Информационные технологии в образовании. 2022. Вып 5

Использование графического интерфейса PySimpleGUI при решении учебных практикумов на Python

Векслер В.А.
vitalv74@mail.ru,
Саратовский государственный университет имени Н.Г. Чернышевского

Аннотация. В статье рассматриваются возможности визуального проектирования форм для отображения запрашиваемой и результирующей информации, посредством разнообразных графических объектов, при создании учебных проектов на языке Python. Проводится обзор графических библиотек и приводятся примеры решения задач в интерфейсе PySimpleGUI.

Ключевые слова: алгоритмизация, интерфейс, учебный проект, обучение, Python.

Алгоритмизация и программирование сегодня являются одним из важнейших направлений школьного курса информатики. Преподавателю дают возможность выбора для изучения любой язык программирования, включенный в материалы ОГЭ по информатике. Ими могут стать: алгоритмический язык, Бейсик, Pascal, C++ или Python. Обычно при изучении таких тем как «Алгоритмы и исполнители», «Алгоритмический язык», «Построение алгоритмов» использовался язык программирования Pascal. Pascal – язык, который был создан специально для того, чтобы научить ребенка программированию. Он достаточно прост для изучения и одновременно строг в синтаксических конструкциях, соответствует решению таких дидактических задач, как развитие алгоритмического мышления и формирование алгоритмической культуры. При этом в последние годы мощную конкуренцию ему составил Python – язык современного программирования, развивающийся и перспективный [6].

Особенностью языка Python стал простой и ясный синтаксис, его обширная стандартная библиотека которая включает большой объём полезных функций, расширяющих базовые языка. Он поддерживает императивную, модульную, процедурную, объектно-ориентированную и функциональную парадигмы программирования, хотя в основе своей является объектно-ориентированным языком программирования. Основными чертами языка являются динамическая типизация, автоматическое управление памятью, возможность использования дополнительных внешних библиотек. Поскольку Python – интерпретируемый язык программирования, при его изучении ребенок может легко воспользоваться методом проб и ошибок, что превратит программирование в занимательную игру.

Если же обратить внимание на визуальную составляющую, то и здесь Python может показать себя как достаточно эффективный инструмент. Добавление графического интерфейса к созданной ребенком программе делает ее более привлекательной, современной и открывает ее для более широкой аудитории.

Информационные технологии в образовании. 2022. Вып 5

Разработка на Python предлагает большой набор библиотек графического интерфейса для построения визуальной части приложения. Каждая из них имеет свои преимущества и недостатки и подходит для различных задач разработки.

Популярными библиотеками данного типа являются:

Pygame – это «игровая библиотека», набор инструментов, помогающих школьникам создавать игры. Включает инструментарий по работе с графикой и анимацией, работе со звуком, управлению игровым процессом с помощью различных устройств (мышь, клавиатура). Контролирует среду игровой цикл. Он отслеживает все происходящие события и реагирует на них [1].

Ren'py – это бесплатно распространяемый движок для создания визуальных новелл, на нем пишутся кроссплатформенные приложения. Данная библиотека может использоваться как в некоммерческих, так и в коммерческих целях [4].

Tkinter устанавливается в Python как стандартный модуль. Библиотека кроссплатформенная; элементы приложения отображаются посредством элементов операционной системы, поэтому приложения, созданные с помощью Tkinter, выглядят так, как будто они и создавались под среду, на которой функцианируют [5].

PyQt5 – набор Python-библиотек для создания графического интерфейса на базе платформы Qt5. Библиотека Qt является одной из самых популярных и мощных библиотек GUI и реализована в виде большого набора Python-модулей. Эта библиотека имеет более 620 классов и 6000 функций и методов [3].

Одним из новых библиотек проектирования и функционирования графического интерфейса стал запущенный в 2018 году PySimpleGUI. Он позволяет создавать пользовательские графические интерфейсы в упрощенной форме, что делает его удобными при решении школьных задач. Определение окна упрощается за счет использования основных типов данных Python, понятных детям (списки и словари). Дальнейшее упрощение происходит за счет использования обработчика событий окна, передающий кортеж с именем нажатой клавиши и коллекцией введенных или измененных в окне параметров. При этом код не обязан иметь объектно-ориентированную архитектуру, что делает пакет PySimpleGUI доступным для более широкой аудитории. PySimpleGUI объединяет tkinter, Qt, WxPython и Remi, так что ребенок получает все те же пакеты (сложные для его понимания в своей основе), но он взаимодействует с ними более дружественным образом. PySimpleGUI в соответствующих версиях оборачивает части каждого из этих пакетов и упрощает их использование. Для установки базового пакета данной библиотеки необходимо выполнить терминале установку: pip install PySimpleGUI [2].

