і систематизація наявних рекомендацій щодо вивчення програмування в закладах загальної середньої освіти, аналіз джерел з програмування, що знаходяться у вільному доступі, аналіз результатів опитування. Було проведено

анкетування 26 практикуючих вчителів інформатики закладів загальної середньої освіти м. Суми та Сумської області.

Виклад основного матеріалу дослідження. Традиційний підхід до вивчення основ програмування будь-якої мови, в основному, передбачає використання вбудованих в середовище програмування математичних функцій. А це обмежує використання вбудованих в мову Python бібліотек модулями math (бібліотека математичних функцій) та random (бібліотека функцій для генерації випадкових елементів).

Процес навчання, який базується на розв'язуванні математичних задач, не є наближеним до життєвих ситуацій і не зацікавлює учнів. Це підтверджує і результати опитування учителів 17 різних закладів загальної середньої освіти у м. Суми та Сумської області.

Анкетовані зазначили, що в школах з 5 по 9 клас вивчають одну мову програмування (46,2%) або різні мови чи середовища (53,8). У школах, де вивчаються різні мови програмування показники розподілилися таким чином:

– навчають учнів програмуванню та алгоритмізації засобами середовища Scratch у 5 класах (92,3%), у 6 класах (88,5%), у 7 класах (34,6%);
– вивчають програмування у середовищі Lazarus у 5 класах (3,8%), у 6 класах (7,7%), у 7 класах (11,5%);
– вивчають Python у 5 класах (3,8%), у 6 класах (3,8%), у 7 класах (53,8%).

Вважаємо за доцільне зауважити, що на вибір мови та середовища програмування надавалися варіанти відповідей також: C++, Java, Pascal. У відповідях про вивчення програмування з 5го по 7 клас жоден з цих варіантів не був обраний респондентами.

Цікавим є подальший розподіл уваги до мов програмування у 8 та 9 класах, який представлено на рис. 1.

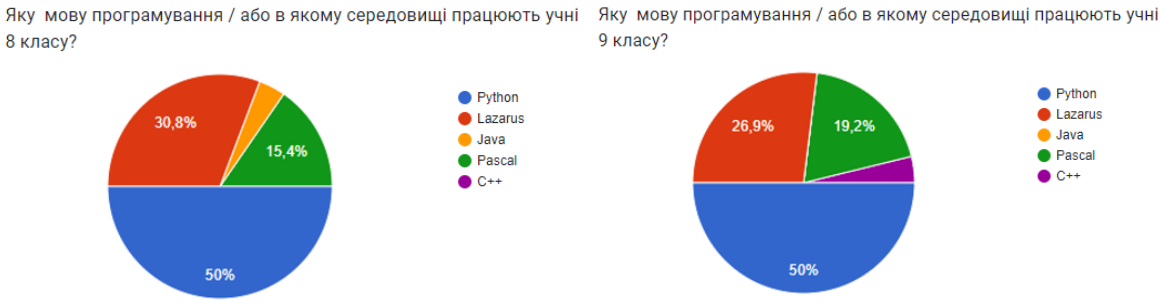

Рис. 1. Розподіл відповідей на питання «Яку мову програмування вивчають або у якому середовищі працюють учні 8 та 9 класів»

Тут уже з'являється вивчення мови програмування Pascal в 8му (15,4%) та 9му (19,2%) класах. Також Java і C++ обрав 1 анкетований, що становить 3,8%. Середовище Lazarus використовують на уроках інформатики 30,8% анкетованих в 8му та 26,9% в 9му класах. Python обрали понад 50% респондентів, зазначаючи цю мову програмування при вивченні у 8х та 9х класах.

Серед проблем при вивченні розділу алгоритмізація та програмування в різних класах вчителі зазначили такі:

– недостатній рівень розвитку алгоритмічного мислення – 65,4%;
– учень не уміє використовувати готовий розроблений алгоритм для нової задачі (462%);
– учень не уміє скласти програму (38,5%);
– учень не розуміє саме опис циклів (34,6%);
– учень не розуміє розгалуження (19,2%);
– учень не уміє тестувати готову програму (15,4%).

