**Міністерство освіти і науки України**

**Національний університет «Запорізька Політехніка»**

Кафедра програмних засобів

**♰ ЗВІТ**

Практична робота №1

Дисципліна “Розробка прикладних програм”

Тема “Розроблення програмного забезпечення з графічним інтерфейсом на основі об’єктно-орієнтованого програмування мовою python”

**Виконав**

Студент КНТ-122 О. А. Онищенко

**Прийняли**

Викладач Л. Ю. Дейнега

2024

Мета роботи

Ознайомитися з принципами реалізації об’єктно-орієнтованого програмування мовою python та навчитися використовувати його для розроблення програмного забезпчення. Навчитися розробляти сучасні графічні інтерфейси користувача для програм мовою python.

Завдання до роботи

Текст програми

Робота програми

Сценарій роботи №1

Інтерфейс програми

Висновки

**Так бо БОГ полюбив світ, що дав Сина Свого Єдиного, аби кожен, хто вірує в Нього, не згинув, але мав життя вічне** ([Йоан 3:16](https://www.bible.com/uk/bible/compare/JHN.3.16))

По виконанню роботи ми ознайомилися з принципами реалізації об’єктно-орієнтованого програмування мовою python та навчилися використовувати їх для розроблення програмного забезпчення. Також ми навчилися розробляти сучасні графічні інтерфейси користувача для програм мовою python.

Контрольні питання

Що таке модуль та пакет?

Модуль це окремий файл з кодом. Пакет це тека з файлом \_\_init\_\_.py і кількома модулями всередині.

Яким чином файли трансформуються в простори 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 mathз назвою m можна використати такий код: import math as m. Тоді для доступу до функції sqrt використовуємо наступний: m.sqrt(7)

Які існують шаблони проєктування?

Які пакети використовуються для побудови графічних інтерфейсів у python?

Tkinter досить розповсюджений і простий у використанні. На його основі є гарний пакет customtkinter який додає вбудовані стилі до елементів.

PyQT кажуть ще люди використовують. Можемо припустити що цей пакет якось пов’язаний з програмною розробки QT.

Kivy наче як кросплатформний фреймворк, теж люди використовують.

Які засоби має бібліотека tkinter?

Серед віджетів має Button, Label, Entry, Checkbox, Combobox, Menu, Scale, … більше віджетів [за посиланням](https://tk-tutorial.readthedocs.io/en/latest/intro/intro.html)

Серед контейнерів має Toplevel, Frame, Notebook, …

Серед засобів розміщення є pack(), grid(), place()

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

Яким чином визначається клас у python?

Клас визначається за допомогою ключового слова class. Після нього йде назва класу, за конвенціями у форматі PascalCase.

PascalCase пише кожне слово з великої літери і не розділяє слова жодними знаками, але поєднує всі слова в одне. Наприклад: PlaneSeat

Для додавання спадкування до оголошення класу додаються дужки з назвою класу або кількох класів від яких будемо успадковувати. Наприклад: class PlaneSeat(Seat)

Конструктор класу визначається назвою методу \_\_init\_\_. Для кожного методу класу як перший параметр додається self для доступу до полів класу.

Для оголошення статичного методу класу використовується декоратор @staticmethod перед рядком оголошення методу.

Яким чином реалізуються принципи об’єктно-орієнтованого програмування в python?

Статуси полів класу у пайтоні не передбачені, але згідно конвенцій можна позначити приватне поле додавши нижнє підкреслення перед назвою поля. Наприклад: self.\_seatNumber

Наслідування класів, як зазначено вище, реалізується через додавання назв бажаних класів у дужки після назви класу під час його декларації. Приклад: class Plane(Transport)

Поліморфізм можна реалізувати через використання зовнішньої функції. Приклад дивитися нижче.

Абстракцію реалізуємо використанням декоратора @abstractmethod. Приклад використання абстракції нижче.

Приклад поліморфізму

class Animal:

def sound(self) -> str:

print("sound of JESUS' Amazing Grace")

class Dove(Animal):

def sound(self) -> str:

return "Dove says: 'JESUS is KING'"

class Lion(Animal):

def sound(self) -> str:

return "Lion says: GOD ALMIGHTY Reigns forever'"

def make\_sound(animal: Animal) -> None:

print(animal.sound())

make\_sound(Dove())

make\_sound(Lion())

Вивід у консоль:

Dove says: 'JESUS is KING'

Lion says: 'GOD ALMIGHTY Reigns forever'

Приклад абстракції

Вивід у консоль: