

Санкт-Петербургский Национальный  
Исследовательский Университет Информационных  
технологий, механики и оптики

**Лабораторная работа иб.  
GUI.**

Выполнил: Фоминцев  
Денис Русланович  
Группа № 3123  
Проверила: Казанова  
Полина Петровна

## Цель работы:

### Постановка задачи

Реализовать программу с помощью GUI, состоящую из двух частей. Первая часть – простой калькулятор. Вторая часть – калькулятор по подсчету периметра прямоугольника и его площади. Также необходимо реализовать возможность просмотра данной фигуры.

## Ход работы:

За основу для реализации GUI было принято решение взять библиотеку tkinter. Обе задачи были реализованы в одном окне, разделенном на две части (см. рисунок 1).

Левая – калькулятор; Правая – прямоугольник (см. рисунок 2 и 3).

```
172 if __name__ == '__main__':
173     root = Tk()
174     ex = Example()
175     db = GUI(root)
176
177     root['bg'] = '#f0f0f0'
178     root.geometry('885x550+200+200')
179     root.title('Calculator')
180     root.resizable(False, False)
181
182     root.mainloop()
```

Рисунок 1. Построение окна

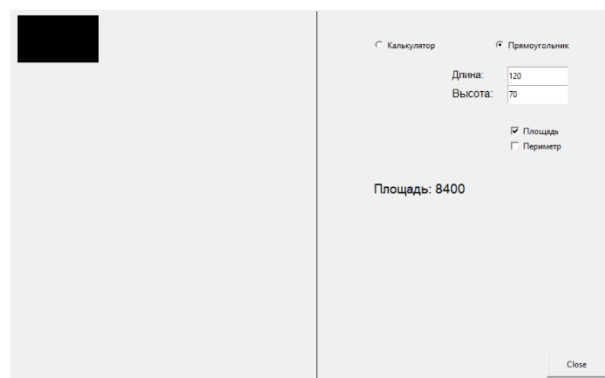


Рисунок 2. Пример с Площадью



Рисунок 3. Пример с Периметром

Выборность Площадь/Периметр реализована с помощью функции Checkbutton. Выборность калькулятор/прямоугольник реализована с помощью функции Radiobutton (см. рисунок 4).

```

155     self.square = Checkbutton(master, text="Площадь", command=show_square)
156     self.square.place(relx=0.825, rely=0.30)
157     self.perim = Checkbutton(master, text="Периметр", command=show_perim)
158     self.perim.place(relx=0.825, rely=0.34)
159
160
161     self.calc = Radiobutton(master, text="Калькулятор", value=1, command=calc_active)
162     self.calc.place(relx=0.6, rely=0.07)
163
164     self.rectangle = Radiobutton(master, text="Прямоугольник", value=2, command=rectangle_active)
165     self.rectangle.place(relx=0.8, rely=0.07)
166
167     self.close_button = Button(master, text="Close", command=master.quit)
168     self.close_button.place(relx=0.89, rely=0.91, relwidth=0.1, relheight=0.07)
169

```

Рисунок 4. Кнопки выборности

При переключении программы на режим калькулятора, выборность прямоугольника блокируется, предыдущие рисунки пропадают, данные длины и высоты тоже исчезают (см. рисунок 5). Также присутствует возможность вернуться обратно на режим прямоугольника и снова нарисовать новую фигуру, при этом калькулятор пропадет.

C	DEL	%	/
7	8	9	*
4	5	6	-
1	2	3	+
0	+/-	^2	=

Калькулятор

Длина:

Высота:

Периметр: 840

Прямоугольник

☐ Площадь

☒ Периметр

Close

Рисунок 5. Пример калькулятора

Реализация построения кнопок калькулятора показана на рисунке 6, логику работы данных кнопок см. на рисунке 7.

Также для режима прямоугольника присутствует валидация строковых символов, функция validate (см. рисунок 8).

```

10     def build(self):
11         self.formula = '0'
12         self.lbl = Label(text=self.formula, font=('Arial', 32, 'bold'), bg='#f0f0f0', foreground='#000')
13         self.lbl.place(x=10, y=50)
14
15
16         buttons = [
17             'C', 'DEL', '%', '/',
18             '7', '8', '9', '*',
19             '4', '5', '6', '-',
20             '1', '2', '3', '+',
21             '0', '+/-', '^2', '=']
22
23
24         x = 10
25         y = 140
26         for btn in buttons:
27             com = lambda f=btn: self.logicalc(f)
28             Button(text=btn, bg='#e9e9e9', font=('Arial', 16), command=com).place(x=x, y=y, width=90, height=78)
29             x += 92
30             if x > 300:
31                 x = 10
32                 y += 80

```

Рисунок 6. Кнопки калькулятора

```

34     def logicalc(self, operation):
35         if operation == 'C':
36             self.formula = ''
37         elif operation == 'DEL':
38             self.formula = self.formula[0:-1]
39         elif operation == '^2':
40             self.formula = str((eval(self.formula)) ** 2)
41         elif operation == '=':
42             self.formula = str(eval(self.formula))
43         elif operation == '+/-':
44             self.formula = str(-eval(self.formula))
45         else:
46             if self.formula == '0':
47                 self.formula = ''
48             self.formula += operation
49         self.update()
50
51     def update(self):
52         if self.formula == '':
53             self.formula = '0'
54         self.lbl.configure(text=self.formula)
55

```

Рисунок 7. Реализация

```

74     def validate(value, action):
75         if action == '1':
76             return value.isdigit()
77         return True
78
79
80     self.len = Label(master, text="Длина: ", font=('Arial', 11))
81     self.len.place(relx=0.73, rely=0.15)
82     self.var = StringVar()
83     self.arg_1_ent = Entry(master, validate="all", textvariable=self.var, width=10)
84     self.arg_1_ent['vcmd'] = (self.arg_1_ent.register(validate), '%P', '%d')
85     self.arg_1_ent.place(relx=0.825, rely=0.15, relwidth=0.1, relheight=0.05)

```

Рисунок 8. Пример валидации для длины прямоугольника

**Вывод:**

В данной работе реализовали программу, состоящую из двух частей:

- Простого калькулятора;
- Калькулятора по подсчету периметра прямоугольника, по указанным пользователем параметрам, и его площади. Также присутствует реализация рисовки данной фигуры.