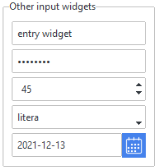
# Отчёт о проделанной работе КейсЛаб Разработчик

Наумов Александр

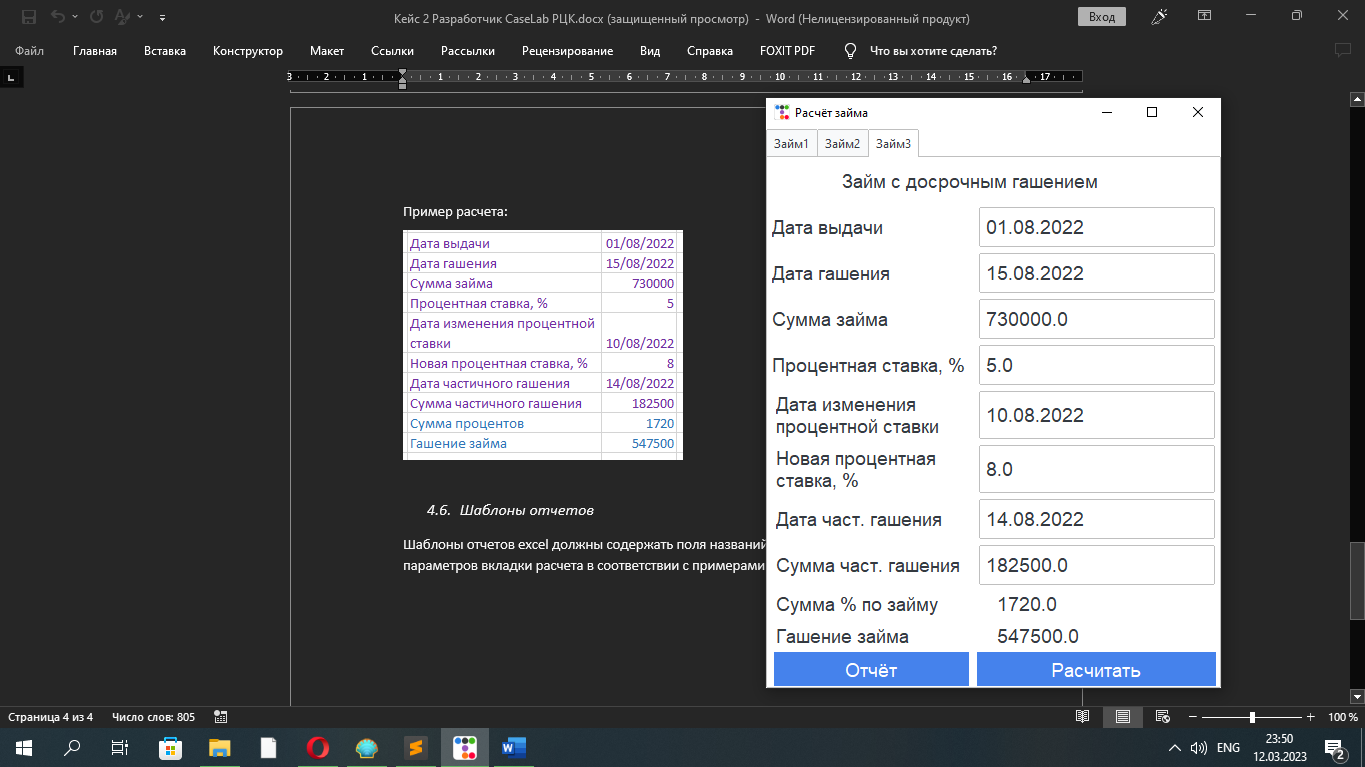
