

Практическое занятие № 12

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

Постановка задачи: 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 .

Текст программы:

```
from tkinter import *

t = Tk()
t.title('PZ12-1')
t.geometry('514x750')
prep = Frame(t, bg='white', bd=10)
prep.pack(pady=10)
p1 = Frame(prepare, bg='#afc346', width=480, height=800, bd=20)
p1.pack()
'''
s1l = Label(p1, text="Step 1: Your details", bg='#afc346',
font=("times new roman bold", 18)).pack()

preps1 = Frame(p1, bg='white', bd=3)
preps1.pack(pady=2)
ps1 = Frame(preps1, bg='#c9d583', width=440, height=44, bd=1)
ps1.pack()

preps2 = Frame(p1, bg='white', bd=3)
preps2.pack(pady=2)
ps2 = Frame(preps2, bg='#c9d583', width=440, height=44, bd=1)
ps2.pack()

preps3 = Frame(p1, bg='white', bd=3)
preps3.pack(pady=2)
ps3 = Frame(preps3, bg='#c9d583', width=440, height=44, bd=1)
ps3.pack()

s2l = Label(p1, text="Step 2: Delivery address", bg='#afc346',
font=("times new roman", 18)).pack()

preps4 = Frame(p1, bg='white', bd=3)
preps4.pack(pady=2)
ps4 = Frame(preps4, bg='#c9d583', width=440, height=44, bd=1)
ps4.pack()

preps5 = Frame(p1, bg='white', bd=3)
```

```
preps5.pack(pady=2)
ps5 = Frame(preps5, bg='#c9d583', width=440, height=44, bd=1)
ps5.pack()
```

```
preps6 = Frame(p1, bg='white', bd=3)
preps6.pack(pady=2)
ps6 = Frame(preps6, bg='#c9d583', width=440, height=44, bd=1)
ps6.pack()
```

```
s3l = Label(p1, text="Step 3: Card details", bg='#afc346',
font=("times new roman", 18)).pack()
```

```
preps7 = Frame(p1, bg='white', bd=3)
preps7.pack(pady=2)
ps7 = Frame(preps7, bg='#c9d583', width=440, height=44, bd=1)
ps7.pack()
```

```
preps8 = Frame(p1, bg='white', bd=3)
preps8.pack(pady=2)
ps8 = Frame(preps8, bg='#c9d583', width=440, height=44, bd=1)
ps8.pack()
```

```
preps9 = Frame(p1, bg='white', bd=3)
preps9.pack(pady=2)
ps9 = Frame(preps9, bg='#c9d583', width=440, height=44, bd=1)
ps9.pack()
```

```
preps10 = Frame(p1, bg='white', bd=3)
preps10.pack(pady=2)
ps10 = Frame(preps10, bg='#c9d583', width=440, height=44, bd=1)
ps10.pack()
```

```
nl = Label(ps1, text="Name", bg='#c9d583', font=("arial italic",
8))
nl.grid(row=0, column=1, pady=5, padx=18)
ne = Entry(ps1)
ne.grid(row=0, column=2, pady=10, padx=20)
```

```
eml = Label(ps2, text="Email", bg='#c9d583', font=("arial italic",
8)).grid(row=0, column=1, pady=5, padx=19)
eme = Entry(ps2).grid(row=0, column=2, pady=5, padx=20)
```

```
phl = Label(ps3, text="Phone", bg='#c9d583', font=("arial italic",
8)).grid(row=0, column=1, pady=5, padx=20)
phe = Entry(ps3).grid(row=0, column=2, pady=5, padx=16)
```

```
addl = Label(ps4, text="Adress", bg='#c9d583', font=("arial
italic", 8)).grid(row=0, column=1, pady=5, padx=16)
```

```
adde = Text(ps4, width=18, height=5).grid(row=0, column=2, pady=5,
padx=16)
```

```
pcl = Label(ps5, text="Post code", bg='#c9d583', font=("arial
italic", 8)).grid(row=0, column=1, pady=5, padx=13)
pce = Entry(ps5).grid(row=0, column=2, pady=5, padx=14)
```

```
col = Label(ps6, text="Country", bg='#c9d583', font=("arial
italic", 8)).grid(row=0, column=1, pady=5, padx=16)
coe = Entry(ps6).grid(row=0, column=2, pady=5, padx=16)
```

```
ctl = Label(ps7, text="Card type", bg='#c9d583', font=("arial
italic", 8)).grid(row=0, column=1, pady=5)
```

```
var = IntVar()
```

```
vsl = Label(ps7, text="Mastercard", bg='#c9d583', font=("arial
italic", 8)).grid(row=1, column=1, pady=5, padx=4)
vsr = Radiobutton(ps7, bg='#c9d583', variable=var,
value=1).grid(row=1, column=2, pady=5, padx=4)
```

```
amexl = Label(ps7, text="AmEx", bg='#c9d583', font=("arial
italic", 8)).grid(row=1, column=3, pady=5, padx=4)
amexr = Radiobutton(ps7, bg='#c9d583', variable=var,
value=2).grid(row=1, column=4, pady=5, padx=4)
```

```
mscrcl = Label(ps7, text="VISA", bg='#c9d583', font=("arial
italic", 8)).grid(row=1, column=5, pady=5, padx=3)
mscrd = Radiobutton(ps7, bg='#c9d583', variable=var,
value=3).grid(row=1, column=6, pady=5, padx=3)
```

```
cnl = Label(ps8, text="Card number", bg='#c9d583', font=("arial
italic", 8)).grid(row=0, column=1, pady=5, padx=11)
cne = Entry(ps8).grid(row=0, column=2, pady=5, padx=10)
```

```
secl = Label(ps9, text="Security code", bg='#c9d583', font=("arial
italic", 8)).grid(row=0, column=1, pady=5, padx=9)
sece = Entry(ps9).grid(row=0, column=2, pady=5, padx=10)
```

```
cardnl = Label(ps10, text="Name on card", bg='#c9d583',
font=("arial italic", 8)).grid(row=0, column=1, pady=5, padx=10)
cardne = Entry(ps10).grid(row=0, column=2, pady=5, padx=10)
```

```
buy = Button(p1, text='buy it', width=10)
buy.pack(pady=10)
'''
t.mainloop()
```

