

ПРИЛОЖЕНИЕ А

Решение заданий, представленных в методическом руководстве

1.1 Задание 1

```
1 import tkinter as tk
2
3 class App(tk.Tk):
4
5     def __init__(self):
6         super().__init__()
7         fields = ["Имя", "Фамилия", "Телефон", "Email"]
8         labels = [tk.Label(self, text=f) for f in fields]
9         entries = [tk.Entry(self) for _ in fields]
10        self.widgets = list(zip(labels, entries))
11        self.submit = tk.Button(self, text="Распечатать",
12                                command=self.print_info)
13
14        for i, (label, entry) in enumerate(self.widgets):
15            label.grid(row=i, column=0, padx=10, sticky=tk.W)
16            entry.grid(row=i, column=1, padx=10, pady=5)
17        self.submit.grid(row=len(fields), column=1, sticky=tk.E,
18                          padx=10, pady=10)
19
20    def print_info(self):
21        for label, entry in self.widgets:
22            print("{} = {}".format(label.cget("text"),
23                                   ↪ entry.get()))
24
25 if __name__ == "__main__":
26     app = App()
27     app.mainloop()
```

1.2 Задание 2

```
1 import tkinter as tk
2
```

```

3  window_main = tk.Tk(className='Tkinter - TutorialKart')
4  window_main.geometry("400x200")
5
6  frame_1 = tk.Frame(window_main, bg='#c4ffd2', width=200,
    ↪ height=50)
7  frame_1.pack()
8  frame_1.pack_propagate(0)
9
10 frame_2 = tk.Frame(window_main, bg='#ffffff', width=200,
    ↪ height=50)
11 frame_2.pack()
12 frame_2.pack_propagate(0)
13
14 #in frame_1
15 label_1 = tk.Label(frame_1, text='Name')
16 label_1.pack(side=tk.LEFT)
17 submit = tk.Entry(frame_1)
18 submit.pack(side=tk.RIGHT)
19
20 #in frame_2
21 button_reset = tk.Button(frame_2, text='Reset')
22 button_reset.pack(side=tk.LEFT)
23 button_submit = tk.Button(frame_2, text='Submit')
24 button_submit.pack(side=tk.RIGHT)
25
26 window_main.mainloop()

```

1.3 Задание 3

```

1  # Importing Tkinter module
2  from tkinter import *
3  from tkinter.ttk import *
4
5  # Creating master Tkinter window
6  master = Tk()
7  master.geometry("175x175")

```

```

8
9  # Tkinter string variable
10 # able to store any string value
11 v = StringVar(master, "1")
12
13 # Dictionary to create multiple buttons
14 values = {"RadioButton 1" : "1",
15           "RadioButton 2" : "2",
16           "RadioButton 3" : "3",
17           "RadioButton 4" : "4",
18           "RadioButton 5" : "5"}
19
20 # Loop is used to create multiple Radiobuttons
21 # rather than creating each button separately
22 for (text, value) in values.items():
23     Radiobutton(master, text = text, variable = v,
24                 value = value).pack(side = TOP, ipady = 5)
25
26 # Infinite loop can be terminated by
27 # keyboard or mouse interrupt
28 # or by any predefined function (destroy())
29 mainloop()

```

1.4 Задание 4

```

1 import tkinter as tk
2
3 def on_button_click():
4     label.config(text="Привет, Tkinter!")
5
6 # Создание главного окна
7 root = tk.Tk()
8 root.title("Простое приложение Tkinter")
9
10 # Создание виджетов

```

```

11 label = tk.Label(root, text="Добро пожаловать в Tkinter!",
    ↪ font=("Helvetica", 16))
12 button = tk.Button(root, text="Нажми меня",
    ↪ command=on_button_click)
13
14 # Размещение виджетов на окне
15 label.pack(pady=20)
16 button.pack(pady=10)
17
18 # Запуск главного цикла обработки событий
19 root.mainloop()

```

1.5 Задание 5

```

1 import tkinter as tk
2
3 def on_button_click(operation):
4     try:
5         num1 = float(entry_num1.get())
6         num2 = float(entry_num2.get())
7
8         if operation == "+":
9             result = num1 + num2
10        elif operation == "-":
11            result = num1 - num2
12        elif operation == "*":
13            result = num1 * num2
14        elif operation == "/":
15            if num2 != 0:
16                result = num1 / num2
17            else:
18                result = "Ошибка: деление на ноль"
19    except ValueError:
20        result = "Ошибка: введите числа"
21
22    label_result.config(text="Результат: " + str(result))