Серед причин виникнення таких проблем учителі відмічають:

– у учнів відсутнє бажання працювати в цілому, що стосується не тільки інформатики (73,1%);
– учень не готовий до кропіткої роботи перевірки та виправлення помилок програми (53,8%);
– програмування не зацікавлює учнів (46,2%);
– відсутність мотивації (26,9%) та важка мова для опанування (26,9%).

Окрім зазначених варіантів учителі також зауважили, що програмування не відповідає віковим особливостям учнів, учні не бачать результатів та застосування результатів програмування, що і зменшує мотивацію до його вивчення. Також 92,3% опитуваних відповіли, що вони використовують переважно математичні вирази при поясненні нового матеріалу і на практичних роботах при вивченні програмування. А на запитання про віджети відповіді розподілилися так, як продемонстровано на рис. 2.

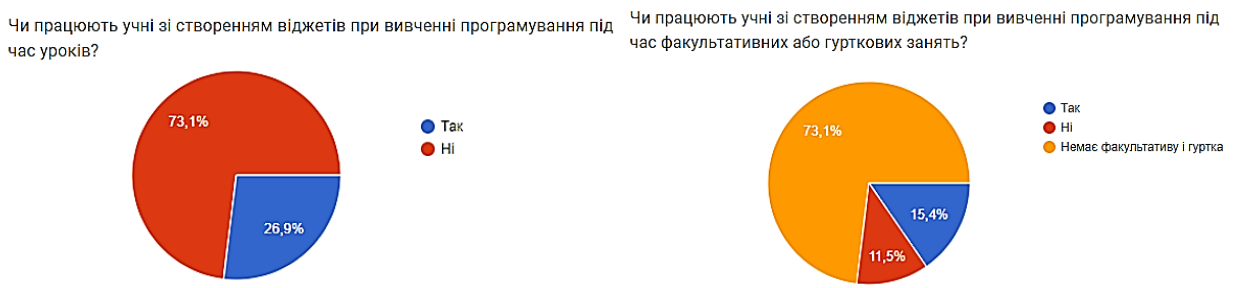

Рис. 2. Розподіл відповідей на питання про роботу учнів з віджетами

Таким чином, переважна більшість учителів не пропонує учням роботу з графічними об'єктами, а працює з математичними прикладами, які не мотивують учнів. Така робота передбачає розуміння математичних виразів та їх особливостей розв'язування, якими володіють не усі учні.

Проблема вибору демонстраційних прикладів та практичних завдань є актуальною. В методичних матеріалах, журналах, підручниках пропонують невелику кількість прикладів програм для роботи з віджетами. Частіше зустрічаються задачі щодо обчислення. Для прикладу візьмемо задачу: написати код програми для розрахунку ідеальної маси чоловіка та жінки за формулою Брокка з використанням вбудованих функцій.

Формула Брокка :

Ідеальна вага для чоловіка = (зріст в сантиметрах – 100) * 1,15.

Ідеальна вага для жінки = (зріст в сантиметрах – 110) * 1,15.

Вирішенням цієї задачі може бути такий код програми:

```
s=input("Введіть вашу стать: Ч-чоловік, Ж-жінка ")
r=int(input("Введіть ваш зріст "))
if s==('Ж'):
        a=print('ідеальна вага ',int((r-110)*1.15))
elif s==('Ч'):
        b=print('ідеальна вага ',int((r-100)*1.15))
```

і результатом її виконання стає:

```
Введіть вашу стать: Ч-чоловік, Ж-жінка Ч
Введіть ваш зріст 185
ідеальна вага 97
>>>
```

Така задача дуже часто зустрічається при навчанні програмуванню. У той же час Python має значну кількість бібліотек, які містять уже заготовлені частини програм та залишаються поза увагою. Щоб ними скористатися треба знати про їх існування, імпортувати у програму та відповідним чином викликати доступні у модулі функції. Перелік бібліотек та доступних у них функцій можна отримати з загальнодоступних онлайн джерел та підручників [1-3].

Зміщення акцентів розв'язання задачі з математичної складової на візуальну привертає увагу та зацікавленість з боку здобувачів, тому що перевага віддається не розбору математичного процесу, а створенню графічного віджету засобами мови програмування. Створення такого віджету передбачає отримання нових знань і умінь. Для створення віджету необхідно створити вікно, мітку, текстове поле, кнопку, перемикач, шкалу і енергомічно розташувати їх у вікні.

Авторами була створені методичні рекомендації для студентів педагогічних спеціальностей, де укладено теоретичний матеріал та розроблені авторські та наведені існуючі приклади для формування практичних навичок програмування. Зокрема, приділено увагу бібліотеці графічних елементів tkinter [7]. На даний час триває впровадження розроблених методичних рекомендацій в закладах загальної середньої освіти м. Суми та Сумської області. В перспективі вбачаємо доречним розширити перелік закладів з урахуванням результатів даного впровадження та оновленням методичних рекомендацій.

Наведемо приклад опанування об'єктів та роботи з ними. Перед створенням об'єктів з графічної бібліотеки tkinter є необхідність розглянути питання їх розташування у вікні. Це питання важливе, тому що в залежності від вибору, так званого, пакувальника залежить зручність їх розташування.

В бібліотеці tkinter передбачено для розташування віджетів пакувальники або менеджери геометрії: *pack, place ma grid*.

1) pack() – цей пакувальник використовують для розміщення віджетів один за одним (зліва направо або зверху вниз). Властивість *side* може вказати до якої сторони основного віджета має примикати створюваний віджет. Пакувальник *pack()* може вести себе непередбачувано при використанні на різних платформах. Параметрами пакувальника *pack()* є side (top, bottom, left, right), fill (x,y), anchor (N,W,S,E).

2) Пакувальник *place()* – дозволяє розміщувати віджет в фіксованому місці з фіксованим розміром. При використанні цього пакувальника необхідно вказувати координати кожного віджета. Він може здатися незручним, але надає повну свободу при розміщенні віджетів у вікні. Методом *place()* віджету вказується його положення або в абсолютних значеннях (в пікселях), або в частках батьківського вікна. Отже абсолютно і відносно можна задавати і розмір самого віджета. Параметри пакувальника *place()* є anchor (N, NE, E, SE, SW, W, NW), relwidth/relheight, relx, rely (x,y), width, height, x/y.

3) Пакувальник *grid()* – розміщує віджети в двомірній сітці (таблиці), наприклад для створення сітки кнопок для калькулятора. Розміщення віджета в тій чи іншій комірці задається через аргументи *row*(рядок) і *column* (стовпчик).

На практичних прикладах розглядаються способи розташування графічних об'єктів у вікні пропонується при знайомстві з віджетом Canvas. Він представляє собою полотно для побудови геометричних фігур (аналог графічного екрана в інших мовах програмування). Задається полотно функцією Canvas (). В дужках вказується назва вікна і параметри.

Синтаксис створення полотна: *змінна=Canvas(ім'я_віджету, параметри)*

Для розуміння здобувачами різниці при розміщення віджетів методами pack() і place() пропонується завдання для самостійної роботи: створити полотно довільних розмірів, на якому зобразіть усі геометричні примітиви і малюнок тексту, який складається з власного імені і прізвища. Малюнки потрібно виконати двома методами *pack()* і *place()*. В результаті виконання роботи потрібно описати як змінюються малюнки в кожному із методів при *збільшенні розмірів полотна*.

Результатами роботи має бути:

- код програми з використанням методу *pack();*
- код програми з використанням методу *place()*;
- скріншоти полотна для обох методів;
- аналіз результатів роботи.

Віджет Label (мітка) використовується для відображення будь якого тексту у вікні і виконує інформаційну роль (повідомлення, підпис елементів інтерфейсу тощо). Синтаксис:

ім'я мітки = Label(ім'я вікна)

Змінити властивість мітки або будь-якого іншого віджета можна одним із способів:

Ім'я_віджета["ім'я_властивості"] = значення;

Ім'я_віджета.config(ім'я_властивості = значення).