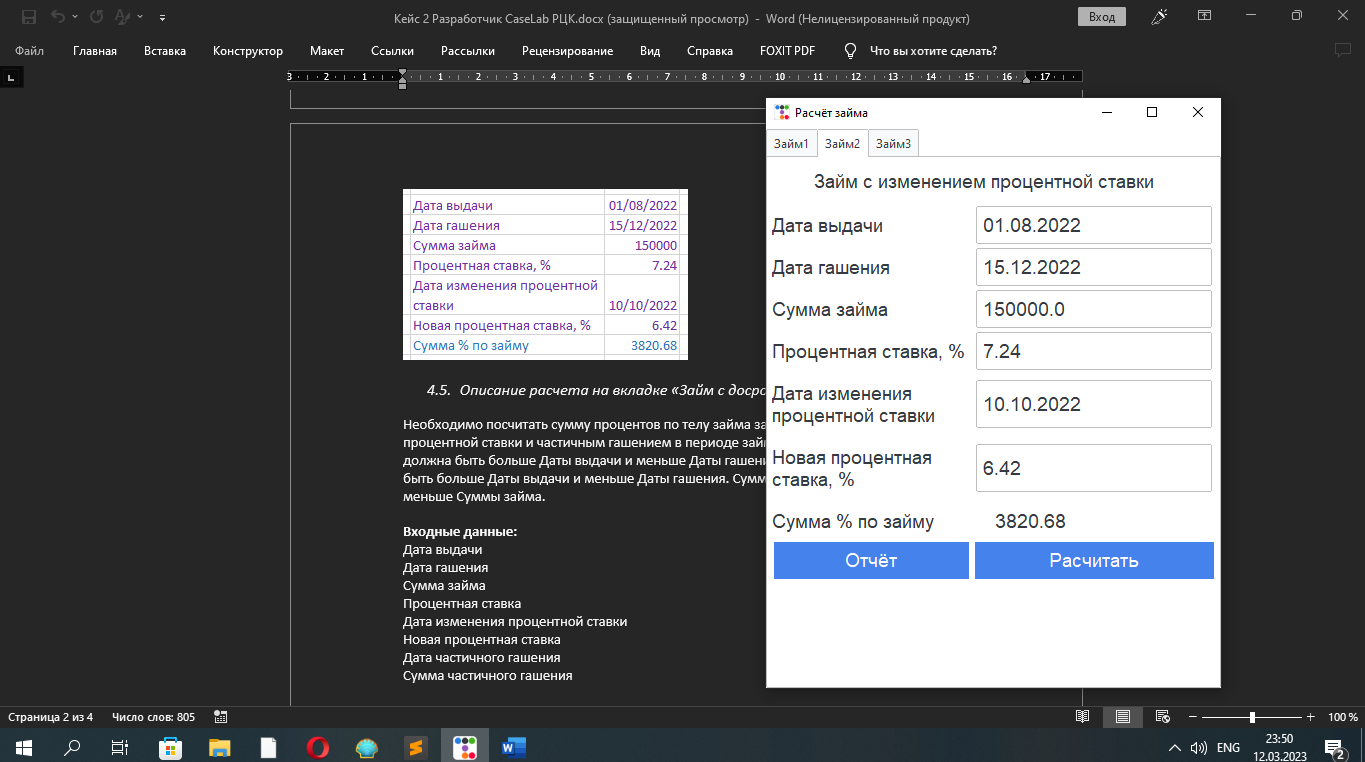
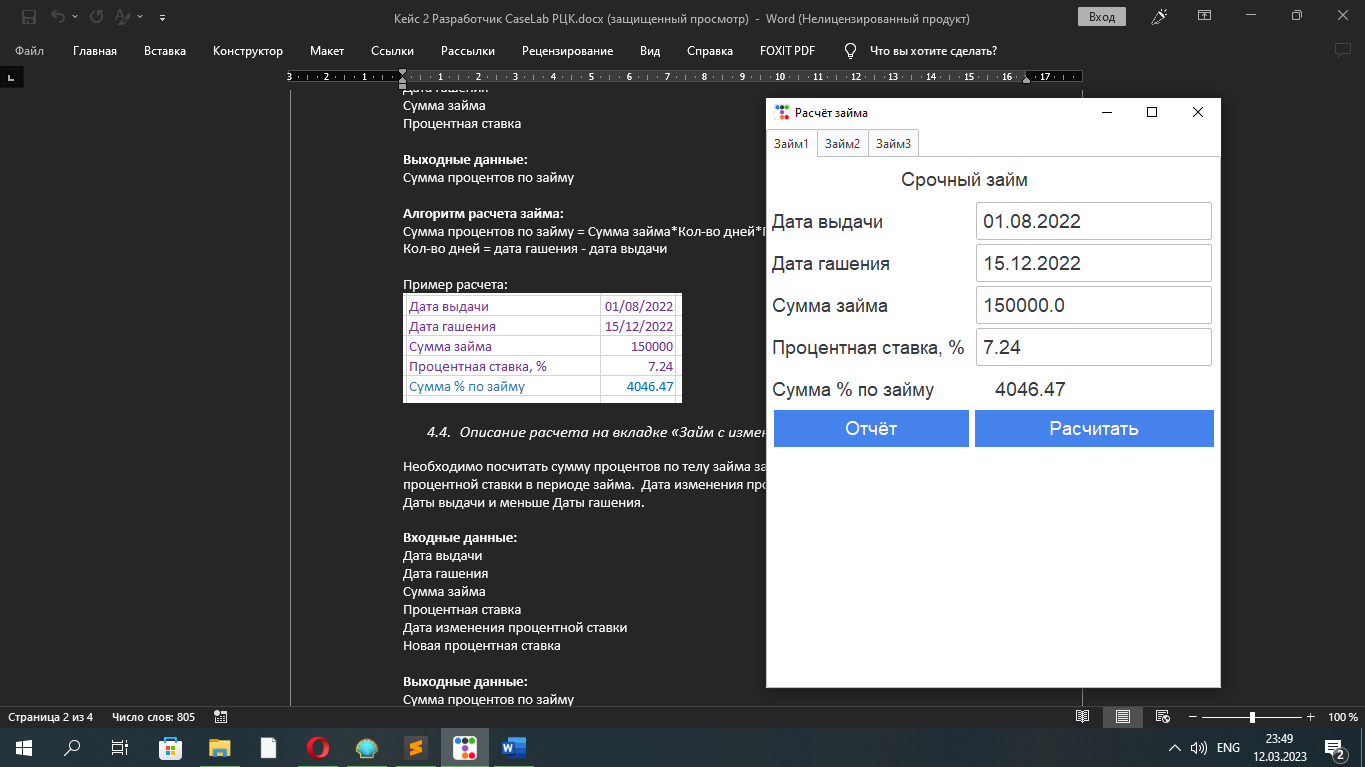
Возникшие проблемы и ход работы:

