

**Студент группы ИС-27 Белоглазов Максим**  
**Практическое занятие №17**

**Тема:** составление программ с использованием GUI Tkinter в IDE PyCharm Community, изучение возможностей модуля OS.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community, изучить возможности модуля OS.

**Постановка задачи:**

**1)** В соответствии с номером варианта перейти по ссылке на прототип.

Реализовать его в IDE PyCharm Community с применением пакета tk. Получить интерфейс максимально приближенный к оригиналу (см. таблицу 1).

**1**

В файле form.html создайте с помощью таблицы форму регистрации.

Анкета Web-разработчика	
Регистрационное имя	<input type="text"/>
Пароль	<input type="password"/> <input type="password"/> : подтвердите пароль
Ваша специализация	Web-мастер <input type="button" value="v"/>
Пол	М <input checked="" type="radio"/> Ж <input type="radio"/>
Ваши навыки	<input type="checkbox"/> знание HTML и CSS <input type="checkbox"/> знание Perl <input type="checkbox"/> знание ASP <input type="checkbox"/> знание Adobe Photoshop <input type="checkbox"/> знание JAVA <input type="checkbox"/> знание JavaScript <input type="checkbox"/> знание Flash
Дополнительные сведения о себе	<input type="text"/>
<input type="button" value="зарегистрировать"/> <input type="button" value="очистить форму"/>	

2) Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ №№ 2 – 9.

3) Задание предполагает, что у студента есть проект с практическими работами (№№ 2-13),

оформленный согласно требованиям. Все задания выполняются с использованием модуля

OS:

✓ перейдите в каталог PZ11. Выведите список всех файлов в этом каталоге.

Имена

вложенных подкаталогов выводить не нужно.

✓ перейти в корень проекта, создать папку с именем test. В ней создать еще одну папку

test1. В папку test переместить два файла из ПЗ6, а в папку test1 - один файл из ПЗ7.

Файл из ПЗ7 переименовать в test.txt. Вывести в консоль информацию о размере файлов в папке test.

✓ перейти в папку с PZ11, найти там файл с самым коротким именем, имя вывести в

консоль. Использовать функцию `basename()` (`os.path.basename()`).

✓ перейти в любую папку где есть отчет в формате .pdf и «запустите» файл в привязанной к нему программе. Использовать функцию `os.startfile()`.

✓ удалить файл test.txt.