В PySimpleGUI есть два уровня поддержки работы с окнами: «High Level» и «Customized».

Первый уровень работает с уже предопределёнными окнами для выполнения частных задач. В данном режиме окно по сути это графический эквивалент оператора "print" или "input" (в зависимости от вида). Подходит для «приостановки» потока вашей программы, пока пользователь не сможет прочитать некоторые сообщения или ввести свою информацию.

Можно выделить ряд следующих вариантов окон:

- sg.popup('текст') # окно с кнопкой OK;
- sg.popup_yes_no('текст') # окно с кнопкой Yes и No;
- sg.popup_cancel('текст') # окно с кнопкой Cancel;
- sg.popup_ok_cancel('текст') # окно с кнопкой OK и Cancel;
- sg.popup_auto_close('текст', auto_close_duration=5) # автоматически закрывающееся окно;
- popup_scrolled #окно с прокруткой для вывода фрагмента текста;
- popup_get_text # получить одну строку текста;
- popup_get_file- #получить имя файла;
- popup_get_folder # получить имя папки.

Пример использования окон при создании учебной задачи «Мини-тест (анкета)». У пользователя запрашивается место на диске и имя файла для сохранения ответов на поставленные вопросы:

```
import PySimpleGUI as sg
answer = sg.popup_yes_no('Вы согласны пройти тест?', title="Тест")
if answer=='No' or answer==None:
   sg.popup('Работа программы окончена')
   exit()
my_folder = sg.popup_get_folder('Введите имя папки для сохранения текста')
if my_folder == None: exit()
my_fam = sg.popup_get_text('Введите вашу фамилию', 'Для создания файла' )
my_text1 = sg.popup_get_text('Раскройте смысл понятия "мобильное обучение"', 'Вопрос 1' )
my_text2 = sg.popup_get_text('Что вы понимаете под образовательной средой?', 'Вопрос 2' )
file_save = my_folder + '/'+my_fam+'.txt'
file_wr = open(file_save, 'w', encoding='utf-8')
file_wr.write(my_text1+'\n'+my_text2)
file_wr.close()
sg.popup('Ответ', 'Сохранен ваш ответ как', file_save)
```

Второй уровень поддержки окон «Customized» позволит ребенку самостоятельно и полностью спроектировать окно. Для создания визуальной формы необходимо определить макет окна, который представляет собой список строк. Каждая из строк является списком элементов, появляющихся в рамках строки (при этом эти элементы можно расположить и в колонки).

Главным строительным блоком является объект window() в котором необходимо указать имя окна, отображаемый макет и дополнительные параметры. Метод read() позволит запустить окно и ожидать любых событий.

Например команда: sg.Window(title="Макет", layout=[[sg.Text("Мое окно")]], margins=(50, 50)).read() позволит создать окно с заголовком, представленным макетом (layout) выводящем текстовую фразу и полями (margins).

Список базовых элементов окна (с алиасами-синонимами):

- *Text (T, Txt) - Отображение текста;*
- *Input (I, In, InputText) - Ввод данных;*
- *Button (B, Btn) - Кнопка без преопределённой надписи;*
- *OK - Кнопка с надписью OK;*
- *Cancel - Кнопка с надписью Cancel;*
- *Image (Im) - PNG или GIF картинка;*
- *FileBrowse -Выбор файла по имени;*
- *FolderBrowse - Выбор папки по имени.*

Пример разработанного интерфейса с функциональной частью для учебной задачи «Создание вычислителя: сложение и вычитание» (рис.1) с выведением результатов в отдельное окно:

```
import PySimpleGUI as sg
layout = [[sg.Text('Введите два числа')],
      [sg.Input(default_text=0, size=(10,1)), sg.Input(default_text=0, size=(10,1))],
      [sg.Button('+'), sg.Button('-'), sg.Button('Exit')]]
window = sg.Window('+-', layout)
while True:
   event, values = window.read()
   if event == '+':
      summa = float(values[0]) + float(values[1])
      sg.popup(summa, title='Ответ')
   if event == '-':
      summa = float(values[0]) - float(values[1])
      sg.popup(summa, title='Ответ')
   if  event == sg.WIN_CLOSED or event == 'Exit':
      break
window.close()
```

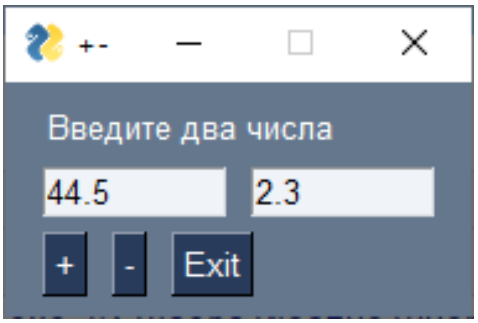

Рис. 1. Интерфейс вычислителя

Функция window.read() возвращает event – надпись на нажатой клавиши и список values с введенными значениями полей ввода. Благодаря введённой циклической конструкции окно не закроется после нажатия клавиш расчета, а после вывода результата будет снова ожидать нажатие клавиши. Для выхода из цикла определена реакция на нажатие кнопок закрытия окна и выхода.

