

Направление подготовки: 13.04.02 Электроэнергетика и электротехника

Направленность (профиль): Электродвижение и электроснабжение наземных транспортных средств

Уровень образования: магистратура

Дисциплина: "Информационные и компьютерные технологии в электротехнике"

**Комаров В.Г. Лабораторная работа 11 21.04.2026
г. Методические указания**

Графический интерфейс среды ELTRO_SIM

Задание:

Создать графическое приложение «калькулятор»

Последовательность создания калькулятора:

- 1) импортировать kivy
- 2) импортировать kivyApp
- 3) импортировать GridLayout
- 4) импортировать конфигурацию (для настройки/регулировки размера окна)
- 5) указать минимальную версию kivy (необязательно)
- 6) создать класс макета: определить функцию калькулятора в нем, использовать try-catch для создания исключения при возникновении каких-либо ошибок в арифметических операциях.
- 7) создать класс приложения
- 8) создать файл .kv (имя файла должно быть таким же, как у класса приложения):
 - создать кнопки
 - добавить стиль к кнопкам
 - добавить функциональность кнопок
- 9) вывести на экран (return) макет/виджет/класс
- 10) запустить экземпляр класса.

Реализация приложения калькулятора:

Листинг файла `calculator.py`

```
# Программа-калькулятор
# импортируем модуль kivy
import kivy

# базовый класс создаваемого приложения наследуем от класса
# App
from kivy.app import App
```

```

# ограничим использование более ранних, чем указано версий
kivy.require('1.9.0')

# для создания нескольких кнопок и их упорядочивания
# используем метод GridLayout
from kivy.uix.gridlayout import GridLayout

# для задания размеров графического окна импортируем Config
from kivy.config import Config

# устанавливаем размеры окна
Config.set('graphics', 'resizable', 1)
## Config.set('graphics', 'width', '400')
## Config.set('graphics', 'height', '400')

# создаём Layout class
class CalcGridLayout(GridLayout):

    def calculate(self, calculation):
        if calculation:
            try:
                # Solve formula and display it in entry
                # which is pointed at by display
                self.display.text = str(eval(calculation))
            except Exception:
                self.display.text = "Error"

# Creating App class
class CalculatorApp(App):

    def build(self):
        return CalcGridLayout()

# creating object and running it
CalculatorApp().run()

```

Листинг файла `calculator.kv`

```

# Custom button
<CustButton@Button>:
    font_size: 32

# Define id so I can refer to the CalcGridLayout
# class functions
# Display points to the entry widget
<CalcGridLayout>:
    id: calculator
    display: entry
    rows: 6
    padding: 10
    spacing: 10

```

```

# Where input is displayed
BoxLayout:
    TextInput:
        id: entry
        font_size: 32
        multiline: False

# When buttons are pressed update the entry
BoxLayout:
    spacing: 10
    CustButton:
        text: "7"
        on_press: entry.text += self.text
    CustButton:
        text: "8"
        on_press: entry.text += self.text
    CustButton:
        text: "9"
        on_press: entry.text += self.text
    CustButton:
        text: "+"
        on_press: entry.text += self.text

BoxLayout:
    spacing: 10
    CustButton:
        text: "4"
        on_press: entry.text += self.text
    CustButton:
        text: "5"
        on_press: entry.text += self.text
    CustButton:
        text: "6"
        on_press: entry.text += self.text
    CustButton:
        text: "-"
        on_press: entry.text += self.text

BoxLayout:
    spacing: 10
    CustButton:
        text: "1"
        on_press: entry.text += self.text
    CustButton:
        text: "2"
        on_press: entry.text += self.text
    CustButton:
        text: "3"
        on_press: entry.text += self.text
    CustButton:
        text: "*"
        on_press: entry.text += self.text

```

```

# When equals is pressed pass text in the entry
# to the calculate function
BoxLayout:
    spacing: 10
    CustButton:
        text: "AC"
        on_press: entry.text = ""
    CustButton:
        text: "0"
        on_press: entry.text += self.text
    CustButton:
        text: "="
        on_press: calculator.calculate(entry.text)
    CustButton:
        text: "/"
        on_press: entry.text += self.text
BoxLayout:
    CustButton:
        font_size: 20
        text: "Простой калькулятор"
        on_press: entry.text = ""

```

Результат запуска программы:

