

Практическая работа №12.

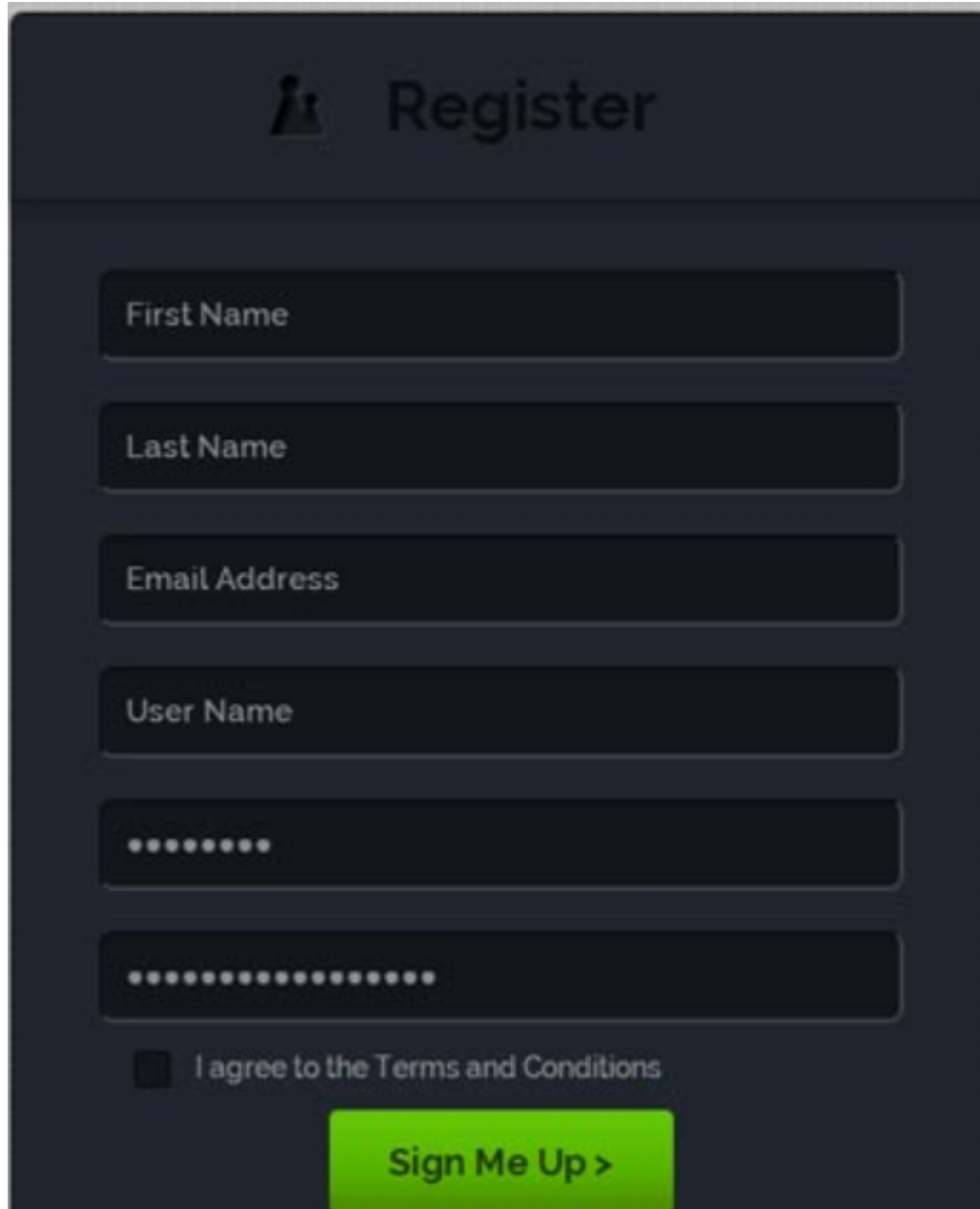
Тема: составление программ с использованием GUI Tkinter в IDE PyCharm Community.

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

Постановка задачи:

Задание 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).



Задание 2. Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ №№ 3 – 8.

Тип алгоритма: циклический и ветвление.