Прикладом може бути класичне створення мітки з текстом «Hello world!», що долучає здобувачів до початківців-програмістів і створює особливу атмосферу вивчення програмування.

```
from tkinter import *        #імпорт графічної бібліотеки
window=Tk()                  #створення головного вікна
window.title ('Графічна програма')          #створення назви головного вікна
widget = Label(window, text='Hello world!')  #створення мітки з іменем widget, яка виведе на екран взятий у лапки текст
widget.pack()                # розміщення мітки на екрані за допомогою пакувальника pack
```

Результат представлено на рис.3

Рис. 3. Результат виконання програми

Цікавими для створення є робота з перемикачами Radiobutton. Синтаксис: Ім'я перемикача = Radiobutton(ім'я вікна). За замовчуванням перемикачі, що розміщені в одному вікні, не пов'язані між собою і можуть бути увімкнені одночасно. Зв'язок доцільно встановити через спільну змінну, різні значення якої будуть відповідати увімкненню різних перемикачів із однієї групи. Для всіх перемикачів цієї групи встановлюється одна і та ж змінна для властивості variable. А властивості value присвоюються різні значення цієї змінної.

Приклад. Створення перемикачів.

```
var = IntVar()          # створюємо змінну, яка приймає цілі значення
var.set(1)              # встановлюємо перше значення для створеної змінної

rad1 = Radiobutton(root, text="Перша", variable=var, value=1)      # створюємо перемикач зі значенням 1
rad2 = Radiobutton(root, text="Друга", variable=var, value=2)      # створюємо перемикач зі значенням 2
rad3 = Radiobutton(root, text="Третя", variable=var, value=3)      # створюємо перемикач зі значенням 3
# розміщуємо перемикачі у вікні
rad1.pack()
rad2.pack()
rad3.pack()
```

Приклад. У вікні розташувати прапорці для вибору улюблених кольорів.

```
from tkinter import * # імпорт графічної бібліотеки
window = Tk() # створення головного вікна
window.title (' Check') #створення назви головного вікна

t = Label(window, text="Оберіть декілька кольорів:") # створення текстової мітки
t.pack() # розміщення текстової мітки

var1 = IntVar() # створення змінної цілого типу для першого прапорця
check1 = Checkbutton(window, # створення першого прапорця
 text="червоний", # текст прапорця
 variable=var1, # значення змінної першого прапорця
 onvalue=1, # значення при включеному прапорці
 offvalue=0) # значення при вимкненому прапорці
check1.pack(side=LEFT) # створення першого прапорця

var2 = IntVar() # створення змінної цілого типу для другого прапорця
check2 = Checkbutton(window, # створення другого прапорця
 text="помаранчевий", # текст прапорця
 variable=var2, # значення змінної другого прапорця
 onvalue=1, # значення при включеному прапорці
 offvalue=0) # значення при вимкненому прапорці
check2.pack(side=LEFT) # створення другого прапорця

var3 = IntVar() # створення змінної цілого типу для третього прапорця
check3 = Checkbutton(window, # створення третього прапорця
 text="жовтий", # текст прапорця
 variable=var3, # значення змінної третього прапорця
 onvalue=1, # значення при включеному прапорці
 offvalue=0) # значення при вимкненому прапорці
check3.pack(side=LEFT) # створення третього прапорця

var4 = IntVar() # створення змінної цілого типу для четвертого прапорця
check4 = Checkbutton(window, # створення четвертого прапорця
 text="зелений", # текст прапорця
 variable=var4, # значення змінної четвертого прапорця
 onvalue=1, # значення при включеному прапорці
 offvalue=0) # значення при вимкненому прапорці
check4.pack(side=LEFT) # створення четвертого прапорця

var5 = IntVar() # створення змінної цілого типу для п'ятого прапорця
check5 = Checkbutton(window, # створення п'ятого прапорця
 text="блакитний", # текст прапорця
 variable=var5, # значення змінної п'ятого прапорця
```

```
 onvalue=1, # значення при включеному прапорці
 offvalue=0) # значення при вимкненому прапорці
check5.pack(side=LEFT) # створення п'ятого прапорця

var6 = IntVar() # створення змінної цілого типу для шостого прапорця
check6 = Checkbutton(window, # створення шостого прапорця
 text="синій", # текст прапорця
 variable=var6, # значення змінної третього прапорця
 onvalue=1, # значення при включеному прапорці
 offvalue=0) # значення при вимкненому прапорці
check6.pack(side=LEFT) # створення шостого прапорця

var7 = IntVar() # створення змінної цілого типу для сьомого прапорця
check7 = Checkbutton(window, # створення сьомого прапорця
 text="фіолетовийий", # текст прапорця
 variable=var7, # значення змінної сьомого прапорця
 onvalue=1, # значення при включеному прапорці
 offvalue=0) # значення при вимкненому прапорці
check7.pack(side=LEFT) # створення сьомого прапорця
```