1. Проблема выбора языка не стояла, так как лучше всего я знаю Python.
2. Оказалось что Python с версии 3.9 прекратил поддерживать использование на windows 7, пришлось перенести программу в итоге на версию 3.8
3. Выбирал, между GUI библиотеками для Python PyQt5 и tkinter, выбор пал на tkinter так как по ней больше материала и она была мне ранее знакома.
4. Tkinter в плане дизайна выглядит до исторически, по этому я поискал библиотеки улучшающие внешне tkinter, из customtkinter и ttkbootstrap я выбрал второе, по способу использования тот же tkinter, но более красивый.
5. Была идея использовать разнообразные виджеты:

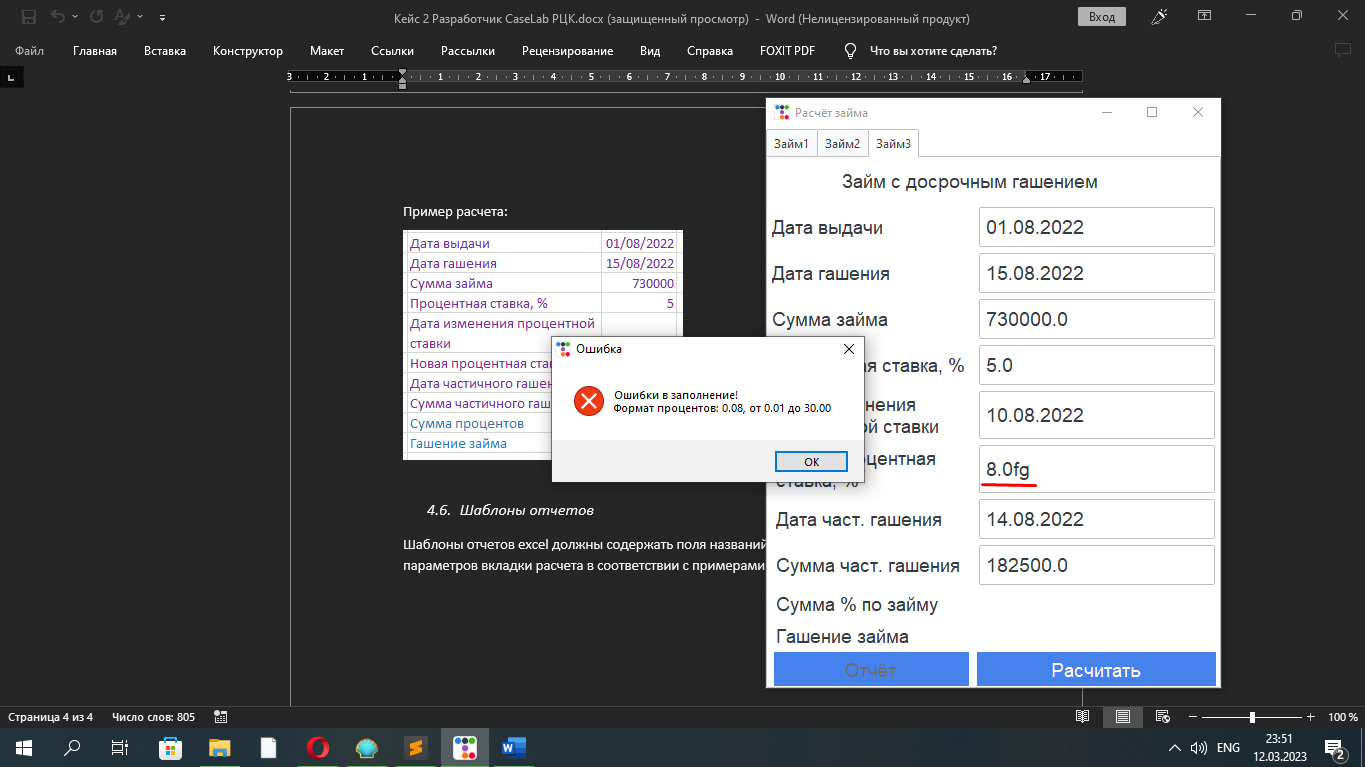


Но их центровка, подгонка по размерам оказались слишком затратными по времени, и я решил использовать простые поля для ввода, и не уверен, что я смог бы использовать интересные виджеты красиво.

1. Для отсеивания не верно введённых чисел (проценты, суммы), я использовал регулярные выражения (библиотека re), для дат я использовал специальный тип объектов datetime (библиотека datetime) и если введённые данные вызывают ошибку (не являются датой) то об этом сообщается. (Уже сейчас, я бы сделал по другому, вообще запретил бы вводить не корректные значения, в том числе для дат)
2. Для упаковки в exe воспользовался библиотекой auto-py-to-exe.
3. Для сохранения excel файла я использовал библиотеку pandas.



При попытке рассчитать займ с некорректными данными, появляется всплывающее окно с указанием типа(ов) ошибки.



# Код прилоения

from tkinter import Button as ButtonTk

import ttkbootstrap as ttk

from ttkbootstrap.constants import \*

from datetime import datetime

from re import match as re\_match

import tkinter.messagebox as mb

from pandas import DataFrame

# фокус для 1ого фрейма

def focus1(event):

widget = root.focus\_get()

if str(type(widget)) == "<class 'tkinter.ttk.Entry'>":

cumma\_1otv["text"] = " "

ot\_ex(btn\_ex1, cumma\_1otv)

def focus2(event):

widget = root.focus\_get()

if str(type(widget)) == "<class 'tkinter.ttk.Entry'>":

cumma\_2otv["text"] = " "

ot\_ex(btn\_ex2, cumma\_2otv)

def focus3(event):

focus1(event)

focus2(event)

widget = root.focus\_get()