Пример учебной задачи «Расчет индекса массы тела» (рис. 2) для решения которой школьник должен разработать интерфейс приложения, активировать его, привести код обработки нажатия клавиши для произведения соответствующего вычисления, при этом результат появляется не в отдельном окне (как в предыдущем примере) а в текущем обновляя содержимое одного из элементов окна:

```
import PySimpleGUI as sg
def calc_bmi(h, w):
   try:
      h, w = float(h)/100, float(w)
      imt = round(w / (h ** 2), 2)
      if imt < 18.5:
         s = "Недостаточная"
      elif 18.5 <= imt <= 25:
```

```
            s = "Нормальный"
        elif 25.0 < imt <= 30:
            s = "Избыток"
        else:
            s = "Ожирение"
    except (ValueError, ZeroDivisionError):
        return None
    else:
        return f'BMI: {imt}, {s}'

layer = [[sg.Text("Рост"), sg.Input(size = (15,1))],  [sg.Text ("Вес "), sg.Input(size = (15,1))],
[sg.Text('', key='imt', size=(20,2))],
         [sg.Button("Найти"), sg.Button ("Выход")]]

window = sg.Window("ИМТ", layer, size=(210, 140))

while True:
    event, value = window.read()
    if event == 'Найти':
        try:
            bmi = calc_bmi(value[0], value[1])
        except:
            bmi=None
        if bmi:
            window['imt'].update(bmi, text_color='black')
        else:
            window['imt'].update("Неверные данные", text_color='red')
    elif event == 'Выход' or event == sg.WIN_CLOSED:
        break

window.close()
```

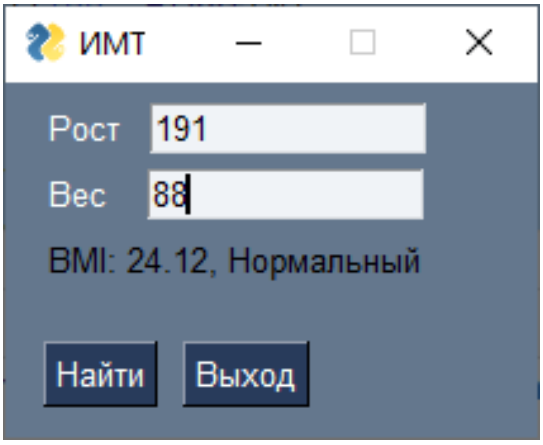

Рис. 2. Интерфейс программы расчёта индекса массы тела

Таким образом, представленные примеры демонстрируют простату использования возможностей библиотеки PySimpleGUI для построения графического интерфейса учебных задач, рассматриваемых на уроках информатики, при изучении языка программирования Python. Графический интерфейс помогает поддерживать интерес учащихся к образовательной

деятельности, программированию и успешно решать ряд учебных задач как в рамках предмета информатика основного курса, так и на занятиях дополнительного образования.

Список литературы

[1] About – Pygame wiki [Электронный ресурс] URL: https://www.pygame.org/wiki/about (дата обращения: 24.09.2022).
[2] PySimpleGUI [Электронный ресурс] URL: https://pypi.org/project/PySimpleGUI/ (дата обращения: 20.09.2022).
[3] Qt for Python. [Электронный ресурс] URL: https://doc.qt.io/qtforpython/ (дата обращения: 18.09.2022).
[4] The Ren'Py Visual Novel Engine. [Электронный ресурс] URL: https://www.renpy.org (дата обращения: 15.09.2022).
[5] Tkinter – Python interface to Tcl/Tk. [Электронный ресурс] URL: https://docs.python.org/3/library/tkinter.html (дата обращения: 20.08.2021).
[6] *Федорова Н.Е.* Структура, содержание и методические подходы к преподаванию языка программирования Python в школе // Современные информационные технологии и ИТ-образование. 2011. №7. [Электронный ресурс] URL: https://cyberleninka.ru/article/n/struktura-soderzhanie-i-metodicheskie-podhody-k-prepodavaniyu-yazyka-programmirovaniya-python-v-shkole (дата обращения: 24.09.2022).