Текст программы:

Задача 1:

```
# Реализовать прототип в IDE PyCharm Community с применением пакета tk.  
# Получить интерфейс максимально приближенный к оригиналу.
```

```
from tkinter import *
```

```

from tkinter import ttk

My_list = ['First Name', 'Last Name', 'Email Address', 'User Name', 'Password',
'Verify Password']
y = 180

root = Tk() # Инициализируем корневое окно приложения.
root.title("Register")
root.geometry('500x700')

label = Label(text="Register", width=100, height=50, bg="black")
label.place(x=0, y=0)
Label(bg='#20242d', text='Register', fg='black', font='comforta
20').place(x=10, y=10,

width=480, height=90)
Label(bg='#20242d').place(x=10, y=101,
                        width=480, height=570)

for i in My_list: # Создание подписей полей ввода.
    Label(text=i, bg='#20242d', fg='white', font=('comforta', 10)).place(x=50,
y=y)
    y += 70

# Добавление панелей ввода.

Entry(bg='#13161b', fg='#444444', font=('comforta', 15)).place(x=50, y=130,
                        width=400,
height=50)
Entry(bg='#13161b', fg='#444444', font=('comforta', 15)).place(x=50, y=200,
                        width=400,
height=50)
Entry(bg='#13161b', fg='#444444', font=('comforta', 15)).place(x=50, y=270,
                        width=400,
height=50)
Entry(bg='#13161b', fg='#444444', font=('comforta', 15)).place(x=50, y=340,
                        width=400,
height=50)
Entry(bg='#13161b', fg='#444444', font=('comforta', 15)).place(x=50, y=410,
                        width=400,
height=50)
Entry(bg='#13161b', fg='#444444', font=('comforta', 15)).place(x=50, y=480,
                        width=400,
height=50)
# Добавление поля соглашения (в процессе работы с данным виджетом не оказалось
возможным заменить фон)

Var1 = IntVar()
chk = ttk.Checkbutton(variable=Var1, onvalue=1, offvalue=0)
chk.place(x=50, y=570, anchor=CENTER)

labell = Label(root, text="I agree to the Terms and Conditions", bg='#20242d',
fg='#444444',
                font=('Arial Bold', 12)).place(x=60, y=555, width=300, height=30)

```

```
# Добавление кнопки отправки данных.

button = Button(root, text='Sign me UP-->', bg='#8FBC8F', fg='#006400',
font=('Arial Bold', 13))
button.place(x=90, y=610, width=300, height=50)

root.resizable(width=False, height=False)    # Блокировка масштабирования окна.
root.mainloop()
```

Задача 2:

```
# Даны положительные числа А, В, С. На прямоугольнике размера А х В размещено
максимально
# возможное количество квадратов со стороной С (без наложений). Найти
количество квадратов,
# размещенных на прямоугольнике. Операции умножения и деления не использовать.
# Создать приложение к данной задаче в Tkinter.
```

```
from tkinter import *
from tkinter import ttk
```

```
# Функция-обработчик нажатия кнопки: вывод будет происходить путём изменения
скрытого текстового поля.
```

```
def square_count():
    a = side_a.get()
    # Обработка исключений
    # (невозможно остановить выполнение функции сразу при вводе неверных
    параметров, решение будет показано далее)
    try:
        a = float(a)
    except ValueError:
        Fin_label.configure(text='Первая сторона введена неверно!', fg='red')

    b = side_b.get()
    try:
        b = float(b)
    except ValueError:
        Fin_label.configure(text='Вторая сторона введена неверно!', fg='red')

    c = val_c.get()
    try:
        c = float(c)
    except ValueError:
        Fin_label.configure(text='Сторона квадрата введена неверно!', fg='red')

    # В случае невозможности присваивания float типа данным, происходит
    повторная проверка данных и обработка ошибок).

    if type(a) != float or type(b) != float or type(c) != float:
        Fin_label.configure(text='Сторона квадрата введена неверно!', fg='red')
    elif a <= 0 or b <= 0 or c <= 0:
```

```

        Fin_label.configure(text='Введён как минимум один отрицательный
показатель', fg='red')
    else:
        t = c # присваивание первоначальных значений переменным
        p = c
        i = 0
        while t <= a: # выполнение условия задачи
            while p <= b:
                p += c
                i += 1
            p = c
            t += c
        Fin_label.configure(text=f'Количество квадратов: {i}')

# Инициализация корневого окна приложения.

root = Tk()
root.title('Счётчик квадратов')
root.geometry('450x500')

main = Label(text='Счётчик квадратов', font=('Arial', 20)).grid(columnspan=2,
row=0)

side_a = Entry(master=main, width=10) # Добавление полей для ввода
информации.
side_a.grid(row=1, column=0, pady=5, padx=5)
lab_sidea = Label(text='Введите первую сторону внешней прямоугольной области',
font=('Arial', 10)).grid(row=1, column=1,

pady=5)
side_b = Entry(master=main, width=10)
side_b.grid(row=2, column=0, padx=5)
lab_sideb = Label(text='Введите вторую сторону внешней прямоугольной области',
font=('Arial', 10)).grid(row=2, column=1,

pady=5)
val_c = Entry(master=main, width=10)
val_c.grid(row=3, column=0, padx=5)
lab_valc = Label(text='Введите сторону внутренних квадратов', font=('Arial',
10)).grid(row=3, column=1, pady=5,

sticky=W)
# Добавление кнопки, к которой привязан обработчик.
button = Button(text='Асепт', font=('Arial', 12), command=square_count)
button.grid(row=4, columnspan=2, sticky=W + E, padx=5)

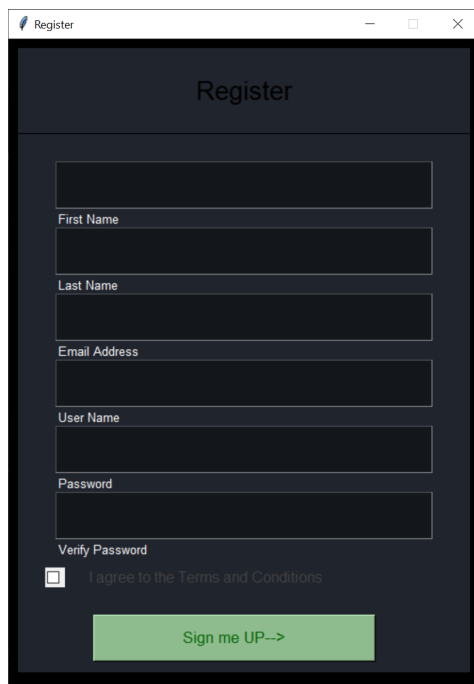
Fin_label = Label(text='', font=('Arial', 12)) # Добавление изменяемого
текстового поля.
Fin_label.grid(row=5, columnspan=2)

root.mainloop()

