

Создание кроссплатформенного приложения с помощью фреймворка Kivy

Андрюенко Иван Сергеевич

*Приамурский государственный университет имени Шолом-Алейхема
Студент*

Аннотация

В данной статье описывается процесс разработки приложения «Калькулятор» на языке программирования Python. Для придания кроссплатформенности приложению использовался фреймворк Kivy. Созданное приложение можно запускать на разных платформах.

Ключевые слова: Python, Kivy, графический интерфейс, фреймворк.

Creating a cross-platform application using the Kivy framework

Andrienko Ivan Sergeevich

*Sholom-Aleichem Priamursky State University
Student*

Abstract

This article describes the process of developing the Calculator application in the Python programming language. The Kivy framework was used to make the application cross-platform. The created application can be run on different platforms.

Keywords: Python, Kivy, GUI, framework.

1 Введение

1.1 Актуальность

Количество мобильных устройств растет, люди все больше проводят времени с гаджетами. Такое положение вызвало серьезный спрос на специалистов в сфере мобильной разработки. В Python нет встроенных возможностей мобильной разработки, но есть пакеты, которые можно использовать для создания мобильных приложений, такие как Kivy. Благодаря этому, можно использовать понятный синтаксис Python для создания кроссплатформенных приложений.

1.2 Обзор исследований

В своей работе А.О. Кизянов, Д.В. Лучанинов продемонстрировали процесс создания игры 2048 с помощью фреймворка Kivy и языка программирования Python [1]. М.В. Болдырева рассмотрела относительно новый подход к разработке с помощью нового фреймворка kivy, работающего как библиотека python [2]. С.Н. Верзунов в своей работе

предложили архитектуру кроссплатформенного программного компонента трассоискателя, основанная на разделении кода, зависящего от целевой платформы от кода, который может без каких-либо изменений запускаться на любой целевой платформе. Кроссплатформенность достигнута с помощью таких тулкетов как Kivy и инструментальных средств сборки приложений CMake, Swig и Buildozer [3]. В своей работе Е.В. Изукаев, Д.А. Шуклин рассмотрели основные возможности инструментов для разработки мобильных приложений с помощью языка Python [4]. Е.А. Таравский, И.Ш. Сингатулло в своей научной статье рассмотрели подход к разработке с помощью фреймворка kivy, работающего как библиотека python [5].

1.3 Цель исследования

Цель исследования – Создать кроссплатформенное приложения «Калькулятор» на языке программирования Python с помощью фреймворка Kivy.

2 Материалы и методы

Для разработки приложения использовалась среда программирования PyCharm и библиотека Kivy.

3 Результаты и обсуждения

Kivy — это бесплатный фреймворк Python с открытым исходным кодом. Он поддерживает мультитач-события в дополнение к обычным вводам с клавиатуры и мыши. Проект распространяется на условиях лицензии MIT и может работать на Android, iOS, Linux, macOS и Windows. Когда приложение создается с помощью Kivy, создается естественный пользовательский интерфейс или GUI. Идея естественного пользовательского интерфейса заключается в том, что пользователь может легко научиться пользоваться программным обеспечением практически без каких-либо инструкций.

Для начала работы создаем проект и устанавливаем библиотеку kivy. Для этого вводим в терминал команду «pip install kivy».

Теперь пишем код программы. Для начала импортируем необходимые классы (рис. 1).

```
1 from kivy.app import App
2     from kivy.uix.boxlayout import BoxLayout
3     from kivy.uix.button import Button
4     from kivy.uix.textinput import TextInput
```

Рисунок 1 – Импорт классов

Далее создаем операторы для вычисления и верхнее окно куда будет выводиться ответ, доступное только для чтения. Также создаем массив, в котором будут храниться кнопки калькулятора (рис. 2).

```
6 class MainApp(App):
7     def build(self):
8         self.operators = ["/", "*", "+", "-"]
9         self.last_was_operator = None
10        self.last_button = None
11        main_layout = BoxLayout(orientation="vertical")
12        self.solution = TextInput(
13            multiline=False, readonly=True, halign="right", font_size=55
14        )
15        main_layout.add_widget(self.solution)
16        buttons = [
17            ["7", "8", "9", "/"],
18            ["4", "5", "6", "*"],
19            ["1", "2", "3", "-"],
20            [".", "0", "C", "+"],
21        ]
```