```

```

23
24 root = tk.Tk()
25 root.title("Простой калькулятор")
26
27 # Создание виджетов
28 entry_num1 = tk.Entry(root, width=10)
29 entry_num2 = tk.Entry(root, width=10)
30 button_add = tk.Button(root, text="+", command=lambda:
    ↪ on_button_click("+"))
31 button_subtract = tk.Button(root, text="-", command=lambda:
    ↪ on_button_click("-"))
32 button_multiply = tk.Button(root, text="*", command=lambda:
    ↪ on_button_click("*"))
33 button_divide = tk.Button(root, text="/", command=lambda:
    ↪ on_button_click("/"))
34 label_result = tk.Label(root, text="Результат: ")
35
36 # Размещение виджетов на окне
37 entry_num1.grid(row=0, column=0, padx=5, pady=5)
38 entry_num2.grid(row=0, column=1, padx=5, pady=5)
39 button_add.grid(row=1, column=0, padx=5, pady=5)
40 button_subtract.grid(row=1, column=1, padx=5, pady=5)
41 button_multiply.grid(row=2, column=0, padx=5, pady=5)
42 button_divide.grid(row=2, column=1, padx=5, pady=5)
43 label_result.grid(row=3, columnspan=2, padx=5, pady=5)
44
45 root.mainloop()

```

1.6 Задание 6

```

1 import tkinter as tk
2 from tkinter import messagebox
3
4 def add_task():
5     task = entry_task.get()
6     if task:

```

```

7         listbox_tasks.insert(tk.END, task)
8         entry_task.delete(0, tk.END)
9     else:
10         messagebox.showwarning("Пустая задача", "Пожалуйста,
            ↪ введите задачу.")
11
12 def delete_task():
13     try:
14         selected_index = listbox_tasks.curselection()[0]
15         listbox_tasks.delete(selected_index)
16     except IndexError:
17         messagebox.showwarning("Нет выбранной задачи", "Выберите
            ↪ задачу для удаления.")
18
19 root = tk.Tk()
20 root.title("Менеджер задач")
21
22 # Создание виджетов
23 entry_task = tk.Entry(root, width=30)
24 button_add = tk.Button(root, text="Добавить задачу",
    ↪ command=add_task)
25 button_delete = tk.Button(root, text="Удалить задачу",
    ↪ command=delete_task)
26 listbox_tasks = tk.Listbox(root, selectmode=tk.SINGLE, width=30)
27
28 # Размещение виджетов на окне
29 entry_task.pack(pady=10)
30 button_add.pack(pady=5)
31 button_delete.pack(pady=5)
32 listbox_tasks.pack()
33
34 root.mainloop()

```

1.7 Задание 7

```
1 import tkinter as tk
2 from tkinter import messagebox
3
4 def on_login():
5     username = entry_username.get()
6     password = entry_password.get()
7
8     # Проверка логина и пароля
9     if username == "admin" and password == "password":
10         messagebox.showinfo("Успешный вход", "Добро пожаловать,
            ↪ {}".format(username))
11     else:
12         messagebox.showerror("Ошибка входа", "Неверный логин или
            ↪ пароль")
13
14 # Создание главного окна
15 root = tk.Tk()
16 root.title("Менеджер паролей")
17
18 # Создание виджетов
19 label_username = tk.Label(root, text="Логин:")
20 label_password = tk.Label(root, text="Пароль:")
21 entry_username = tk.Entry(root)
22 entry_password = tk.Entry(root, show="*") # show="*" скрывает
            ↪ вводимые символы
23 button_login = tk.Button(root, text="Войти", command=on_login)
24
25 # Размещение виджетов на окне
26 label_username.grid(row=0, column=0, padx=10, pady=5, sticky=tk.E)
27 label_password.grid(row=1, column=0, padx=10, pady=5, sticky=tk.E)
28 entry_username.grid(row=0, column=1, padx=10, pady=5)
29 entry_password.grid(row=1, column=1, padx=10, pady=5)
30 button_login.grid(row=2, columnspan=2, pady=10)
```

```
31
32 # Запуск главного цикла обработки событий
33 root.mainloop()
```