if str(type(widget)) == "<class 'tkinter.ttk.Entry'>":

cumma\_3otv["text"] = " "

ghz\_3otv['text'] = " "

ot\_ex(btn\_ex3, cumma\_3otv)

# проверка ответа на заполненность, для разблокировки кнопки отчёта

def ot\_ex(btn\_ex, otv\_label):

if otv\_label["text"] == " ":

btn\_ex['state'] = ttk.DISABLED

else:

btn\_ex['state'] = ttk.NORMAL

def Create\_excel\_fail(dictionary, label):

df = DataFrame(dictionary)

str\_path = "./" + label + ".xlsx"

df.to\_excel(str\_path, index=False)

def Create\_label(frame, txt):

label = ttk.Label(frame, text=txt, width=17, font=("Arial", 14), borderwidth=0, relief="ridge")

return label

def Create\_entry(frame):

entry = ttk.Entry(frame, font=("Arial", 14))

return entry

def click\_button(date\_str\_arr, proc\_str\_arr, num\_str\_arr):

# проверка дат

try:

bul\_data = True

date\_arr = []

for i in date\_str\_arr:

date1 = datetime.strptime(i, '%d.%m.%Y').date()

if date1.year > 2100 or date1.year < 2000:

bul\_data = False

break

else:

date\_arr.append(date1)

if len(date\_arr) == 2:

if date\_arr[0] > date\_arr[-1] :

bul\_data = False

elif len(date\_arr) < 2 or len(date\_arr) > 4:

bul\_data = False

else:

if date\_arr[0] > date\_arr[-1] or date\_arr[1]<date\_arr[0] or date\_arr[1]>date\_arr[-1] or date\_arr[-2]<date\_arr[0] or date\_arr[-2]>date\_arr[-1]:

bul\_data = False

except ValueError as err:

bul\_data = False

global main\_date\_arr

main\_date\_arr = date\_arr

#проверка процентов

try:

bul\_proc = True

proc\_date = []

for i in proc\_str\_arr:

if re\_match("^\d{1,2}[.]\d{1,2}$", i) is None:

bul\_proc = False

if float(i) > 30.00 or float(i)< 0.01:

bul\_proc = False

else:

proc\_date.append(float(i))

except ValueError:

bul\_proc = False

global main\_proc\_arr

main\_proc\_arr = proc\_date

#проверка чисел

try:

bul\_num = True

num\_arr = []

for i in num\_str\_arr:

if re\_match("^\d{1,}[.]\d{1,2}$", i) is None:

bul\_num = False

if float(i) > 1000000.00 or float(i)< 0.99:

bul\_num = False

else:

num\_arr.append(float(i))

except ValueError:

bul\_num = False

global main\_num\_arr

main\_num\_arr = num\_arr

# проверка наличия ошибок

err\_str = "Ошибки в заполнение!"