### 1) Текст программы:

```
1 import tkinter as tk
2 from tkinter import ttk
3
4 1 usage
5 def submit():
6     print("Регистрация прошла успешно")
7     1 usage
8 def clear():
9     entry_username.delete(first=0, tk.END)
10    entry_password.delete(first=0, tk.END)
11    entry_confirm_password.delete(first=0, tk.END)
12    combo_specialization.set('')
13    gender.set(None)
14    for var in skills_vars:
15        var.set(0)
16    text_additional_info.delete(index='1.0', tk.END)
17
18 # Создание главного окна
19 root = tk.Tk()
20 root.title("Анкета Web-разработчика")
21
22 # Фреймы для левой и правой стороны с разными цветами фона
23 left_frame = tk.Frame(root, bg='lightgray')
24 right_frame = tk.Frame(root, bg='gray')
25
26 left_frame.grid(row=0, column=0, rowspan=13, sticky='nsew')
27 right_frame.grid(row=0, column=1, columnspan=3, rowspan=13, sticky='nsew')
```

```

28 # Добавление разделительной линии
29 separator1 = ttk.Separator(root, orient='vertical')
30 separator1.grid(row=0, column=1, rowspan=12, padx=5)
31
32 # Метка и поле для имени пользователя
33 tk.Label(root, text="Регистрационное имя", bg='lightgray').grid(row=0, column=0, sticky='e')
34 entry_username = tk.Entry(root)
35 entry_username.grid(row=0, column=1, columnspan=3, sticky='we', padx=(0, 5))
36
37 # Добавление разделительной линии
38 separator1 = ttk.Separator(root, orient='horizontal')
39 separator1.grid(row=1, column=0, columnspan=4, sticky='ew', pady=5)
40
41 # Метка и поля для пароля
42 tk.Label(root, text="Пароль", bg='lightgray').grid(row=2, column=0, sticky='e')
43 entry_password = tk.Entry(root, show='*')
44 entry_password.grid(row=2, column=1, sticky='we')
45
46 tk.Label(root, text="Подтвердите пароль", bg='gray').grid(row=3, column=2, sticky='w')
47 entry_confirm_password = tk.Entry(root, show='*')
48 entry_confirm_password.grid(row=3, column=1, sticky='we')
49
50 separator2 = ttk.Separator(root, orient='horizontal')
51 separator2.grid(row=4, column=0, columnspan=4, sticky='ew', pady=5)
52
53 # Метка и выпадающий список для специализации
54 tk.Label(root, text="Ваша специализация", bg='lightgray').grid(row=5, column=0, sticky='e')
55 combo_specialization = ttk.Combobox(root, values=["Web-мастер", "Программист", "Дизайнер"])
56 combo_specialization.grid(row=5, column=1, sticky='we')
57
58 separator2 = ttk.Separator(root, orient='horizontal')
59 separator2.grid(row=6, column=0, columnspan=4, sticky='ew', pady=5)
60
61 # Метка и радиокнопки для выбора пола
62 tk.Label(root, text="Пол", bg='lightgray').grid(row=7, column=0, sticky='e')
63 gender = tk.StringVar(value="М")
64 tk.Radiobutton(root, text="М", variable=gender, value='М', bg='gray').grid(row=7, column=1, sticky='w')
65 tk.Radiobutton(root, text="Ж", variable=gender, value='Ж', bg='gray').grid(row=7, column=2, sticky='w')
66
67 separator2 = ttk.Separator(root, orient='horizontal')
68 separator2.grid(row=8, column=0, columnspan=4, sticky='ew', pady=5)
69
70 # Метка и флажки для навыков
71 tk.Label(root, text="Ваши навыки", bg='lightgray').grid(row=9, column=0, sticky='e')
72 skills_frame = tk.Frame(root)
73 skills_frame.grid(row=9, column=1, columnspan=3, sticky='w')
74
75 skills = ["Знание HTML и CSS", "Знание Perl", "Знание ASP", "Знание Adobe Photoshop", "Знание JAVA", "Знание JavaScript", "Знание Flash"]
76 skills_vars = [tk.IntVar() for _ in skills]
77
78 for skill, var in zip(skills, skills_vars):
79     tk.Checkbutton(skills_frame, text=skill, variable=var, bg='gray').pack(anchor='w')
80
81 separator2 = ttk.Separator(root, orient='horizontal')
82 separator2.grid(row=10, column=0, columnspan=4, sticky='ew', pady=5)
83
84 # Метка и текстовое поле для дополнительной информации
85 tk.Label(root, text="Дополнительные сведения о себе", bg='lightgray').grid(row=11, column=0, sticky='e')
86 text_additional_info = tk.Text(root, width=40, height=5)
87 text_additional_info.grid(row=11, column=1, columnspan=3, sticky='we', padx=(0, 5))

```

```

88
89 separator2 = ttk.Separator(root, orient='horizontal')
90 separator2.grid(row=12, column=0, colspan=4, sticky='ew', pady=5)
91
92 # Кнопки регистрации и очистки формы
93 button_frame = tk.Frame(root)
94 button_frame.grid(row=13, column=1)
95
96 tk.Button(button_frame, text="зарегистрироваться", command=submit).pack(side='left', padx=5, pady=(0, 5))
97 tk.Button(button_frame, text="очистить форму", command=clear).pack(side='left', padx=5, pady=(0, 5))
98
99 # Настройка расширяемости столбцов
100 root.grid_columnconfigure(index=1, weight=1)
101 root.grid_columnconfigure(index=2, weight=1)
102 root.grid_columnconfigure(index=3, weight=1)
103
104 root.mainloop()

