

Міністерство освіти і науки України

Національний університет «Львівська Політехніка»

Кафедра ЕОМ



## **ЗВІТ**

з лабораторної роботи № 9

з дисципліни: «Кросплатформні засоби програмування»

на тему: «ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ  
У PYTHON»

**Виконав:**

студент гр. КІ-306

Приймак Ю.О.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

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

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.

## **Базовий клас: Аудіоплеєр**

## **Похідний клас: Диктофон**

### **Виконання:**

**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

    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

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

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):  
        super().__init__(brand, model, supported_formats, is_portable)  
        self._recording_time = recording_time  
        self._microphone_quality = microphone_quality  
  
    def get_recording_time(self):  
        return self._recording_time  
  
    def get_microphone_quality(self):  
        return self._microphone_quality  
  
    def set_recording_time(self, recording_time):  
        self._recording_time = recording_time  
  
    def set_microphone_quality(self, microphone_quality):  
        self._microphone_quality = microphone_quality  
  
    def get_info(self):  
        super().get_info()  
        print("Recording Time:", self._recording_time)  
        print("Microphone Quality:", self._microphone_quality)
```

## Результат:

```
=== Audio Player Information ===
Brand: Sony
Model: Walkman
Supported Formats: ['MP3', 'WAV', 'AAC']
Is Portable: Yes
=====

=== Updated Audio Player Information ===
Brand: Sony
Model: Walkman Pro
Supported Formats: ['MP3', 'FLAC', 'WAV']
Is Portable: No
=====

=== Dictaphone Information ===
Brand: Sony
Model: ICD-UX570
Supported Formats: ['MP3', 'WAV', 'AAC']
Is Portable: Yes
Recording Time: 20 hours
Microphone Quality: High
=====

=== Updated Dictaphone Information ===
Brand: Sony
Model: ICD-UX570
Supported Formats: ['MP3', 'WAV', 'AAC']
Is Portable: Yes
Recording Time: 30 hours
Microphone Quality: Very High
=====
```

**Висновок:** у ході данної лабораторної роботи я ознайомився реалізацією парадигм об'єктно-орієнтованого програмування використовуючи засоби мови Python.