1.8 Задание 8

```
1 import tkinter as tk
2 from tkinter import messagebox
3 import random
4
5 class GuessNumberGame:
6     def __init__(self, master):
7         self.master = master
8         self.master.title("Угадай число")
9
10        self.secret_number = random.randint(1, 100)
11        self.attempts = 0
12
13        self.label_info = tk.Label(master, text="Угадай число от 1
14        ↪ до 100:")
15
16        self.label_info.pack(pady=10)
17
18        self.entry_guess = tk.Entry(master)
19        self.entry_guess.pack(pady=10)
20
21        self.button_guess = tk.Button(master, text="Попробовать",
22        ↪ command=self.check_guess)
23
24        self.button_guess.pack()
25
26
27        def check_guess(self):
28            try:
29                guess = int(self.entry_guess.get())
30                self.attempts += 1
31
32                if guess == self.secret_number:
```



```

28         messagebox.showinfo("Поздравляем!", f"Вы угадали
        ↳ число за {self.attempts} попыток!")
29         self.master.destroy()
30     elif guess < self.secret_number:
31         messagebox.showinfo("Подсказка", "Число больше!")
32     else:
33         messagebox.showinfo("Подсказка", "Число меньше!")
34
35         self.entry_guess.delete(0, tk.END)
36     except ValueError:
37         messagebox.showwarning("Ошибка", "Пожалуйста, введите
        ↳ целое число.")
38
39     # Создание главного окна
40     root = tk.Tk()
41
42     # Создание объекта игры
43     game = GuessNumberGame(root)
44
45     # Запуск главного цикла обработки событий
46     root.mainloop()

```

1.9 Задание 9

```

1  import tkinter as tk
2  from tkinter import messagebox
3
4  class CountdownTimer:
5      def __init__(self, master):
6          self.master = master
7          self.master.title("Таймер обратного отсчета")
8
9          self.time_remaining = tk.StringVar()
10         self.time_remaining.set("00:00")
11

```

```

12         self.label_timer = tk.Label(master,
    ↪     textvariable=self.time_remaining, font=("Helvetica",
    ↪     24))
13     self.label_timer.pack(pady=20)
14
15     self.entry_seconds = tk.Entry(master, width=5)
16     self.entry_seconds.insert(0, "60")
17     self.entry_seconds.pack(pady=10)
18
19     self.button_start = tk.Button(master, text="Старт",
    ↪     command=self.start_timer)
20     self.button_start.pack()
21
22     def start_timer(self):
23         try:
24             seconds = int(self.entry_seconds.get())
25             if seconds <= 0:
26                 messagebox.showwarning("Ошибка", "Пожалуйста,
    ↪     введите положительное количество секунд.")
27             else:
28                 self.run_timer(seconds)
29         except ValueError:
30             messagebox.showwarning("Ошибка", "Пожалуйста, введите
    ↪     корректное количество секунд (целое число).")
31
32     def run_timer(self, seconds):
33         self.button_start.config(state=tk.DISABLED)
34         self.time_remaining.set(self.format_time(seconds))
35         self.master.after(1000, self.update_timer, seconds)
36
37     def update_timer(self, seconds):
38         if seconds > 0:
39             seconds -= 1
40             self.time_remaining.set(self.format_time(seconds))

```

```

41         self.master.after(1000, self.update_timer, seconds)
42     else:
43         self.button_start.config(state=tk.NORMAL)
44         messagebox.showinfo("Таймер завершен", "Время
        ↪ вышло!")
45
46     def format_time(self, seconds):
47         minutes, seconds = divmod(seconds, 60)
48         return "{:02d} : {:02d}".format(minutes, seconds)
49
50     # Создание главного окна
51     root = tk.Tk()
52
53     # Создание объекта таймера
54     timer = CountdownTimer(root)
55
56     # Запуск главного цикла обработки событий
57     root.mainloop()

```

1.10 Задание 10

```

1  import tkinter as tk
2  from tkinter import messagebox
3  import random
4
5  class RockPaperScissorsGame:
6      def __init__(self, master):
7          self.master = master
8          self.master.title("Камень, ножницы, бумага")
9
10         self.choices = ["Камень", "Ножницы", "Бумага"]
11
12         self.label_instruction = tk.Label(master, text="Выберите
        ↪ ваш вариант:")
13         self.label_instruction.pack(pady=10)
14

