Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ

ЗВІТ

до лабораторної роботи №9

З дисципліни: «Кросплатформні засоби програмування»

На тему: «ОСНОВИ ОБ’ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ У PYTHON»

Варіант 26

Виконав:

ст. групи КІ-306

Тимків С.В.

Прийняв:

Олексів М.В.

Львів – 2024

Мета: оволодіти навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python.

Завдання:

1. Написати та налагодити програму на мові Python згідно варіанту. Програма має задовольняти наступним вимогам:
    - класи програми мають розміщуватися в окремих модулях в одному пакеті;
    - точка входу в програму (main) має бути в окремому модулі;
    - мають бути реалізовані базовий і похідний класи предметної області згідно варіанту;
    - програма має містити коментарі.
2. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.
4. Дати відповідь на контрольні запитання.

GitHub Repository:
https://github.com/SofiaTymkiv/CPPT_Tymkiv_SV_KI-36_2.git

Базовий клас згідно варіанту №26: «Шлюпка на веслах»
Похідний клас згідно варіанту №26: «Моторний човен»

Хід роботи

Код програми:

Main.py

```
from Person import Person
from Location import Location
from Supply import Supply
from MotorBoat import MotorBoat

if __name__ == "__main__":
    # Create owner and initial supplies
    owner = Person("John Doe", 45)
    supplies = [
        Supply(10, "Water Bottles"),
        Supply(5, "Food Packs")
    ]
    start_location = Location("USA", "New York", "Hudson River")

    # Create a motor boat
    boat = MotorBoat(owner, supplies, start_location)

    # 1. Add a new supply
    boat.add_supply(Supply(15, "Life Jackets"))
```

```
    # 2. Remove a supply
    boat.remove_supply("Food Packs")

    # 3. Move the boat to a new location
    new_location = Location("Canada", "Toronto", "Lake Ontario")
    boat.move_to_new_location(new_location)

    # 4. Get total supplies on the boat
    total_supplies = boat.get_total_supplies()
    print(f"Total supplies: {total_supplies}")

    # 5. Get owner's name
    owner_name = boat.get_owner_name()
    print(f"Boat owner: {owner_name}")

    # 6. Change the owner of the boat
    new_owner = Person("Jane Smith", 30)
    boat.change_owner(new_owner)

    # 7. Get current location of the boat
    current_location = boat.get_current_location()
    print(f"Current location: {current_location.city}")

    # 8. Check if a specific supply exists
    has_water = boat.has_supply("Water Bottles")
    print(f"Has water bottles: {has_water}")

    # 9. Get the total quantity of all supplies
    total_quantity = boat.get_total_supply_quantity()
    print(f"Total supply quantity: {total_quantity}")

    # 10. Print a summary of the boat
    boat.print_boat_summary()
```

MotorBoat.py

```
from RowingBoat import RowingBoat

class MotorBoat(RowingBoat):
    def __init__(self, owner, supplies, location):
        super().__init__(owner, supplies, location)

    def print_boat_summary(self):
        print("Just a cool motor boat, hell yeah!!")
```

RowingBoat.py

```
class RowingBoat:
    def __init__(self, owner, supplies, location):
        self.owner = owner
        self.supplies = supplies
```

```
        self.location = location

    def add_supply(self, supply):
        self.supplies.append(supply)

    def remove_supply(self, supply_name):
        self.supplies = [s for s in self.supplies if s.name != supply_name]

    def move_to_new_location(self, new_location):
        self.location = new_location

    def get_total_supplies(self):
        return len(self.supplies)

    def get_owner_name(self):
        return self.owner.name

    def change_owner(self, new_owner):
        self.owner = new_owner

    def get_current_location(self):
        return self.location

    def has_supply(self, supply_name):
        return any(s.name == supply_name for s in self.supplies)

    def get_total_supply_quantity(self):
        return sum(s.quantity for s in self.supplies)

    def print_boat_summary(self):
        print(f"Boat Summary:")
        print(f"Owner: {self.owner}")
        print(f"Location: {self.location}")
        print("Supplies:")
        for supply in self.supplies:
            print(f"  {supply}")
```

Person.py

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __str__(self):
        return f"{self.name}, Age: {self.age}"
```

Location.py

```
class Location:
    def __init__(self, country, city, street):
        self.country = country
```

```
        self.city = city
        self.street = street

    def unknown():
        return Location("unknown", "unknown", "unknown")

    def __str__(self):
        return f"{self.country}, {self.city}, {self.street}"
```

Supply.py

```
class Supply:
    def __init__(self, quantity, name):
        self.quantity = quantity
        self.name = name

    def __str__(self):
        return f"{self.name} (Quantity: {self.quantity})"
```

```
Total supplies: 2
Boat owner: John Doe
Current location: Toronto
Has water bottles: True
Total supply quantity: 25
Just a cool motor boat, hell yeah!!
PS E:\work\завдання\Кросплатформні засоби програмування\26вар\Lab9>
```

Рис.1 Вивід результату у консоль

Висновок: На лабораторній роботі я оволоділа навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python.