МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ І ІНФОРМАТИКИ

Практична робота №5

з дисципліни «Методи та системи штучного інтелекту»

на тему «ПРОЕКТУВАННЯ КЛАСІВ ТА ЇХ ІЄРАРХІЙ»

Варіант 4

Виконав:

студент гр. КІБ-21

Андрій Севастьянов

Перевірив:

асистент кафедри ПМІ

Андрій НІКІТЕНКО

Луцьк – 2024

**Мета роботи**: вивчити та засвоїти базові навички роботи з класами та їх спадкуванням у Python. Ознайомитися з поняттями конструктор, деструктор, метод класу, рівні доступу, перевизначення методу, ієрархія класів, доповнення методу та отримати практичні навички їх застосування.

**Завдання:**

Перша частина

Створіть клас planet, який містить інформацію про планету сонячної системи, що включає наступні атрибути: назва, діаметр, масу і відстань від Сонця в тисячах кілометрів у закритій частині класу. В класі повинна бути відкрита функція, яка повертає відстань від Сонця в милях і відкрита функцію, яка виводить усю інформацію про планету на екран.

Друга частина

У таблиці задано сімейство об'єктів, що мають деяку схожість. Необхідно виділити найбільш загальні риси об'єктів, на основі яких скласти базовий клас. На основі базового класу розробити ієрархію класів- нащадків, кінцевими гілками в якій будуть задані об'єкти. Ієрархія класів повинна бути мінімум трирівнева, тобто між базовим класом і класами, що описують задані об'єкти, повинен бути хоча б один проміжний рівень ієрархії. Продемонструвати роботу з об'єктами, заданими за таблицею 6.2. У звіті привести діаграму класів.  
  
кухня, спальня, аудиторія, кабінет, ванна кімната, спортивний зал, гараж

Програмний код першої частини:  
class Planet:  
 def \_\_init\_\_(self, name, diameter, mass, distance\_from\_sun):  
 self.\_\_name = name  
 self.\_\_diameter = diameter  
 self.\_\_mass = mass  
 self.\_\_distance\_from\_sun = distance\_from\_sun  
 def \_\_str\_\_(self):  
 return f"Назва планети: {self.\_\_name}\nДіаметр планети: {self.\_\_diameter} км\nМаса планети: {self.\_\_mass} кг\nВідстань від Сонця: {self.\_\_distance\_from\_sun} тис. км\nВідстань від Сонця в милях: {self.distance\_from\_sun\_in\_miles()} миль\n"  
 def distance\_from\_sun\_in\_miles(self):  
 return self.\_\_distance\_from\_sun \* 0.621371192 *# переведення кілометрів в милі* def display\_info(self):  
 print(self)  
  
 def get\_name(self):  
 return self.\_\_name  
  
 def get\_diameter(self):  
 return self.\_\_diameter  
  
 def get\_mass(self):  
 return self.\_\_mass  
  
 def get\_distance\_from\_sun(self):  
 return self.\_\_distance\_from\_sun  
  
*# Створення об'єктів класу Planet*mercury = Planet("Меркурій", 4879, 3.3011e23, 57909)  
venus = Planet("Венера", 12104, 4.8675e24, 108160)  
earth = Planet("Земля", 12742, 5.972e24, 149600)  
mars = Planet("Марс", 6792, 6.4171e23, 227936)  
  
*# Демонстрація роботи методів класу*mercury.display\_info()  
print(mercury.get\_name())  
venus.display\_info()  
earth.display\_info()  
mars.display\_info()

Програмний код другої частини:

*# Базовий клас Приміщення*class Room:  
 def \_\_init\_\_(self, room\_type, functionality):  
 self.\_\_room\_type = room\_type *# Тип приміщення* self.\_\_functionality = functionality *# Функціональність* def set\_room\_type(self, type):  
 self.\_\_room\_type = type  
 def set\_functionality(self, func):  
 self.\_\_functionality = func  
 *# Метод для виведення інформації про приміщення* def display\_info(self):  
 print(f"Тип приміщення: {self.\_\_room\_type}\nФункціональність: {self.\_\_functionality}\n", end = '')  
  
  
*# Клас Вітальня*class LivingRoom(Room):  
 def \_\_init\_\_(self, room\_type, functionality, furniture):  
 super().\_\_init\_\_(room\_type, functionality)  
 self.\_\_furniture = furniture *# Меблі* def add\_furniture(self, \*items):  
 self.\_\_furniture.extend(items)  
 print("Меблі успішно додано!")  
  
