**Міністерство освіти і науки України**

**Національний університет «Запорізька політехніка»**

Кафедра програмних засобів

**ЗВІТ**

Дисципліна «Розробка прикладних програм»

Робота №1

Тема «Розроблення програмного забезпечення з графічним інтерфейсом на основі об’єктно-орієнтованого програмування мовою Python»

**Виконав варіант 19**

Студент КНТ-122 Онищенко О. А.

**Прийняли**

Викладач Дейнега Л. Ю.

2024

[Мета роботи 3](#_Toc180046256)

[Індивідуальне завдання 3](#_Toc180046257)

[Тексти файлів 3](#_Toc180046258)

[constants.py 3](#_Toc180046259)

[movies.json 3](#_Toc180046260)

[objects.py 4](#_Toc180046261)

[run.py 4](#_Toc180046262)

[Результати виконання 7](#_Toc180046263)

[Контрольні питання 8](#_Toc180046264)

[Поняття модулю та пакету 8](#_Toc180046265)

[Засоби бібліотеки tkinter 9](#_Toc180046266)

[Оголошення класу Python 10](#_Toc180046267)

[Принципи об’єктно-орієнтованого програмування у Python 10](#_Toc180046268)

[*Приклад поліморфізму* 11](#_Toc180046269)

[*Приклад абстракції* 11](#_Toc180046270)

Мета роботи

Ознайомитися з принципами реалізації об’єктно-орієнтованого програмування у мові Python та навчитися використовувати його для розроблення програмного забезпечення.

Навчитися розробляти сучасні графічні інтерфейси користувача для програм, написаних мовою Python.

Індивідуальне завдання

Продаж квитків у кінотеатр з можливістю переглядати фільми, переглядати доступні та зайняті місця для перегляду заданого фільму у відповідній залі, бронювання та звільнення місць.

Тексти файлів

constants.py

# всі кольори обрано зі стандартних кольорів tkinter

GREEN\_BUTTON="lawn green"

BLUE\_BUTTON="dodger blue"

# ці кольори незавершені

# вони мають варіації з номером на кінці

COLORS=["AntiqueWhite","LightSteelBlue","khaki"]

movies.json

[

{

"name": "Страсті Христові",

"description": "Рік 33 нашої ери. У римській провінції Юдея таємничий столяр на ім’я Ісус із Назарету починає сповіщати про прихід «Божого царства» і оточує себе групою покірних рибалок: Апостолів. Протягом століть єврейські люди чекали приходу Месії — провідної фігури, яка звільнить свою священну батьківщину та встановить новий порядок, заснований на справедливості. Вчення Ісуса привертає велику кількість послідовників, які визнають Його Месією. Стурбований ситуацією, синедріон за допомогою Юди Іскаріота, одного з дванадцяти апостолів, арештовує Ісуса. Звинувачений у державній зраді проти Риму, Христос передається Понтію Пілату, який, щоб уникнути бунту, засуджує його на смерть на хресті як звичайного злочинця.",

"year": 2004,

"image": "https://image.tmdb.org/t/p/w1280/v9f9MMrq2nGQrN7cHnQRmEq9lSE.jpg"

},

{

"name": "Син Божий",

"description": "Історія Ісуса Христа від Народження, до Його вчення, Розп'яття і Вознесіння.",

"year": 2014,

"image": "https://image.tmdb.org/t/p/w1280/1ICN5qakevvkAKI2MKWDhTFm9kh.jpg"

},

{

"name": "Народження Христа",

"description": "Фільм оповідає про життєвий шлях Діви Марії та Йосипа, починаючи з моменту їх вигнання з Назарету та закінчуючи прибуттям у Віфлєєм, де, як відомо, повинен народитися Ісус. У цій непростій подорожі не раз перевірятиметься на міцність їх любов та істинність переконань.",

"year": 2006,

"image": "https://image.tmdb.org/t/p/w1280/vB55RLWmO2NYdGwL0XG30769cPO.jpg"

}

]

objects.py

# головний об'єкт вікна програми

window=None

# глобально обраний фільм

movie=None

# глобально вибрана зала

room=None

# контейнери для фільмів, зал, місць

movies\_container=None

rooms\_container=None

seats\_container=None

run.py

import os

import json

import random

from tkinter import \*

from constants import \*

from objects import \*

class Movie:

def \_\_init\_\_(self, name:str, description:str, year:int, image: str):

self.name:str=name

self.description:str=description

self.year:int=year

# посилання на файл зображення

self.image:str=image

self.rooms=[

# генеруємо три зали

[

# у кожній залі маємо три ряди сидінь

[

# у кожному ряду маємо сім місць

# заповнюємо місця "випадковими" булевими

random.choice([0,1])

for seat in range(7)