Результат представлено на рисунку 4

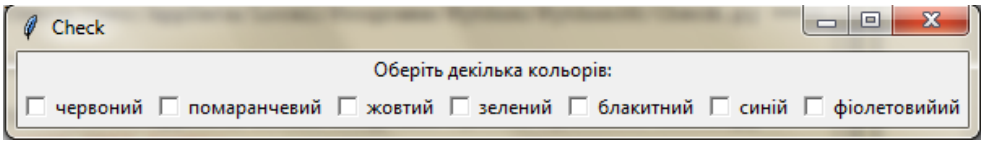

Рис. 4. Результат виконання програми

Вивчення віджету Scale (повзунок)

Scale – клас повзунка (шкала). Це віджет, який дозволяє обрати будь-яке значення із заданого діапазону.

Синтаксис:

ім'я повзунка = Scale(ім'я вікна)

Приклад. Розмістити у вікні шкалу від 0 до 10.

```
from tkinter import * # імпорт графічної бібліотеки
window = Tk() # створення головного вікна
window.title ('Check') #створення назви головного вікна

scale = Scale(window, # створення шкали
 orient=HORIZONTAL, # орієнтація повзунка
 length=300, # довжина
 from_=0, # початкове значення на шкалі
 to=10, # кінцеве значення на шкалі
 tickinterval=1, # інтервал, через який відображаються мітки на шкалі
 resolution=1) # мінімальна відстань пересування повзунка
scale.pack() # розміщення шкали
```

Результат представлено на рис 5.

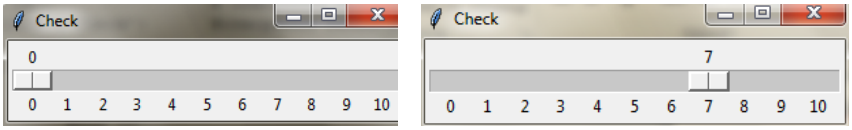

Рис. 5. Результати виконання програми

Доцільним вважаємо надавати завдання створення віджета за зразком. При цьому варто не обмежувати здобувачів щодо вибору кольорів, тексту чи іншого оформлення. Зразки для завдань представлені на рисунках 6-9.

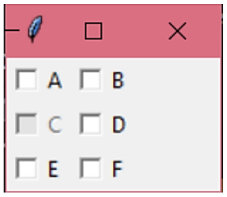

Рис. 6. Зразок для практичного завдання

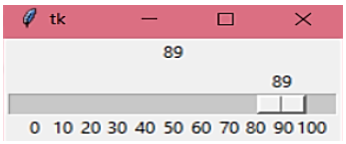

Рис. 7. Зразок для практичного завдання

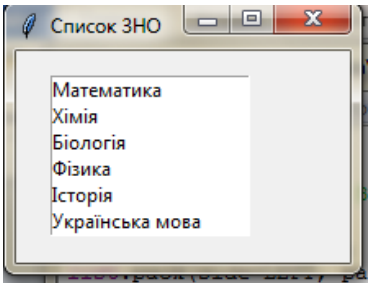

Рис. 8. Зразок для практичного завдання

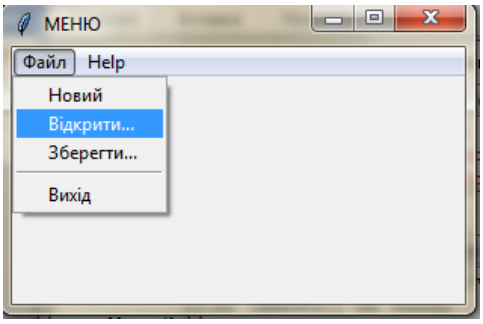

Рис. 9. Зразок для практичного завдання

Висновки. Підсумовуючи результати проведеного дослідження вважаємо за доцільне зазначити:

– при вивченні програмування в завданнях пропонується переважно робота з математичними виразами;

– для опанування програмування через роботу з математичними виразами учень повинен мати гарну математичну підготовку;

– використання виключно математичних виразів при опануванні програмування або їх переважної більшості робить нецікавим програмування для певної кількості учнів, які не мають відповідної підготовки з математики;

– опитуванні вчителів підтвердило проблеми мотивації та зацікавленості учнів програмуванням;

– більшість вчителів не використовує віджети або роботу з графічними об'єктами при навчанні програмуванню;

– опанування мови програмування Python при роботі з віджетами дає змогу учню отримати реальний продукт як результат своєї роботи, дає змогу опанувати різні бібліотеки та функції мови програмування Python, а не лише математичні, демонструє прикладне застосування мови програмування.