```

```

15     self.button_rock = tk.Button(master, text="Камень",
    ↪     command=lambda: self.play_round("Камень"))
16     self.button_scissors = tk.Button(master, text="Ножницы",
    ↪     command=lambda: self.play_round("Ножницы"))
17     self.button_paper = tk.Button(master, text="Бумага",
    ↪     command=lambda: self.play_round("Бумага"))
18
19     self.button_rock.pack(pady=5)
20     self.button_scissors.pack(pady=5)
21     self.button_paper.pack(pady=5)
22
23     def play_round(self, player_choice):
24         computer_choice = random.choice(self.choices)
25         result = self.determine_winner(player_choice,
    ↪         computer_choice)
26
27         messagebox.showinfo("Результат", f "Вы выбрали:
    ↪     {player_choice} \n Компьютер выбрал:
    ↪     {computer_choice} \n {result} ")
28
29     def determine_winner(self, player_choice, computer_choice):
30         if player_choice == computer_choice:
31             return "Ничья!"
32         elif (
33             (player_choice == "Камень" and computer_choice ==
    ↪             "Ножницы") or
34             (player_choice == "Ножницы" and computer_choice ==
    ↪             "Бумага") or
35             (player_choice == "Бумага" and computer_choice ==
    ↪             "Камень")
36         ):
37             return "Вы выиграли!"
38         else:
39             return "Компьютер выиграл."

```

```
40
41 # Создание главного окна
42 root = tk.Tk()
43
44 # Создание объекта игры
45 game = RockPaperScissorsGame(root)
46
47 # Запуск главного цикла обработки событий
48 root.mainloop()
```

ПРИЛОЖЕНИЕ Б

Тест

2.1 Вопросы

1. Что такое Tkinter?
 - a) Язык программирования
 - b) Библиотека для создания графических интерфейсов в Python
 - c) Фреймворк для работы с базами данных
2. Какой виджет используется для создания кнопки в Tkinter?
 - a) Label
 - b) Button
 - c) Entry
3. Что такое главное окно в Tkinter?
 - a) Окно, которое всегда находится поверх других окон
 - b) Окно, в котором отображаются основные элементы интерфейса
 - c) Окно, которое закрывается последним
4. Какой метод используется для запуска главного цикла обработки событий в Tkinter?
 - a) run()
 - b) main()
 - c) mainloop()
5. Какой виджет используется для отображения текста в Tkinter?
 - a) Text
 - b) Label
 - c) Entry
6. Как добавить изображение в графический интерфейс с использованием Tkinter?
 - a) add_image()
 - b) insert_image()
 - c) PhotoImage()
7. Как создать радиокнопки в Tkinter?
 - a) RadioButton
 - b) CheckButton
 - c) Radio
8. Как изменить цвет фона окна в Tkinter?

- a) `root.bgcolor("color")`
 - b) `root.configure(bg="color")`
 - c) `root.set_bgcolor("color")`
9. Что такое сетка (grid) в Tkinter?
- a) Виджет для отображения таблицы
 - b) Система для размещения виджетов в виде сетки
 - c) Метод для рисования линий в окне
10. Как связать функцию с событием щелчка кнопки в Tkinter?
- a) `bind()`
 - b) `link()`
 - c) `connect()`
11. Как вывести диалоговое окно с предупреждением в Tkinter?
- a) `show_info()`
 - b) `message_box.showwarning()`
 - c) `popup.alert()`
12. Как установить размер окна в Tkinter?
- a) `set_size()`
 - b) `size()`
 - c) `geometry()`

2.2 Ответы

- 1. b) Библиотека для создания графических интерфейсов в Python
- 2. b) Button
- 3. b) Окно, в котором отображаются основные элементы интерфейса
- 4. c) `mainloop()`
- 5. b) Label
- 6. c) `PhotoImage()`
- 7. b) CheckButton
- 8. b) `root.configure(bg="color")`
- 9. b) Система для размещения виджетов в виде сетки
- 10. a) `bind()`
- 11. b) `message_box.showwarning()`
- 12. c) `geometry()`