 *# Перевизначений метод для виведення інформації про житлове приміщення* def display\_info(self):  
 super().display\_info()  
 print(f"Меблі: {', '.join(self.\_\_furniture)}\n", end = '')  
  
  
*# Клас Робоче приміщення*class Workspace(Room):  
 def \_\_init\_\_(self, room\_type, functionality, equipment):  
 super().\_\_init\_\_(room\_type, functionality)  
 self.\_\_equipment = equipment *# Обладнання для роботи* def add\_equipment(self, \*items):  
 self.\_\_equipment.extend(items)  
 print("Обладнання успішно додано!")  
  
 *# Перевизначений метод для виведення інформації про робоче приміщення* def display\_info(self):  
 super().display\_info()  
 print(f"Обладнання: {', '.join(self.\_\_equipment)}\n", end = '')  
  
  
*# Клас Кухня*class Kitchen(LivingRoom):  
 def \_\_init\_\_(self, room\_type, functionality, furniture, appliances):  
 super().\_\_init\_\_(room\_type, functionality, furniture)  
 self.\_\_appliances = appliances *# Побутова техніка* def add\_appliances(self, \*items):  
 self.\_\_appliances.extend(items)  
 print("Побутова техніка успішно додана!")  
  
 *# Перевизначений метод для виведення інформації про кухню* def display\_info(self):  
 super().display\_info()  
 print(f"Побутова техніка: {', '.join(self.\_\_appliances)}\n")  
  
  
*# Клас Спальня*class Bedroom(LivingRoom):  
 def \_\_init\_\_(self, room\_type, functionality, furniture, bed\_size):  
 super().\_\_init\_\_(room\_type, functionality, furniture)  
 self.\_\_bed\_size = bed\_size *# Розмір ліжка  
 # Перевизначений метод для виведення інформації про спальню* def display\_info(self):  
 super().display\_info()  
 print(f"Розмір ліжка: {self.\_\_bed\_size}\n")  
  
  
*# Клас Аудиторія*class Auditorium(Workspace):  
 def \_\_init\_\_(self, room\_type, functionality, equipment, capacity):  
 super().\_\_init\_\_(room\_type, functionality, equipment)  
 self.\_\_capacity = capacity *# Ємність (кількість місць)  
 # Перевизначений метод для виведення інформації про аудиторію* def display\_info(self):  
 super().display\_info()  
 print(f"Ємність: {self.\_\_capacity}\n")  
  
  
*# Клас Офіс*class Office(Workspace):  
 def \_\_init\_\_(self, room\_type, functionality, equipment, desks):  
 super().\_\_init\_\_(room\_type, functionality, equipment)  
 self.\_\_desks = desks *# Кількість робочих столів  
 # Перевизначений метод для виведення інформації про офіс* def display\_info(self):  
 super().display\_info()  
 print(f"Кількість робочих столів: {self.\_\_desks}\n")  
  
*# Клас Ванна кімната*class Bathroom(LivingRoom):  
 def \_\_init\_\_(self, room\_type, functionality, furniture, fixtures):  
 super().\_\_init\_\_(room\_type, functionality, furniture)  
 self.\_\_fixtures = fixtures *# Сантехніка* def add\_fixtures(self, \*items):  
 self.\_\_fixtures.extend(items)  
 print("Сантехніка успішно додана!")  
  
 *# Перевизначений метод для виведення інформації про ванну кімнату* def display\_info(self):  
 super().display\_info()  
 print(f"Сантехніка: {', '.join(self.\_\_fixtures)}\n")  
  
  
*# Клас Гараж*class Garage(Room):  
 def \_\_init\_\_(self, room\_type, functionality, vehicles):  
 super().\_\_init\_\_(room\_type, functionality)  
 self.\_\_vehicles = vehicles *# Транспортні засоби* def add\_vehicles(self, \*items):  
 self.\_\_vehicles.extend(items)  
 print("Транспортні засоби успішно додані!")  
  
 *# Перевизначений метод для виведення інформації про гараж* def display\_info(self):  
 super().display\_info()  
 print(f"Транспортні засоби: {', '.join(self.\_\_vehicles)}\n")  
  
  
*# Клас Спортивний зал*class Gym(Workspace):  
 def \_\_init\_\_(self, room\_type, functionality, equipment, machines):  
 super().\_\_init\_\_(room\_type, functionality, equipment)  
 self.\_\_machines = machines *# Тренажери та обладнання для фітнесу* def add\_machines(self, \*items):  
 self.\_\_machines.extend(items)  
 print("Тренажери та обладнання для фітнесу успішно додані!")  
  
