ПРОГА ИЗ НАЧАЛА

class User:  
 def \_\_init\_\_(self, name, surname, balance):  
 self.name = name  
 self.surname = surname  
 self.\_\_balance = balance  
  
 def show\_info(self):  
 print(f'Я клиент, моё имя: {self.name}, моя фамилия: {self.surname}, мой баланс {self.\_\_balance}')  
  
 def \_\_eq\_\_(self, other):  
 print("Вызван магический метод сравнения (==)")  
 if isinstance(other, User):  
 return self.\_\_balance == other.\_\_balance  
  
 def \_\_ne\_\_(self, other):  
 print("Вызван магический метод сравнения (!=)")  
 if isinstance(other, User):  
 return self.\_\_balance != other.\_\_balance  
  
 def \_\_lt\_\_(self, other):  
 print("Вызван магический метод сравнения (<)")  
 if isinstance(other, User):  
 return self.\_\_balance < other.\_\_balance  
  
 def \_\_gt\_\_(self, other):  
 print("Вызван магический метод сравнения (>)")  
 if isinstance(other, User):  
 return self.\_\_balance > other.\_\_balance  
  
 def \_\_add\_\_(self, other):  
 print("Вызван магический метод сложения")  
 if isinstance(other, User):  
 return self.\_\_balance + other.\_\_balance  
  
 def \_\_sub\_\_(self, other):  
 print("Вызван магический метод вычитания")  
 if isinstance(other, User):  
 return self.\_\_balance - other.\_\_balance  
  
  
def show\_info():  
 print('1 - добавить клиента')  
 print('2 - удалить клиента')  
 print('3 - действия с клиентами')  
 print('4 - вывести список всех клиентов')  
 print('5 - закончить работу программы')  
  
  
def add\_new\_user():  
 users.append(User(name=input('Введите имя клиента: '), surname=input('Введите фамилию клиента: '), balance=int(input('Введите баланс клиента: '))))  
  
  
def delete\_user():  
 users.remove(users[int(input("Индекс пользователя, которого необходимо удалить: ")) - 1])  
  
  
def show\_all():  
 [user.show\_info() for user in users]  
  
  
def add\_users():  
 text = input("Напишите нужную вам операцию(C + C): ")  
 text = text.split(" ")  
 text[0], text[2] = int(text[0]), int(text[2])  
 if text[1] == '+':  
 print('Суммарный баланс: ', users[text[0] - 1] + users[text[2] - 1])  
 elif text[1] == '-':  
 print('Разность балансов: ', abs(users[text[0] - 1] - users[text[2] - 1]))  
 elif text[1] == '==':  
 print('Результат операции: ', users[text[0] - 1] == users[text[2] - 1])  
 elif text[1] == '!=':  
 print('Результат операции: ', users[text[0] - 1] != users[text[2] - 1])  
 elif text[1] == '!=':  
 print('Результат операции: ', users[text[0] - 1] > users[text[2] - 1])  
 elif text[1] == '!=':  
 print('Результат операции: ', users[text[0] - 1] < users[text[2] - 1])  
  
  
users = []  
while True:  
 show\_info()  
 what\_do = input('Что делаем?\n')  
 if what\_do == '1':  
 add\_new\_user()  
 elif what\_do == '2':  
 delete\_user()  
 elif what\_do == '3':  
 add\_users()  
 elif what\_do == '4':  
 show\_all()  
 elif what\_do == '5':  
 break

ФОРМЫ

Help elements

import tkinter as tk  
from tkinter import ttk  
  
  
def create\_button(width, height, font, background, font\_color, text, command, position):  
 btn = tk.Button(  
 text=text,  
 background=background,  
 width=width,  
 height=height,  
 font=font,  
 foreground=font\_color,  
 command=command,  
 )  
 btn.place(x=position[0], y=position[1])  
 return btn  
  
  
def create\_label(font, font\_color, text, position, background):  
 label = tk.Label(  
 text=text,  
 font=font,  
 foreground=font\_color,  
 justify=tk.LEFT,  
 background=background  
 )  
 label.place(x=position[0], y=position[1])  
 return label  
  