```

Протокол работы программы:

Задача 1:



Register

First Name

Last Name

Email Address

User Name

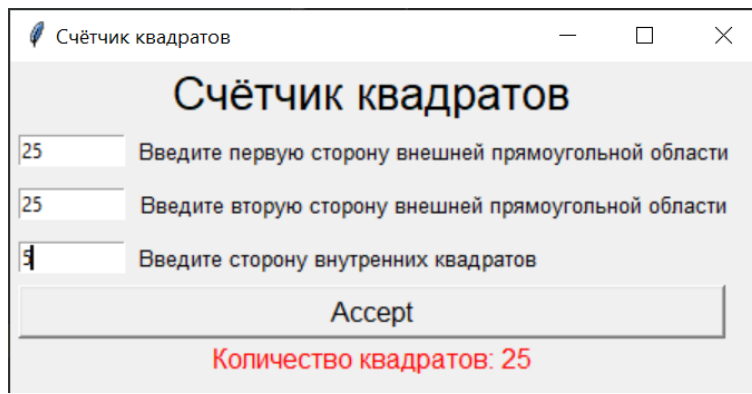
Password

Verify Password

☐ I agree to the Terms and Conditions

Sign me UP-->

Задача 2:



Счётчик квадратов

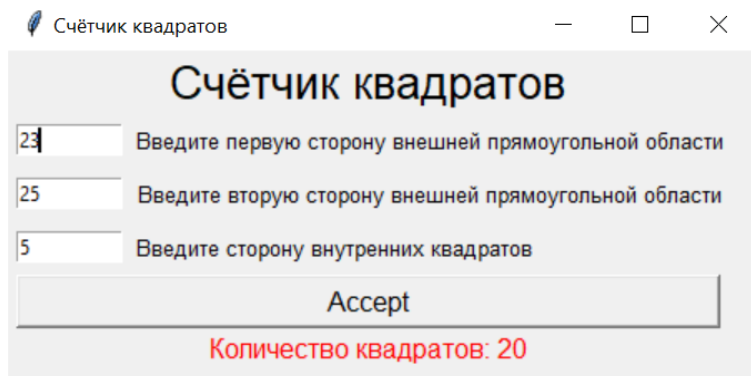
25 Введите первую сторону внешней прямоугольной области

25 Введите вторую сторону внешней прямоугольной области

5 Введите сторону внутренних квадратов

Асепт

Количество квадратов: 25



Счётчик квадратов

23 Введите первую сторону внешней прямоугольной области

25 Введите вторую сторону внешней прямоугольной области

5 Введите сторону внутренних квадратов

Асепт

Количество квадратов: 20

Счётчик квадратов

Счётчик квадратов

Введите первую сторону внешней прямоугольной области

Введите вторую сторону внешней прямоугольной области

Введите сторону внутренних квадратов

Accept

Сторона квадрата введена неверно!

Счётчик квадратов

Счётчик квадратов

Введите первую сторону внешней прямоугольной области

Введите вторую сторону внешней прямоугольной области

Введите сторону внутренних квадратов

Accept

Введён как минимум один отрицательный показатель

Вывод: в процессе выполнения практического занятия закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community.