Рисунок 2 – Создание операторов, окна для ответа и массива

Теперь прописываем цикл. В нем будет строка, которая создает «BoxLayout» с горизонтальной ориентацией, а также еще один цикл, в котором будут созданы кнопки и привязаны обработчики событий к ним (рис. 3).

```
22         for row in buttons:
23             h_layout = BoxLayout()
24             for label in row:
25                 button = Button(
26                     text=label,
27                     pos_hint={"center_x": 0.5, "center_y": 0.5},
28                 )
29                 button.bind(on_press=self.on_button_press)
30                 h_layout.add_widget(button)
31             main_layout.add_widget(h_layout)
32
```

Рисунок 3 – Цикл для создания кнопок

Отдельно пропишем кнопку «=» и привяжем к нему обработчик события (рис. 4).

```
33     equals_button = Button(  
34         text="=", pos_hint={"center_x": 0.5, "center_y": 0.5}  
35     )  
36     equals_button.bind(on_press=self.on_solution)  
37     main_layout.add_widget(equals_button)  
38  
39     return main_layout
```

Рисунок 4 – Создание кнопки «=»

Теперь необходимо создать сами обработчики. Зададим действие, что при нажатии кнопки «C» очищалось поле ввода. Так же добавим проверку на наличие рядом стоящих операторов. Это не позволит пользователю написать два знака подряд, например: «5*-3». Создадим проверку против написания оператора в начале примера (рис. 5).

```
45     if button_text == "C":  
46         self.solution.text = ""  
47     else:  
48         if current and (  
49             self.last_was_operator and button_text in self.operators):  
50             return  
51         elif current == "" and button_text in self.operators:  
52             return  
53         else:  
54             new_text = current + button_text  
55             self.solution.text = new_text  
56             self.last_button = button_text  
57             self.last_was_operator = self.last_button in self.operators
```

Рисунок 5 – Создание обработчиков событий

Теперь пропишем сам расчет и получение данных. Для этого будет использована встроенная функция «eval». Eval() выполнит код и вернет результат (рис. 6).

```
59     def on_solution(self, instance):  
60         text = self.solution.text  
61         if text:  
62             solution = str(eval(self.solution.text))  
63             self.solution.text = solution  
64
```

Рисунок 6 – Расчет и получение данных

Осталось прописать вызов метода run для запуска проекта (рис.7).

```
66 ▶ if __name__ == "__main__":  
67     app = MainApp()  
68     app.run()
```

Рисунок 7 – Запуск проекта

Запускаем и тестируем созданное приложение (рис.8).



Рисунок 8 – Тестирование приложения

Выводы

В данной работе было разработана кроссплатформенное приложение «Калькулятор» на языке программирования Python с помощью фреймворка Kivy. Созданное приложение можно запускать на разных платформах.

Библиографический список

1. Кизянов А.О., Лучанинов Д.В. Разработка игры 2048 на фреймворке kivy. // Постулат. 2016. № 9 (11). С. 3.
2. Болдырева М.В. Разработка кроссплатформенных приложений с помощью фреймворка kivy. // Наука сегодня: факты, тенденции, прогнозы. Материалы международной научно-практической конференции. 2018. С.

11-12.

3. Верзунов С.Н. Разработка кроссплатформенного программного компонента трассоискателя. // Проблемы автоматизации и управления. 2020. № 1 (38). С. 50-59.
4. Изукаев Е.В., Шуклин Д.А. Сравнительный анализ разработки мобильных приложений на Python // Альманах научных работ молодых ученых Университета ИТМО. в 5 т. 2016. С. 229-230.
5. Таравский Е.А., Сингатуллов И.Ш. Использование kivy framework для создания мобильного приложения на языке программирования Python // Концепция развития и эффективного использования научного потенциала общества. Сборник статей Международной научно-практической конференции. В 2-х частях. 2020. С. 99-101.