]

for row in range(3)

]

for room in range(3)

]

class DataLoader:

@staticmethod

def load\_movies() -> list[Movie]:

"""

Завантажує дані про фільми з файлу JSON, конвертує їх до об'єктів типу Movie

> Список завантажених фільмів як масив об'єктів

"""

current\_folder:str=os.path.dirname(os.path.abspath(\_\_file\_\_))

movies\_data\_file\_path:str=os.path.join(current\_folder,"movies.json")

movies:list[Movie]=[]

with open(movies\_data\_file\_path,mode="r",encoding="utf-8") as f: movies\_data:dict=json.load(f)

for movie in movies\_data:

# оскільки ключі прочитаних об'єктів співпадають з полями класу

# їх можна одразу конвертувати у об'єкти

movie\_object:Movie=Movie(\*\*movie)

movies.append(movie\_object)

return movies

class Window(Tk):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.geometry("700x300")

self.resizable(0,0)

self.title("Кінотеатр 'ІСУСОВА Благодать'")

# при натисканні кнопки Esc програма закривається

self.bind("<Escape>",lambda \_: window.destroy())

class Container(Frame):

def \_\_init\_\_(self,parent,expand:bool=0):

super().\_\_init\_\_(parent,background="white")

self.pack(fill=BOTH,expand=expand)

def clear(self):

for widget in self.winfo\_children():

widget.destroy()

def place\_button(button\_object:Button,command):

button\_object.config(command=command)

button\_object.pack(side=LEFT,expand=1,fill=BOTH)

def on\_movie\_select(movie\_index:int):

global movie

movie=movies[movie\_index]

build\_rooms()

def build\_movies():

movies\_container.clear()

for movie\_index,movie in enumerate(movies):

movie\_button=Button(

movies\_container,

# текст кнопки ставимо на назву фільму

text=movie.name,

# обираємо колір за номером фільму

background=f"{COLORS[movie\_index]}1"

)

place\_button(movie\_button,lambda movie\_index=movie\_index:on\_movie\_select(movie\_index))

def on\_room\_select(room\_index:int):

global room

room=movie.rooms[room\_index]

build\_seats()

def build\_rooms():

rooms\_container.clear()

for room\_index,room in enumerate(movie.rooms):

room\_button=Button(

rooms\_container,

# текст кнопки ставимо як номер зари

text=f"Зала {room\_index+1}",

# колір ставимо як номер фільму

background=f"{COLORS[movies.index(movie)]}1"

)

place\_button(room\_button,lambda room\_index=room\_index:on\_room\_select(room\_index))

def choose\_button\_color(value): return GREEN\_BUTTON if not value else BLUE\_BUTTON

def on\_seat\_select(row\_index,seat\_index,button\_object):

global room

# обираємо протилежне значення до поточного

room[row\_index][seat\_index]=0 if room[row\_index][seat\_index] else 1

# ставимо колір зелений якщо місце вільне, синій якщо зайняте

button\_object.config(background=choose\_button\_color(room[row\_index][seat\_index]))

def build\_seats():

seats\_container.clear()

for row\_index, row in enumerate(room):

row\_container=Container(seats\_container,expand=1)

for seat\_index,seat in enumerate(room[row\_index]):

seat\_button=Button(row\_container,background=choose\_button\_color(seat))

place\_button(seat\_button,lambda row\_index=row\_index,seat\_index=seat\_index,button\_object=seat\_button:on\_seat\_select(row\_index,seat\_index,button\_object))

window=Window()

movies=DataLoader.load\_movies()

movies\_container=Container(window)

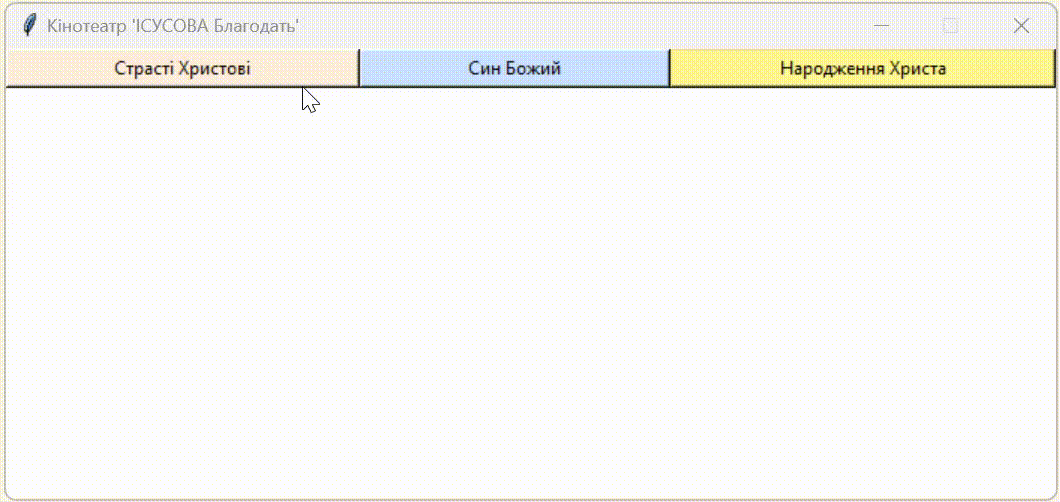
rooms\_container=Container(window)