Протокол работы программы:

The screenshot shows a web application window titled 'PZ12-1'. The form is divided into three steps:

- Step 1: Your details**
 - Name:
 - Email:
 - Phone:
- Step 2: Delivery address**
 - Address:
 - Post code:
 - Country:
- Step 3: Card details**
 - Card type: ☐ Mastercard ☐ AmEx ☒ VISA
 - Card number:
 - Security code:
 - Name on card:
 -

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

Текст программы:

```
from tkinter import *

# Разработать программу с применением пакета tk, взяв в качестве
# условия одну
# любую задачу из ПЗ №№ 3 – 8.
# пз-5.2
# Описать функцию DigitCountSum(K, C, S), находящую количество C
# цифр
# целого положительного числа K, а также их сумму S (K – входной,
# C и S –
# выходные параметры целого типа). С помощью этой функции найти
# количество
# и сумму цифр для каждого из пяти данных целых чисел.
```

```

def digit_count_sum(): # functions names must be in lowercase
    global e, r

    v1 = 't'
    try:
        v1 = int(e.get()) # ввод
    except ValueError: # обработка исключения
        a = 0 # заглушка, требовавшаяся для модификации
        if type(v1) == int and v1 > 0: # проверка на инт + модификация
            для отбивания отрицательных
            k = v1
        else:
            r.config(text='введите корректное значение\n')
            return

    q = len(str(k)) # фиксируем длину числа, которая будет
    выступать счётчиком
    s = 0 # объявление суммы
    while q:
        q -= 1 # счётчик
        s += int(str(k)[q]) # прибавляю к сумме цифру
        r.config(text='длина числа: {}, сумма цифр:
        {}'.format(len(str(k)), s)) # возвращаем длину числа и сумму цифр

t = Tk()
t.title('PZ12-2')
t.geometry('400x150')
d = Label(t, text='Введите число').pack() # инструкция
e = Entry(t, width=25) # ввод
e.pack()
r = Label(t, text='') # для результата
w = Button(t, text='нажмите для результата',
command=digit_count_sum)
w.pack()
r.pack()
t.mainloop()

```

Протокол работы программы:

PZ12-2

Введите число

49

нажмите для результата

длина числа: 2, сумма цифр: 13

PZ12-2

Введите число

заварка

нажмите для результата

введите корректное значение