 *# Перевизначений метод для виведення інформації про спортивний зал* def display\_info(self):  
 super().display\_info()  
 print(f"Тренажери та обладнання для фітнесу: {', '.join(self.\_\_machines)}\n")  
  
*# Створення об'єктів*kitchen = Kitchen("Kitchen", "Житлове", ["Стіл", "Стільці", "Плита"], ["Холодильник", "Мікрохвильовка"])  
bedroom = Bedroom("Bedroom", "Житлове", ["Ліжко", "Шафа", "Тумбочка"], "King Size")  
auditorium = Auditorium("Auditorium", "Робоче", ["Проектор", "Стільці", "Столи"], 100)  
office = Office("Office", "Робоче", ["Комп'ютер", "Столи", "Стільці"], 10)  
bathroom = Bathroom("Bathroom", "Санітарний", ["Диван", "Стіл", "Лампа"], ["Умивальник", "Туалет", "Душ"])  
garage = Garage("Garage", "Зберігання", ["Автомобіль", "Велосипед", "Інструменти"])  
gym = Gym("Gym", "Фітнес", ["Гантелі", "Мати для вправ"], ["Бігова доріжка", "Статичний велосипед"])  
  
*# Виведення інформації про об'єкти*kitchen.display\_info()  
bedroom.display\_info()  
auditorium.display\_info()  
office.display\_info()  
bathroom.display\_info()  
garage.display\_info()  
gym.display\_info()  
  
gym.add\_machines("Штанга")  
gym.display\_info()

Скріншот роботи програмного коду першої частини:

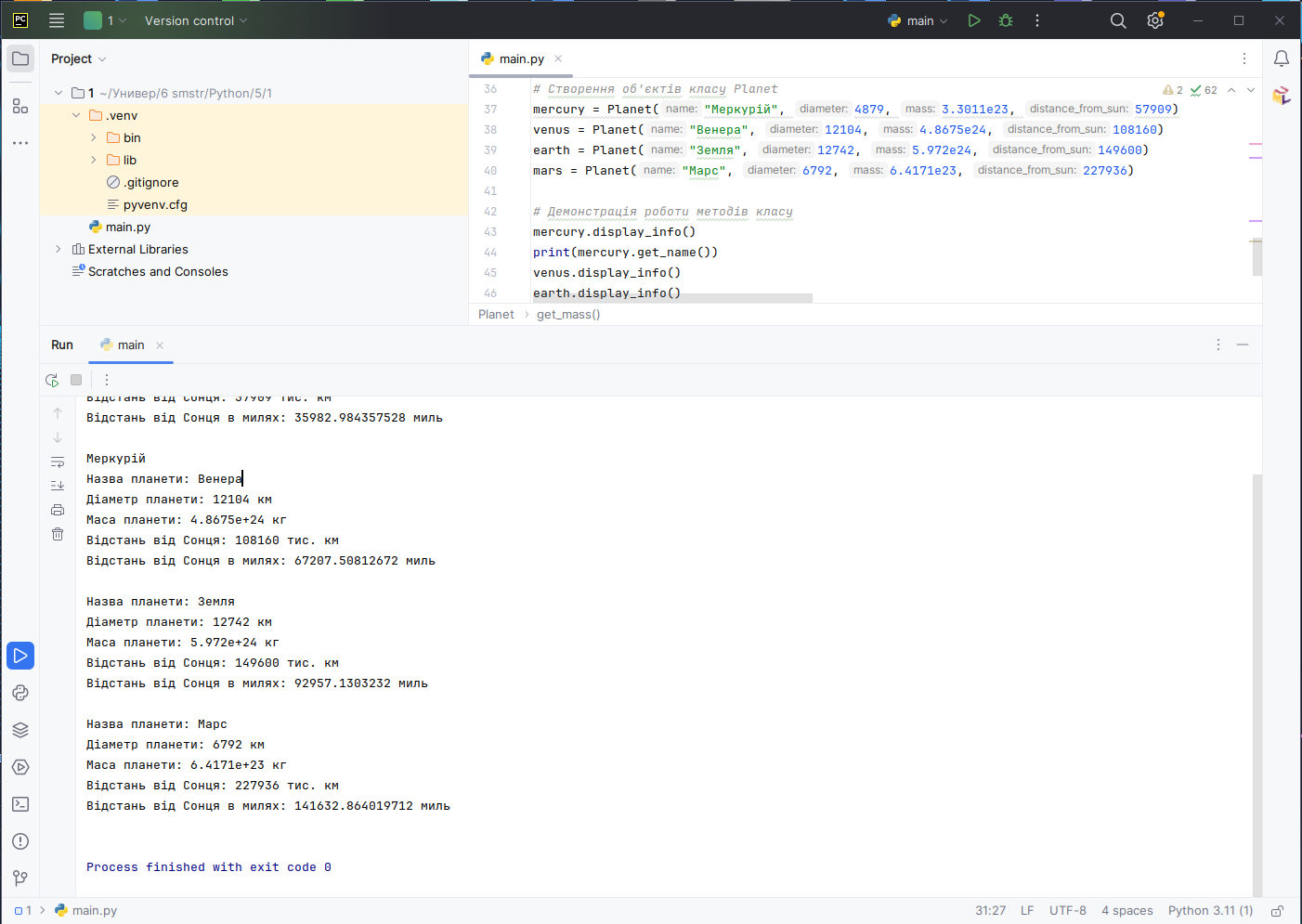


Рисунок 1 – робота першої частини

Скріншот роботи програмного коду другої частини:

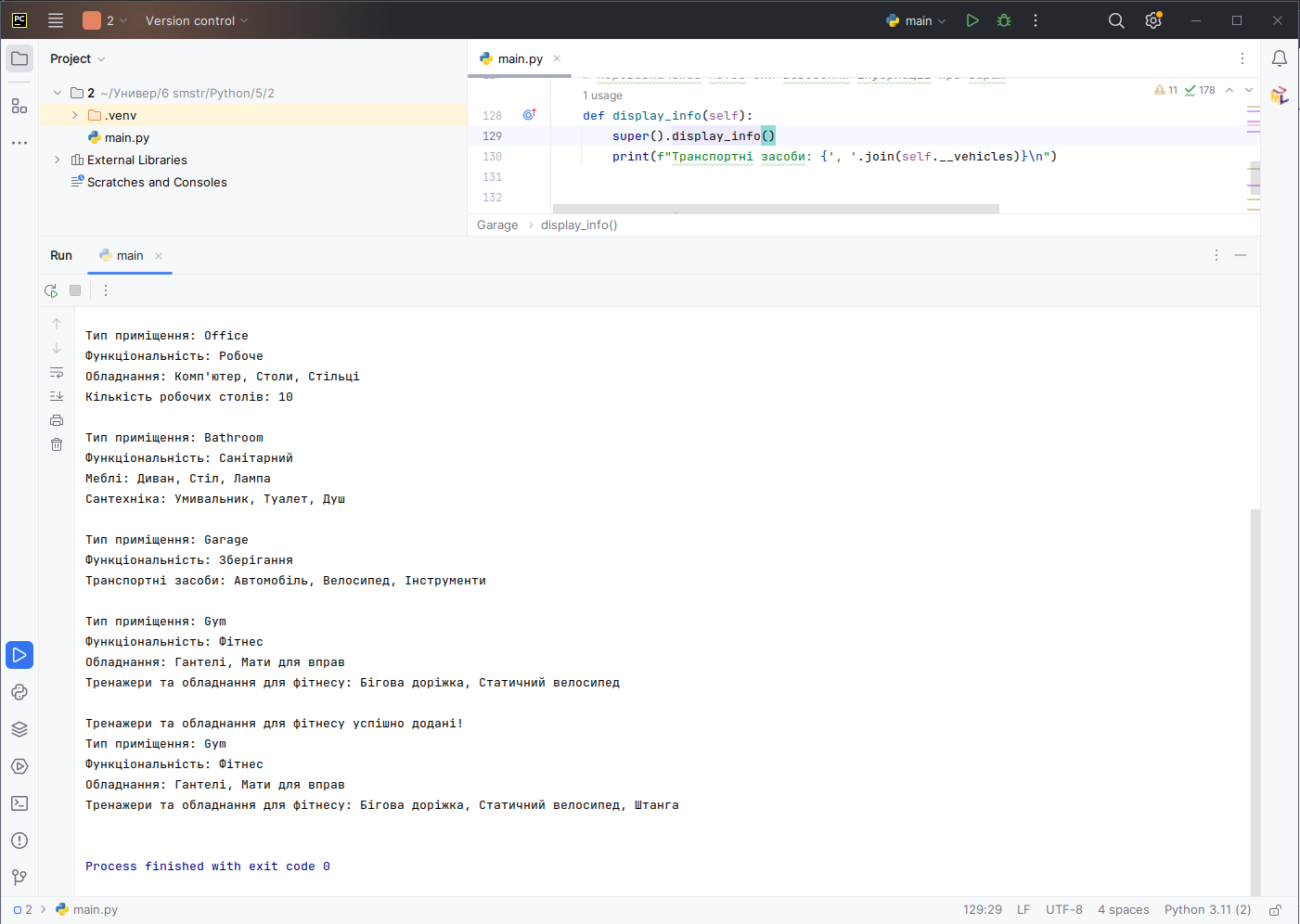


Рисунок 2 – робота другої частини

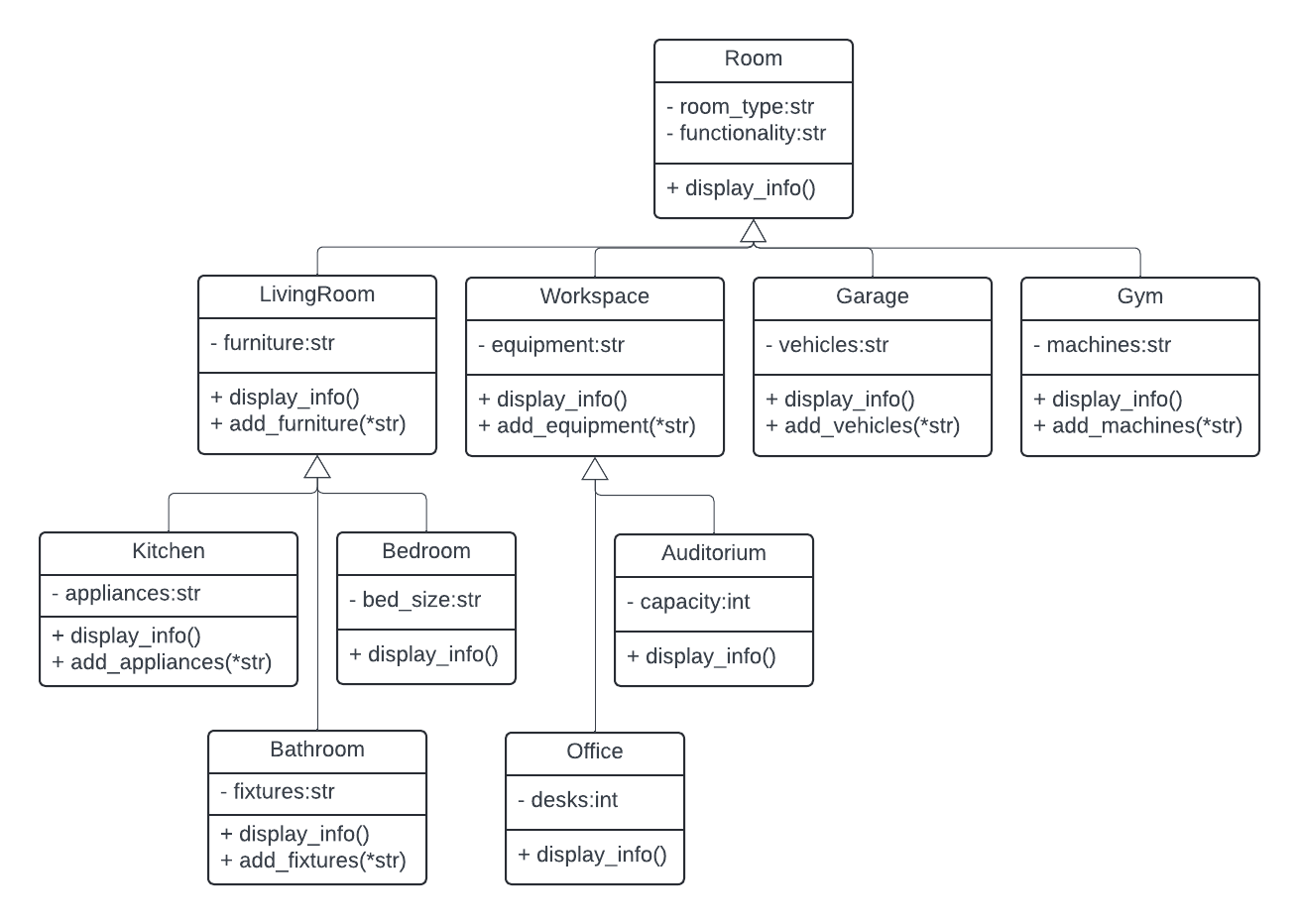


Рисунок 3 – Діаграма UML

Висновок:  
 Ми вивчити та засвоїти базові навички роботи з класами та їх спадкуванням у Python. Ознайомитися з поняттями конструктор, деструктор, метод класу, рівні доступу, перевизначення методу, ієрархія класів, доповнення методу та отримати практичні навички їх застосування.