seats\_container=Container(window,expand=1)

build\_movies()

window.mainloop()

Результати виконання



GIF Зображення 1.1 – Процес роботи з програмою

Зображення, що містить текст, знімок екрана, програмне забезпечення, Операційна система

Автоматично згенерований опис

Рисунок 1.1 – Початковий вигляд програми

Зображення, що містить знімок екрана, текст, квадрат, Барвистість

Автоматично згенерований опис

Рисунок 1.2 – Вигляд програми в користуванні

Контрольні питання

Поняття модулю та пакету

Модуль це окремий файл з кодом.

Пакет це тека з файлом \_\_init\_\_.py і кількома модулями всередині.

**Перетворення файлу у простір імен**

При імпортуванні його до іншого файлу. Тоді до простору імен можна мати доступ через назваФайлу.назваФункції.

Назву простору імен можна змінити якщо додати до рядку імпорту параметер as. Наприклад: для імпортування модулю math з назвою m можна використати такий код: import math as m. Тоді для доступу до функції sqrt використовуємо наступний код: m.sqrt(7)

**Пакети Python для побудови графічного інтерфейсу**

* Tkinter
* PyQT
* Kivy

Засоби бібліотеки tkinter

Віджети:

* Button,
* Label,
* Entry,
* Checkbox,
* Combobox,
* Menu,
* Scale

Контейнери:

* Toplevel,
* Frame,
* Notebook

Розташувальники:

* pack(),
* grid(),
* place()

[Повна документацію tkinter](https://docs.python.org/uk/3/library/tkinter.html)

Оголошення класу Python

Клас визначається ключовим словом class. Далі назва класу, краще у форматі PascalCase (кожне слово з великої літери, без розділення: PlaneSeat)

Спадкування: до оголошення класу додаємо дужки з назвою класу - або кількох - від яких успадковуємо: class PlaneSeat(Seat)

Конструктор: визначається оголошенням методу \_\_init\_\_(). Для кожного методу класу як перший параметр додається self для доступу до полів.

Статичний метод: додати декоратор @staticmethod перед оголошення методу.

Принципи об’єктно-орієнтованого програмування у Python

Статуси полів класу (private, public, protected) у пайтоні не передбачені, але можна визначити «приватне» поле додавши нижнє підкреслення перед назвою: self.\_seatNumber

Наслідування класів реалізується додаванням назв бажаних класів у дужки після назви класу при оголошенні: class Plane(Transport)

Поліморфізм реалізовується через використання зовнішньої функції. Приклад нижче.

Абстракцію реалізуємо використанням декоратора @abstractmethod. Приклад нижче.

*Приклад поліморфізму*

Код:

class Tvaryna:

fraza = "**ІСУС - ГОСПОДЬ**"

def kazhy(self):

print("Звук ІСУСОВОЇ Дивовижної Благодаті")

class Golubka(Tvaryna):

def kazhy(self):

print(f"Голубка каже: {super().fraza}")

class Lev(Tvaryna):

def kazhy(self):

print(f"Лев каже: {super().fraza}")

def roby\_zvuk(tvaryna: Tvaryna):

tvaryna.kazhy()

Консоль:

Голубка каже: **ІСУС - ГОСПОДЬ**

Лев каже: **ІСУС – ГОСПОДЬ**

*Приклад абстракції*

Код:

from abc import abstractmethod

class Transport:

def \_\_init\_\_(self,name:str):

self.name=name

@abstractmethod

def show\_kind(self): ...

class Plane(Transport):

def \_\_init\_\_(self,name:str,kind:str):

super().\_\_init\_\_(name)

self.kind:str=kind

def show\_kind(self): print(f"Транспорт типу {self.kind}")

plane\_object=Plane("Boeing 777","літак")

plane\_object.show\_kind()

Консоль:

Транспорт типу літак