

Создание графического интерфейса программы с помощью Kivy

Романов Даниил Алексеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Целью данной статьи является, создание графического интерфейса программы с помощью библиотеки Kivy языка программирования Python. Результатом исследования является шаблон графического интерфейса, обладающий некоторым функционалом, который в будущем можно использовать для создания полноценного приложения с функционалом. Шаблон создан с помощью языка программирования Python и его библиотеки Kivy в среде программирования PyCharm.

Ключевые слова: Python, Kivy, графический интерфейс, шаблон, PyCharm

Creating a graphical program interface using Kivy

Romanov Daniil Alekseevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to create a graphical program interface using the Kivy library of the Python programming language. The result of the study is a graphical interface template with some functionality that can be used in the future to create a full-fledged application with functionality. The template was created using the Python programming language and its Kivy library in the PyCharm programming environment.

Keywords: Python, Kivy, GUI, Template, PyCharm

1 Введение

1.1 Актуальность

На основе языка Python можно создавать не только web-сайты или проекты по машинному обучению, но также можно создавать вполне реальные программы с графическим интерфейсом, которые будут кроссплатформенными и с хорошей скоростью выполнения. Хотя для языка Python сфера полноценных проектов не первостепенно-важная, но всё же язык предоставляет отличные возможности для разработки полноценных программ под разные операционные системы. Kivy является довольно мощным модулем для этого, на его основе было построено много проектов под платформу Android. Если сравнивать Kivy с другими библиотеками Python по набору функций, то среди крупных конкурентов можно выделить

PyQT, а все прочие будут явно уступать по функционалу. Kivy имеет большой набор уже готовых проектов, которые можно использовать для построения программ.

1.2 Обзор исследований

В своей работе А. Cereto-Massagué рассматривали методы создания GUI на основе языка программирования Python [1]. МР Pinjani описывает в своей статье внедрение библиотеки Kivy при создании собственного приложения [2]. V.K. Sharma описывает использование Kivy для создания пользовательского интерфейса в проекте системы по распознаванию лиц [3].

1.3 Цель исследования

Цель исследования - рассмотрение библиотеки Kivy и создание на её основе графический интерфейс программы.

2 Материалы и методы

Для создания программы потребуется несколько вещей. Во-первых, это язык программирования Python [4], библиотека Kivy [5] и среда программирования PyCharm [6].

3 Результаты и обсуждение

В первую очередь создаём новый проект в PyCharm. Импортируем основной класс App, а также дополнительные на основе которых делаются различные виджеты. Класс Label нужен для создания текстовой надписи. Window нужен для установки различных глобальных настроек, таких как размер окна, его цвет и название. Button нужен для создания кнопки, а BoxLayout для работы с несколькими объектами. Так же импортируем метод randint для генерации случайного значения, оно понадобится при создании функционала кнопки (рис.1).

```
from kivy.app import App
from kivy.uix.label import Label
from kivy.uix.button import Button
from kivy.uix.boxlayout import BoxLayout
from kivy.core.window import Window
from random import randint
```

Рисунок 1 - Импортирование нужных классов

Используем класс Window для настройки размера, цвета в формате RGB и названия окна (рис.2).

```
Window.size = (300, 100)
Window.clearcolor = (255/255, 186/255, 3/255, 1)
Window.title = "Приложение"
```

Рисунок 2 - Настройка параметров окна

Создаём собственный класс и наследуем всё от класса App. Что бы обращаться к текстовой надписи из другого метода нужно указывать что она принадлежит всему классу, для этого создаём конструктор, в котором создаём саму надпись. Создаём метод btn_pressed, который будет срабатывать при нажатии кнопки, он будет устанавливать случайный цвет для текстовой надписи. В данном методе обращаемся к надписи и к её свойству color. Для установки случайного значения используем импортированный метод randint. Метод build нужен для реализации основных настроек. В нём создаём объект на основе класса BoxLayout, также создаём объект кнопки и добавляем надпись что будет на ней отображаться. На кнопку нужно повесить событие on_press, что будет срабатывать при нажатии на неё, в нём будет вызываться метод btn_pressed. Помещаем кнопку и текст надписи внутрь BoxLayout (рис.3).

```
class MyApp(App):
    def __init__(self):
        super().__init__()
        self.label = Label(text='Моя программа\nВсё работает!')

    def btn_pressed(self, *args):
        self.label.color = (randint(0, 255)/255, randint(0, 255)/255, randint(0, 255)/255, 1)

    def build(self):
        box = BoxLayout()
        btn = Button(text='Нажми на меня')
        btn.bind(on_press=self.btn_pressed)
        label = Label(text='Моя программа\nВсё работает!')
        box.add_widget(self.label)
        box.add_widget(btn)

    return box
```

Рисунок 3 - Создание класса для функционала программы

Осталось прописать вызов метода run для запуска проекта (рис.4).

```
if __name__ == "__main__":
    MyApp().run()
```

Рисунок 4 - Вызов метода run

Выводы

В данной работе была создана программа с графическим интерфейсом для пользователей, которая имеет небольшой функционал. Программу можно использовать как шаблон для создания других приложений с более продвинутым функционалом.

Библиографический список

1. Cereto-Massagué A. et al. DecoyFinder: an easy-to-use python GUI application for building target-specific decoy sets //Bioinformatics. 2012. Т. 28. №. 12. С. 1661-1662.
2. Pinjani M. P. et al. Adopting Python-Kivy Framework for Personal Safety Application //Journal of Android and IOS Applications and Testing. 2022. Т. 7. №. 1. С. 22-26.
3. Sharma V. K. Designing of face recognition system //2019 International Conference on Intelligent Computing and Control Systems (ICCS). IEEE, 2019. С. 459-461.
4. Python URL: <https://www.python.org/downloads>
5. Kivy URL: <https://pypi.org/project/Kivy>
6. PyCharm URL: <https://www.jetbrains.com/ru-ru/pycharm/download/?ysclid=14e394tnm7260579901#section=windows>