Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних технологій, автоматики та метрології

Кафедра ЕОМ



**Звіт**

**До лабораторної роботи №9**

# З дисципліни: «Кросплатформні засоби програмування»

На тему «ОСНОВИ ОБ’ЄКТНО-ОРІЄНТОВАНОГО

ПРОГРАМУВАННЯ У PYTHON»

# Варіант №16

Виконав: ст. гр. КІ-305

Лавро В. В.

Прийняв:

Іванов Ю. С.

**Львів – 2023**

**Мета**

Оволодіти навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python

**Завдання(Варіант 16)**

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

**Варіант завдання:**

Базовий клас: Аудіоплеєр, похідний: Диктофон

Код програми:

MAIN.PY

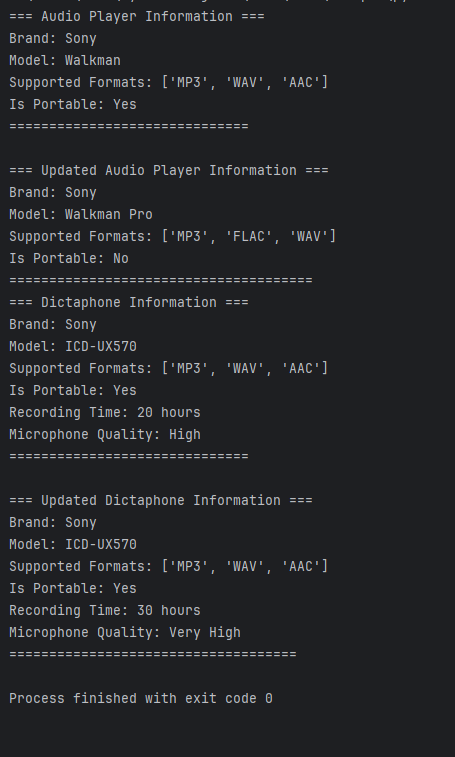
from Dictaphone import AudioPlayer, Dictaphone  
if \_\_name\_\_ == "\_\_main\_\_":  
 # Створення об'єкта AudioPlayer  
 my\_audio\_player = AudioPlayer(brand="Sony", model="Walkman", supported\_formats=["MP3", "WAV", "AAC"],  
 is\_portable=True)  
  
 # Виведення інформації про аудіоплеєр  
 print("=== Audio Player Information ===")  
 my\_audio\_player.get\_info()  
 print("==============================")  
  
 # Зміна деяких параметрів  
 my\_audio\_player.set\_model("Walkman Pro")  
 my\_audio\_player.set\_supported\_formats(["MP3", "FLAC", "WAV"])  
 my\_audio\_player.set\_portable\_status(False)  
  
 # Виведення оновленої інформації про аудіоплеєр  
 print("\n=== Updated Audio Player Information ===")  
 my\_audio\_player.get\_info()  
 print("======================================")  
  
 # Створення об'єкта Dictaphone  
 my\_dictaphone = Dictaphone(brand="Sony", model="ICD-UX570", supported\_formats=["MP3", "WAV", "AAC"],  
 is\_portable=True, recording\_time="20 hours", microphone\_quality="High")  
  
 # Виведення інформації про диктофон  
 print("=== Dictaphone Information ===")  
 my\_dictaphone.get\_info()  
 print("==============================")  
  
 # Зміна деяких параметрів  
 my\_dictaphone.set\_recording\_time("30 hours")  
 my\_dictaphone.set\_microphone\_quality("Very High")  
  
 # Виведення оновленої інформації про диктофон  
 print("\n=== Updated Dictaphone Information ===")  
 my\_dictaphone.get\_info()  
 print("====================================")

audioplayer.PY

class AudioPlayer:  
 def \_\_init\_\_(self, brand, model, supported\_formats, is\_portable):  
 self.\_brand = brand  
 self.\_model = model  
 self.\_supported\_formats = supported\_formats  
 self.\_is\_portable = is\_portable  
  
 # Getter methods  
 def get\_brand(self):  
 return self.\_brand  
  
 def get\_model(self):  
 return self.\_model  
  
 def get\_supported\_formats(self):  
 return self.\_supported\_formats  
  
 def is\_portable(self):  
 return self.\_is\_portable  
  
 # Setter methods  
 def set\_brand(self, brand):  
 self.\_brand = brand  
  
 def set\_model(self, model):  
 self.\_model = model  
  
 def set\_supported\_formats(self, supported\_formats):  
 self.\_supported\_formats = supported\_formats  
  
 def set\_portable\_status(self, is\_portable):  
 self.\_is\_portable = is\_portable  
  
 # Method to get information about the audio player  
 def get\_info(self):  
 print("Brand:", self.\_brand)  
 print("Model:", self.\_model)  
 print("Supported Formats:", self.\_supported\_formats)  
 print("Is Portable:", "Yes" if self.\_is\_portable else "No")

Dictaphone.PY

from AudioPlayer import AudioPlayer  
class Dictaphone(AudioPlayer):  
 def \_\_init\_\_(self, brand, model, supported\_formats, is\_portable, recording\_time, microphone\_quality):  
 # Викликаємо конструктор батьківського класу AudioPlayer  
 super().\_\_init\_\_(brand, model, supported\_formats, is\_portable)  
 self.\_recording\_time = recording\_time  
 self.\_microphone\_quality = microphone\_quality  
  
 # Додані методи для отримання значень часу запису та якості мікрофона  
 def get\_recording\_time(self):  
 return self.\_recording\_time  
  
 def get\_microphone\_quality(self):  
 return self.\_microphone\_quality  
  
 # Додані методи для встановлення значень часу запису та якості мікрофона  
 def set\_recording\_time(self, recording\_time):  
 self.\_recording\_time = recording\_time  
  
 def set\_microphone\_quality(self, microphone\_quality):  
 self.\_microphone\_quality = microphone\_quality  
  
 # Перевизначений метод для виведення інформації про диктофон  
 def get\_info(self):  
 # Викликаємо метод батьківського класу для отримання загальної інформації  
 super().get\_info()  
 print("Recording Time:", self.\_recording\_time)  
 print("Microphone Quality:", self.\_microphone\_quality)

****

**Висновок**

Я ознайомився з основними принципами мови програмування python та оволодів навиками застосування них. І оволодів навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python. Закріпив теорію на практиці.