if bul\_data==True and bul\_proc==True and bul\_num==True:

return True

else:

if bul\_data == False:

err\_str += "\nФормат дат: ДД.ММ.ГГГГ, от 01.01.2000 до 31.12.2099 "

if bul\_proc == False:

err\_str += "\nФормат процентов: 0.08, от 0.01 до 30.00 "

if bul\_num == False:

err\_str += "\nФормат чисел: 123.00, от 1.00 до 1000000.00 "

return err\_str

root = ttk.Window()

root.title("Расчёт займа")

root.geometry("455x560+400+100")

root.minsize(455, 560)

root.maxsize(455, 560)

# создаем набор вкладок

notebook = ttk.Notebook(bootstyle="light")

notebook.pack(expand=True, fill=BOTH)

# создаем пару фреймвов

frame1 = ttk.Frame(notebook)

frame2 = ttk.Frame(notebook)

frame3 = ttk.Frame(notebook)

frame1.pack(fill=BOTH, expand=True)

frame2.pack(fill=BOTH, expand=True)

frame3.pack(fill=BOTH, expand=True)

# добавляем фреймы в качестве вкладок

notebook.add(frame1, text="Займ1")

notebook.add(frame2, text="Займ2")

notebook.add(frame3, text="Займ3")

#======================frame1==========================

label\_1 = Create\_label(frame1, " Срочный займ").grid(row=0, column=0, columnspan=2, ipadx=6, ipady=6, padx=4, pady=4)

date\_give\_1 = Create\_label(frame1, "Дата выдачи").grid(row=1, column=0, ipadx=6, ipady=6, padx=4, pady=4)

date\_take\_1 = Create\_label(frame1, "Дата гашения").grid(row=2, column=0, ipadx=6, ipady=6, padx=4, pady=4)

num\_1 = Create\_label(frame1, "Сумма займа").grid(row=3, column=0, ipadx=6, ipady=6, padx=4, pady=4)

proc\_1 = Create\_label(frame1, "Процентная ставка, %").grid(row=4, column=0, ipadx=6, ipady=6, padx=4, pady=4)

cumma\_1 = Create\_label(frame1, "Сумма % по займу").grid(row=5, column=0, ipadx=6, ipady=6, padx=4, pady=4)

cumma\_1otv = Create\_label(frame1, " ")

cumma\_1otv.grid(row=5, column=1, ipadx=6, ipady=6, padx=4, pady=4)

date\_give\_1e = Create\_entry(frame1)

date\_give\_1e.grid(row=1, column=1, ipadx=1, ipady=1)

date\_take\_1e = Create\_entry(frame1)

date\_take\_1e.grid(row=2, column=1, ipadx=1, ipady=1)

num\_1e = Create\_entry(frame1)

num\_1e.grid(row=3, column=1, ipadx=1, ipady=1)

proc\_1e = Create\_entry(frame1)

proc\_1e.grid(row=4, column=1, ipadx=1, ipady=1)

def Cl1():

root.focus\_force()

main\_date\_arr = [date\_give\_1e.get(), date\_take\_1e.get()]

main\_proc\_arr = [proc\_1e.get()]

main\_num\_arr = [num\_1e.get()]

btn1 = click\_button(main\_date\_arr, main\_proc\_arr, main\_num\_arr)

if btn1==True:

#логика приложения

main\_date\_arr[0]=datetime.strptime(main\_date\_arr[0], '%d.%m.%Y')

main\_date\_arr[-1]=datetime.strptime(main\_date\_arr[-1], '%d.%m.%Y')

date = str(main\_date\_arr[-1]-main\_date\_arr[0]).split(" ", 1)

date = int(date[0])

cum = float(main\_num\_arr[0])\*date\*float(main\_proc\_arr[0])/100/365

global cumma\_1otv

cum = round(cum, 2)

cumma\_1otv["text"] = cum

else:

# вывод окна с ошибкой

msg = btn1

mb.showerror("Ошибка", msg)