Дослідження щодо використання віджетів при навчанні програмуванню на даний час триває. Подальшу діяльність в цьому напрямі вбачаємо у вдосконаленні розроблених методичних рекомендацій, розширення експериментального поля впровадження на заклади інших регіонів України, створення посібника з програмування мовою Python.

Список використаних джерел

1. Python 3.10.4 documentation. URL: https://docs.python.org/release/3.10.4/
2. Swaroop Chitlur. A Byte of Python. URL: https://github.com/swaroopch/byte-of-python
3. Вчимося програмувати разом! Python – просто! URL: https://sites.google.com/comp-sc.if.ua/python-easy/
4. Козуб Г.О., Семенов Н.А. *Програмування (Python)*: метод. рек. до лаб. робіт для студ. спец. 121 – «Інженерія програмного забезпечення». Старобільськ : ДЗ «ЛНУ імені Тараса Шевченка», 2020. 108 с.
5. Костюченко А.О. *Основи програмування мовою Python*: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
6. Лапінський В. Проблема вибору першої мови програмування – сьогоднішнє бачення. *Комп'ютер в школі та сім'ї*. № 1, 2014. С. 14-17.
7. Петренко С.І., Дегтярьова Н.В. *Створення віджетів мовою програмування Python*. Методичні рекомендації. Суми: ФОП Цьома С.П., 2023. 32 с.
8. Спірін О.М., Вакалюк Т.А. Критерії добору відкритих web-орієнтованих технологій навчання основ програмування майбутніх учителів інформатики. *Інформаційні технології і засоби навчання*. 2017. Т. 60. Вип. 4. С. 275-287.
9. Яковенко А.В. *Основи програмування. Python*. Частина 1: підручник для студ. спеціальності 122 «Комп'ютерні науки», спеціалізації «Інформаційні технології в біології та медицині». Київ. КПІ ім. Ігоря Сікорського, 2018. 195 с.

References

1. Python 3.10.4 documentation. URL: https://docs.python.org/release/3.10.4/
2. Swaroop Chitlur. A Byte of Python. URL: https://github.com/swaroopch/byte-of-python
3. Vchymosia prohramuvaty razom! Python – prosto! URL: https://sites.google.com/comp-sc.if.ua/python-easy/
4. Kozub H.O., Semenov N.A. Prohramuvannia (Python): metod. rek. do lab. robit dlia stud. spets. 121 – «Inzheneriia prohramnoho zabezpechennia». Starobilsk : DZ «LNU imeni Tarasa Shevchenka», 2020. 108 s.
5. Kostiuchenko A.O. Osnovy prohramuvannia movoiu Python: navchalnyi posibnyk. Ch.: FOP Balykina S.M., 2020. 180 s.
6. Lapinskyi V. Problema vyboru pershoi movy prohramuvannia – sohodnishnie bachennia. *Kompiuter v shkoli ta simi*. № 1, 2014. S. 14-17.
7. Petrenko S.I., Dehtiarova N.V. Stvorennia vidzhetiv movoiu prohramuvannia Python. Metodychni rekomendatsii. Sumy: FOP Tsoma S.P., 2023. 32 s.
8. Spirin O.M., Vakaliuk T.A. Kryterii doboru vidkrytykh web-opiientovanykh tekhnolohii navchannia osnov prohramuvannia maibutnikh uchyteliv informatyky. *Informatsiini tekhnolohii i zasoby navchannia*. 2017. T. 60. Vyp. 4. S. 275-287.
9. Iakovenko A.V. Osnovy prohramuvannia. Python. Chastyna 1: pidruchnyk dlia stud. spetsialnosti 122 «Kompiuterni nauky», spetsializatsii «Informatsiini tekhnolohii v biolohii ta medytsyni». Kyiv. KPI im. Ihoria Sikorskoho, 2018. 195 s.