Міністерство освіти та науки України

Харківський національний університет радіоелектроніки

Кафедра Інформатики

Звіт

з лабораторної роботи №2

з дисципліни «Теорія програмування»

Виконав: Перевірив:

Ст.гр. ІТІНФ-20-1 Сінельнікова. Т. Ф.

Самченко С. О.

Харків 2023

**Мета роботи**: Опанувати теоретичні знання про патерни програмування, їх розуміння і вміння їх застосовувати на практиці. Опанувати практичні навички застосування патернів програмування у проектах та усвідомити їхню практичну цінність у процесі розробки програмного забезпечення.

**Завдання:** Розробити додаток із астосуванням одного або декількох петернів програмування.

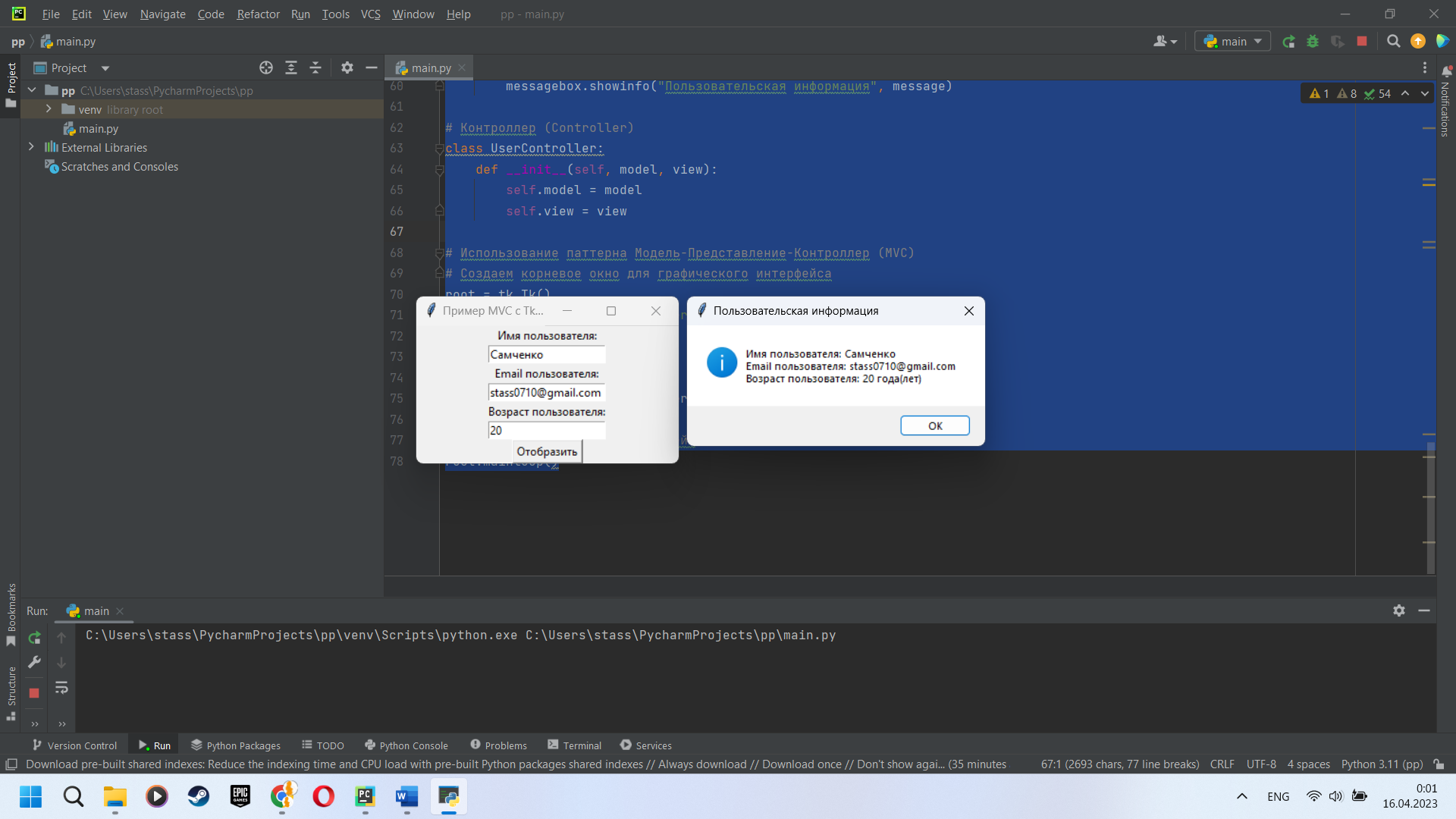
**Хід роботи:**

import tkinter as tk  
from tkinter import messagebox  
  
# Базовый класс (Component)  
class UserModel:  
 def \_\_init\_\_(self):  
 self.username = ""  
 self.email = ""  
  
 def display\_user(self):  
 return "Имя пользователя: {}\nEmail пользователя: {}".format(self.username, self.email)  
  