# проверка ответа на заполнение

if cumma\_1otv["text"] == " ":

btn\_ex1['state'] = ttk.DISABLED

else:

btn\_ex1['state'] = ttk.NORMAL

def Cl1\_ex():

dictionary1 ={'Дата выдачи': [date\_give\_1e.get()],

'Дата гашения': [date\_take\_1e.get()],

'Сумма займа': [num\_1e.get()],

'Процентня ставка, в %':[proc\_1e.get()],

'Сумма % по займу':[cumma\_1otv['text']]}

Create\_excel\_fail( dictionary1, "Срочный займ")

btn1 = ButtonTk(frame1, text="Расчитать", font=("Arial", 14), command=Cl1, width=21)

btn1.grid(row=6, column=1, ipadx=1, ipady=1)

btn\_ex1 = ButtonTk(frame1, text="Отчёт", font=("Arial", 14), state="disabled", width=17, command=Cl1\_ex )

btn\_ex1.grid(row=6, column=0, ipadx=1, ipady=1)

#frame1.bind('<Button-1>', lambda e1: focus1(e1))

#====================================================================

#======================frame2==========================

label\_2 = ttk.Label(frame2, text="Займ с изменением процентной ставки", width=31, font=("Arial", 14)).grid(row=0, column=0, columnspan=2, ipadx=6, ipady=6, padx=4, pady=4)

date\_give\_2 = Create\_label(frame2, "Дата выдачи").grid(row=1, column=0, ipadx=6, ipady=6, padx=4, pady=4)

date\_take\_2 = Create\_label(frame2, "Дата гашения").grid(row=2, column=0, ipadx=6, ipady=6, padx=4, pady=4)

num\_2 = Create\_label(frame2, "Сумма займа").grid(row=3, column=0, ipadx=6, ipady=6, padx=4, pady=4)

proc\_2 = Create\_label(frame2, "Процентная ставка, %").grid(row=4, column=0, ipadx=6, ipady=6, padx=4, pady=4)

date\_edit\_2 = Create\_label(frame2, "Дата изменения\nпроцентной ставки").grid(row=5, column=0, ipadx=6, ipady=6, padx=4, pady=4)

proc\_2\_new = Create\_label(frame2, "Новая процентная\nставка, %").grid(row=6, column=0, ipadx=6, ipady=6, padx=4, pady=4)

cumma\_2 = Create\_label(frame2, "Сумма % по займу").grid(row=7, column=0, ipadx=6, ipady=6, padx=4, pady=4)

cumma\_2otv = Create\_label(frame2, " ")

cumma\_2otv.grid(row=7, column=1, ipadx=6, ipady=6, padx=4, pady=4)

date\_give\_2e = Create\_entry(frame2)

date\_give\_2e.grid(row=1, column=1, ipadx=1, ipady=1)

date\_take\_2e = Create\_entry(frame2)

date\_take\_2e.grid(row=2, column=1, ipadx=1, ipady=1)

num\_2e = Create\_entry(frame2)

num\_2e.grid(row=3, column=1, ipadx=1, ipady=1)

proc\_2e = Create\_entry(frame2)

proc\_2e.grid(row=4, column=1, ipadx=1, ipady=1)

date\_edit\_2e = Create\_entry(frame2)

date\_edit\_2e.grid(row=5, column=1, ipadx=1, ipady=6)

proc\_2\_newe = Create\_entry(frame2)

proc\_2\_newe.grid(row=6, column=1, ipadx=1, ipady=6)

def Cl2():

root.focus\_force()

main\_date\_arr = [date\_give\_2e.get(), date\_edit\_2e.get(), date\_take\_2e.get()]

main\_proc\_arr = [proc\_2e.get(), proc\_2\_newe.get()]

main\_num\_arr = [num\_2e.get()]

btn2 = click\_button(main\_date\_arr, main\_proc\_arr, main\_num\_arr)

if btn2==True:

#логика приложения

main\_date\_arr[0]=datetime.strptime(main\_date\_arr[0], '%d.%m.%Y')

main\_date\_arr[-1]=datetime.strptime(main\_date\_arr[-1], '%d.%m.%Y')

main\_date\_arr[1]=datetime.strptime(main\_date\_arr[1], '%d.%m.%Y')

col\_dn1 = str(main\_date\_arr[1]-main\_date\_arr[0] ).split(" ", 1)

col\_dn1 = int(col\_dn1[0])-1

col\_dn2 = str(main\_date\_arr[-1]-main\_date\_arr[1] ).split(" ", 1)

col\_dn2 = int(col\_dn2[0])+1

cum = float(main\_num\_arr[0])\*(col\_dn1\*float(main\_proc\_arr[0])+col\_dn2\*float(main\_proc\_arr[-1]))/100/365

global cumma\_2otv

cum = round(cum, 2)

cumma\_2otv["text"] = cum

else:

# вывод окна с ошибкой

msg = btn2

mb.showerror("Ошибка", msg)