  
def create\_entry(width, font, font\_color, position):  
 entry = tk.Entry(  
 width=width,  
 font=font,  
 foreground=font\_color,  
 )  
 entry.place(x=position[0], y=position[1])  
 return entry  
  
  
def create\_combo\_box(width, font, font\_color, position, values, default=0, callback=None):  
 combo = ttk.Combobox(  
 width=width,  
 font=font,  
 foreground=font\_color,  
 values=values,  
 state="readonly",  
 )  
 combo.bind("<<ComboboxSelected>>", callback)  
 combo.current(default)  
 combo.place(x=position[0], y=position[1])  
 return combo

MainForm

import tkinter as tk  
from help\_elements import create\_button, create\_entry, create\_label, create\_combo\_box  
  
  
class Patient:  
 def \_\_init\_\_(self, name, surname, second\_surname, data):  
 self.name = name  
 self.surname = surname  
 self.birthday = data  
 self.second\_surname = second\_surname  
  
  
class MainForm:  
 def \_\_init\_\_(self):  
 self.active\_elements = {}  
 self.window = tk.Tk()  
 self.patients = []  
 self.window.geometry('1400x875')  
 self.window.title("Ванюююша самый лучший")  
 self.show\_main\_screen()  
 self.window.mainloop()  
  
 def destroy\_all(self):  
 for elem in self.active\_elements:  
 self.active\_elements[elem].destroy()  
 self.active\_elements.clear()  
  
 def add\_new\_patient(self):  
 self.patients.append(Patient(self.active\_elements['patient\_name\_entry'].get(),  
 self.active\_elements['patient\_surname\_entry'].get(),  
 self.active\_elements['patient\_second\_surname\_entry'].get(),  
 self.active\_elements['patient\_birthaday\_date\_entry'].get()))  
 self.show\_main\_screen()  
  
 def show\_main\_screen(self):  
 self.destroy\_all()  
 self.active\_elements['patient\_label\_count'] = create\_label(font\_color="#0C8EEC", text=f"Количество докторов: {len(self.patients)}",  
 position=[400, 100], backround="#b5effb",  
 font="Sedan 14")  
 self.active\_elements['patient\_label'] = create\_label(font\_color="#0C8EEC",  
 text="Страница добавления пациента",  
 position=[400, 40], background="#b5effb",  
 font="Sedan 14")  
 self.active\_elements['patient\_name'] = create\_label(font\_color="#000000",  
 text="Имя пациента", position=[325, 275],  
 background="#b5effb",  
 font="Sedan 14")  
 self.active\_elements['patient\_name\_entry'] = create\_entry(width=25, font="Sedan 14",  
 position=[250, 325], font\_color="#000000")  
 self.active\_elements['patient\_surname'] = create\_label(font\_color="#000000",  
 text="Фамилия пациента", position=[325, 175],  
 background="#b5effb",  
 font="Sedan 14")  
 self.active\_elements['patient\_surname\_entry'] = create\_entry(width=25, font="Sedan 14",  
 position=[250, 225], font\_color="#000000")  
 self.active\_elements['patient\_second\_surname'] = create\_label(font\_color="#000000",  
 text="Отчество пациента",  
 position=[325, 375],  
 background="#b5effb",  
 font="Sedan 14")  
 self.active\_elements['patient\_second\_surname\_entry'] = create\_entry(width=25, font="Sedan 14",  
 position=[250, 425],  
 font\_color="#000000")  
 self.active\_elements['patient\_birthaday\_date'] = create\_label(font\_color="#000000", text="Дата рождения пациента", position=[300, 475],  
 background="#b5effb",  
 font="Sedan 14")  
 self.active\_elements['patient\_birthaday\_date\_entry'] = create\_entry(width=25, font="Sedan 14",  
 position=[250, 525],  
 font\_color="#000000")  
 self.active\_elements['complete'] = create\_button(font\_color='#ffffff', text="Выполнить", command=self.add\_new\_patient, position=[325, 575],  
 background='#2998E9', width=12, height='3',  
 font="Sedan 12")

main

from Classes.MainForm import MainForm  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 MainForm()