```

Вывод

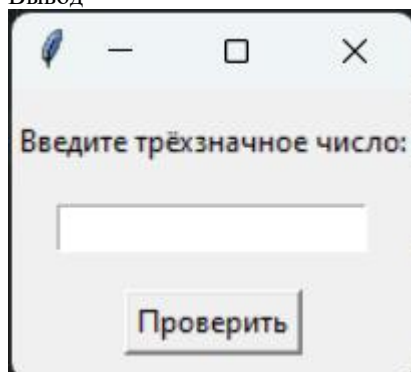
2)

```

1 import tkinter as tk
2 from tkinter import messagebox
3
4 1 usage
5 def main_number(number: int) -> None:
6     last_digit = number % 10
7     middle_digit = (number % 100) // 10
8     result = f"Последняя цифра (единица): {last_digit}\nСредняя цифра (десятки): {middle_digit}"
9     messagebox.showinfo( title: "Результат", result)
10
11 1 usage
12 def check_number() -> None:
13     number_str = entry.get()
14     try:
15         number = int(number_str)
16         if len(str(number)) == 3:
17             main_number(number=number)
18         else:
19             messagebox.showerror( title: "Ошибка", message: "Введенное вами число не трехзначное")
20     except ValueError:
21         messagebox.showerror( title: "Ошибка", message: "Требуется ввести число")
22
23 # Создаем основное окно
24 root = tk.Tk()
25 root.title("Анализ трёхзначного числа")
26
27 # Создаем метку и поле ввода
28 label = tk.Label(root, text="Введите трёхзначное число:")
29 label.pack(pady=10)
30
31 entry = tk.Entry(root)
32 entry.pack(pady=5)
33
34 # Создаем кнопку для запуска проверки
35 button = tk.Button(root, text="Проверить", command=check_number)
36 button.pack(pady=10)
37
38 # Запускаем главный цикл приложения
39 root.mainloop()

```

Вывод



3)



```

1 import os
2
3 # перейдите в каталог PZ_11. Выведите список всех файлов в этом каталоге. Имена вложенных подкаталогов выводить не нужно.
4 os.chdir("../PZ_11")
5 files_in_pz11 = [f for f in os.listdir() if os.path.isfile(f)]
6 print("Список файлов в каталоге PZ_11:", files_in_pz11)

```

```

1 import os
2 import shutil
3
4 # перейти в корень проекта, создать папку с именем test. В ней создать еще одну папку
5 # test1. В папку test переместить два файла из P36, а в папку test1 - один файл из P37.
6 # Файл из P37 переименовать в PZ_7.1.py. Вывести в консоль информацию о размере
7 # файлов в папке test.
8
9 os.chdir('../..')
10 os.makedirs(name='test/test1', exist_ok=True)
11
12 file_from_pz6_1 = 'PZ_6/PZ_6.1.py'
13 file_from_pz6_2 = 'PZ_6/PZ_6.2.py'
14 file_from_pz7 = 'PZ_7/PZ_7.1.py'
15
16 shutil.move(file_from_pz6_1, dst='test/')
17 shutil.move(file_from_pz6_2, dst='test/')
18 shutil.move(file_from_pz7, dst='test/test1/PZ_7.1.py')
19
20 files_in_test = [f for f in os.listdir('test') if os.path.isfile(os.path.join('test', f))]
21 for file in files_in_test:
22     file_size = os.path.getsize(os.path.join('test', file))
23     print(f"Размер файла {file} в папке test: {file_size} байт")
24

```

```

1 import os
2
3 # Найти файл с самым коротким именем в папке PZ_11
4 os.chdir('/PZ_11')
5 shortest_name_file = min((f for f in os.listdir() if os.path.isfile(f)), key=len)
6 print("Файл с самым коротким именем:", os.path.basename(shortest_name_file))

```

```

1 import os
2
3 # перейти в любую папку где есть отчет в формате .pdf и «запустите» файл в
4 # привязанной к нему программе. Использовать функцию os.startfile().
5 pdf_report_path = 'D:/ProgramPython/Practic Two/Reports/reports pz_16.pdf'
6 os.startfile(pdf_report_path)

```

```
1  import os
2
3  # удалить файл test.txt.
4  test_file_path = '../test/test1/test.txt'
5
6  os.remove(test_file_path)
7  print("Файл test.txt удален")
```

#### Вывод

В процессе выполнения практического занятия №17 я закрепил понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community, изучить возможности модуля OS.