# проверка ответа на заполнение

if cumma\_2otv["text"] == " ":

btn\_ex2['state'] = ttk.DISABLED

else:

btn\_ex2['state'] = ttk.NORMAL

def Cl2\_ex():

dictionary2 ={'Дата выдачи': [date\_give\_2e.get()],

'Дата гашения': [date\_take\_2e.get()],

'Сумма займа': [num\_2e.get()],

'Процентня ставка, в %':[proc\_2e.get()],

'Дата изменения процентной ставки':[date\_edit\_2e.get()],

'Навая процентная ставка, в %':[proc\_2\_newe.get()],

'Сумма % по займу':[cumma\_2otv['text']]}

Create\_excel\_fail( dictionary2, "Займ с изменением процентной ставки")

btn2 = ButtonTk(frame2, text="Расчитать", font=("Arial", 14), command=Cl2, width=21)

btn2.grid(row=8, column=1, ipadx=1, ipady=1)

btn\_ex2 = ButtonTk(frame2, text="Отчёт", font=("Arial", 14), state="disabled", width=17, command=Cl2\_ex )

btn\_ex2.grid(row=8, column=0, ipadx=1, ipady=1)

#frame2.bind('<Button-1>', lambda e2: focus2(e2))

#====================================================================

#======================frame3==========================

label\_3 = ttk.Label(frame3, text=" Займ с досрочным гашением", width=31, font=("Arial", 14)).grid(row=0, column=0, columnspan=2, ipadx=6, ipady=6, padx=4, pady=4)

date\_give\_3 = Create\_label(frame3, "Дата выдачи").grid(row=1, column=0, ipadx=6, ipady=2, padx=4, pady=3)

date\_take\_3 = Create\_label(frame3, "Дата гашения").grid(row=2, column=0, ipadx=6, ipady=2, padx=4, pady=3)

num\_3 = Create\_label(frame3, "Сумма займа").grid(row=3, column=0, ipadx=6, ipady=2, padx=4, pady=3)

proc\_3 = Create\_label(frame3, "Процентная ставка, %").grid(row=4, column=0, ipadx=6, ipady=2, padx=4, pady=3)

date\_edit\_3 = Create\_label(frame3, "Дата изменения\nпроцентной ставки").grid(row=5, column=0, ipadx=2, ipady=2, padx=4, pady=3)

proc\_3\_new = Create\_label(frame3, "Новая процентная\nставка, %").grid(row=6, column=0, ipadx=2, ipady=2, padx=4, pady=3)

date\_chg\_3 = Create\_label(frame3, "Дата част. гашения").grid(row=7, column=0, ipadx=2, ipady=2, padx=4, pady=3)

num\_chg\_3 = Create\_label(frame3, "Сумма част. гашения").grid(row=8, column=0, ipadx=2, ipady=2, padx=4, pady=3)

cumma\_3 = Create\_label(frame3, "Сумма % по займу").grid(row=9, column=0, ipadx=2, ipady=2, padx=4, pady=3)

ghz\_3 = Create\_label(frame3, "Гашение займа").grid(row=10, column=0, ipadx=2, ipady=2, padx=4, pady=3)

cumma\_3otv = Create\_label(frame3, " ")

cumma\_3otv.grid(row=9, column=1, ipadx=6, ipady=2, padx=4, pady=3)

ghz\_3otv = Create\_label(frame3, " ")

ghz\_3otv.grid(row=10, column=1, ipadx=6, ipady=2, padx=4, pady=3)

date\_give\_3e = Create\_entry(frame3)

date\_give\_3e.grid(row=1, column=1, ipadx=1, ipady=2, padx=4, pady=3)

date\_take\_3e = Create\_entry(frame3)

date\_take\_3e.grid(row=2, column=1, ipadx=1, ipady=2, padx=4, pady=3)

num\_3e = Create\_entry(frame3)

num\_3e.grid(row=3, column=1, ipadx=1, ipady=2, padx=4, pady=3)

proc\_3e = Create\_entry(frame3)

proc\_3e.grid(row=4, column=1, ipadx=1, ipady=2, padx=4, pady=3)

date\_edit\_3e = Create\_entry(frame3)

date\_edit\_3e.grid(row=5, column=1, ipadx=1, ipady=6, padx=4, pady=3)

proc\_3\_newe = Create\_entry(frame3)

proc\_3\_newe.grid(row=6, column=1, ipadx=1, ipady=6, padx=4, pady=3)

date\_chg\_3e = Create\_entry(frame3)

date\_chg\_3e.grid(row=7, column=1, ipadx=1, ipady=2, padx=4, pady=3)

num\_chg\_3e = Create\_entry(frame3)

num\_chg\_3e.grid(row=8, column=1, ipadx=1, ipady=2, padx=4, pady=3)

def Cl3():

root.focus\_force()

main\_date\_arr = [date\_give\_3e.get(), date\_edit\_3e.get(), date\_chg\_3e.get(), date\_take\_3e.get()]

main\_proc\_arr = [proc\_3e.get(), proc\_3\_newe.get()]

main\_num\_arr = [num\_3e.get(), num\_chg\_3e.get()]

btn3 = click\_button(main\_date\_arr, main\_proc\_arr, main\_num\_arr)

if btn3==True:

#логика приложения

main\_date\_arr[0]=datetime.strptime(main\_date\_arr[0], '%d.%m.%Y')

main\_date\_arr[-1]=datetime.strptime(main\_date\_arr[-1], '%d.%m.%Y')

main\_date\_arr[1]=datetime.strptime(main\_date\_arr[1], '%d.%m.%Y')

main\_date\_arr[2]=datetime.strptime(main\_date\_arr[2], '%d.%m.%Y')

col\_day1 = 0

col\_day2 = 0

col\_day3 = 0

if main\_date\_arr[2] >= main\_date\_arr[1]:

col\_day1 = str(main\_date\_arr[1] - main\_date\_arr[0]).split(" ", 1)

col\_day1 = int(col\_day1[0])-1

col\_day2 = str(main\_date\_arr[2] - main\_date\_arr[1]).split(" ", 1)

col\_day2 = int(col\_day2[0])+1

col\_day3 = str(main\_date\_arr[3] - main\_date\_arr[2]).split(" ", 1)

col\_day3 = int(col\_day3[0])

else:

col\_day1 = str(main\_date\_arr[2] - main\_date\_arr[0]).split(" ", 1)

col\_day1 = int(col\_day1[0])

col\_day2 = str(main\_date\_arr[1] - main\_date\_arr[2]).split(" ", 1)

col\_day2 = int(col\_day2[0])-1

col\_day3 = str(main\_date\_arr[3] - main\_date\_arr[1]).split(" ", 1)

col\_day3 = int(col\_day3[0])+1

cum1 = float(main\_num\_arr[0])\*col\_day1\*float(main\_proc\_arr[0])/100/365

cum2 = 0

if main\_date\_arr[2] >= main\_date\_arr[1]:

cum2 = float(main\_num\_arr[0])\*col\_day2\*float(main\_proc\_arr[1])/100/365

else:

cum2 = (float(main\_num\_arr[0]) - float(main\_num\_arr[1]))\*col\_day2\*float(main\_proc\_arr[0])/100/365

cum3 = (float(main\_num\_arr[0])- float(main\_num\_arr[1]))\*col\_day3\*float(main\_proc\_arr[1])/100/365

cum = cum1 + cum2 + cum3

ghz = float(main\_num\_arr[0]) - float(main\_num\_arr[1])

global cumma\_3otv

cum = round(cum, 2)

cumma\_3otv["text"] = cum

global ghz\_3otv

ghz = round(ghz, 2)

ghz\_3otv["text"] = ghz

else:

# вывод окна с ошибкой

msg = btn3

mb.showerror("Ошибка", msg)

# проверка ответа на заполнение

if cumma\_3otv["text"] == " ":

btn\_ex3['state'] = ttk.DISABLED

else:

btn\_ex3['state'] = ttk.NORMAL

def Cl3\_ex():

dictionary3 ={'Дата выдачи': [date\_give\_3e.get()],

'Дата гашения': [date\_take\_3e.get()],

'Сумма займа': [num\_3e.get()],

'Процентня ставка, в %':[proc\_3e.get()],

'Дата изменения процентной ставки':[date\_edit\_3e.get()],

'Навая процентная ставка, в %':[proc\_3\_newe.get()],

'Дата частичного погашения':[date\_chg\_3e.get()],

'Сумма частичного гашения':[num\_chg\_3e.get()],

'Сумма % по займу':[cumma\_3otv['text']],

'Гашение займа':[ghz\_3otv['text']]}

Create\_excel\_fail( dictionary3, "«Займ с досрочным гашением")

btn3 = ButtonTk(frame3, text="Расчитать", font=("Arial", 14), command=Cl3, width=21)

btn3.grid(row=11, column=1, ipadx=1, ipady=1)

btn\_ex3 = ButtonTk(frame3, text="Отчёт", font=("Arial", 14), state="disabled", width=17, command=Cl3\_ex )

btn\_ex3.grid(row=11, column=0, ipadx=1, ipady=1)

frame3.bind\_all('<Button-1>', lambda e3: focus3(e3))

#====================================================================

root.mainloop()