# Декоратор (Decorator)  
class UserModelDecorator(UserModel):  
 def \_\_init\_\_(self, user\_model):  
 self.\_user\_model = user\_model  
  
 def display\_user(self):  
 return self.\_user\_model.display\_user()  
  
 def display\_user\_with\_age(self, age):  
 return "{}\nВозраст пользователя: {} года(лет)".format(self.\_user\_model.display\_user(), age)  
  
# Представление (View)  
class UserView:  
 def \_\_init\_\_(self, root):  
 self.root = root  
 self.username\_label = tk.Label(root, text="Имя пользователя:")  
 self.username\_label.pack()  
 self.username\_entry = tk.Entry(root)  
 self.username\_entry.pack()  
 self.email\_label = tk.Label(root, text="Email пользователя:")  
 self.email\_label.pack()  
 self.email\_entry = tk.Entry(root)  
 self.email\_entry.pack()  
 self.age\_label = tk.Label(root, text="Возраст пользователя:")  
 self.age\_label.pack()  
 self.age\_entry = tk.Entry(root)  
 self.age\_entry.pack()  
 self.display\_button = tk.Button(root, text="Отобразить", command=self.display\_user)  
 self.display\_button.pack()  
  
 def display\_user(self):  
 username = self.username\_entry.get()  
 email = self.email\_entry.get()  
 age = self.age\_entry.get()  
 # Проверяем, что введенное значение возраста является целым числом  
 if age.isdigit():  
 user\_model = UserModel()  
 user\_model.username = username  
 user\_model.email = email  
 # Декорируем объект UserModel, добавляя возраст  
 user\_model\_with\_age = UserModelDecorator(user\_model)  
 message = user\_model\_with\_age.display\_user\_with\_age(age)  
 self.show\_message(message)  
 else:  
 self.show\_message("Ошибка: Возраст должен быть целым числом!")  
  
 def show\_message(self, message):  
 messagebox.showinfo("Пользовательская информация", message)  
  
# Контроллер (Controller)  
class UserController:  
 def \_\_init\_\_(self, model, view):  
 self.model = model  
 self.view = view  
  
# Использование паттерна Модель-Представление-Контроллер (MVC)  
# Создаем корневое окно для графического интерфейса  
root = tk.Tk()  
root.title("Пример MVC с Tkinter")  
# Создаем объект представления  
user\_view = UserView(root)  
# Создаем объект контроллера  
user\_controller = UserController(None, user\_view)  
  
# Запускаем графический интерфейс  
root.mainloop()

**Опис:**

Дана програма створює вікно для введення даних користувача. ці дані після натискання на кнопку "Надіслати" виводяться в окремому вікні. Також у цьому додатку використовується перевірка на коректне введення даних у полі "Ворость пользователя".

**Результат:**



У цьому коді використовуються такі патерни програмування:

* *Модель-Подання-Контролер (MVC)* - у коді реалізований патерн MVC, де UserModel представляє модель, UserView представляє уявлення, і UserController представляє контролер. Модель (UserModel) містить дані та логіку, Подання (UserView) відповідає за відображення даних та взаємодію з користувачем, і Контролер (UserController) обробляє події та взаємодіє з моделлю та поданням для реалізації бізнес-логіки.
* *Декоратор (Decorator)* - у коді реалізований патерн декоратор, де UserModelDecorator декорує базовий клас UserModel, додаючи додаткову функціональність (у разі, відображення віку користувача).
* *Одинак (Singleton)* - в даному коді об'єкт класу tk.Tk(), який представляє кореневе вікно для графічного інтерфейсу, створюється лише один раз і використовується як єдиний екземпляр (singleton) протягом усього виконання програми.

**Висновок:** у рамках даної роботи були здобуті теоретичні знання про патерни програмування, їх розуміння і вміння їх застосовувати на практиці та практичні навички застосування патернів програмування у проектах та усвідомити їхню практичну цінність у процесі розробки програмного забезпечення.