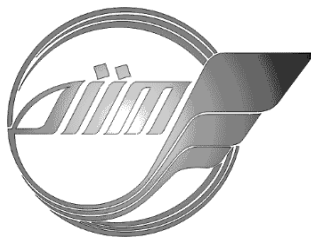


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп'ютерні інформаційні технології»

**Лабораторна робота №7**

**з дисципліни «Архітектура та проектування програмних засобів»**

**на тему: «Генерація коду»**

Виконав:  
студент гр.ПЗ1911  
Сафонов Д. Є.  
Прийняла:  
Куропятник О. С.

Дніпро, 2020

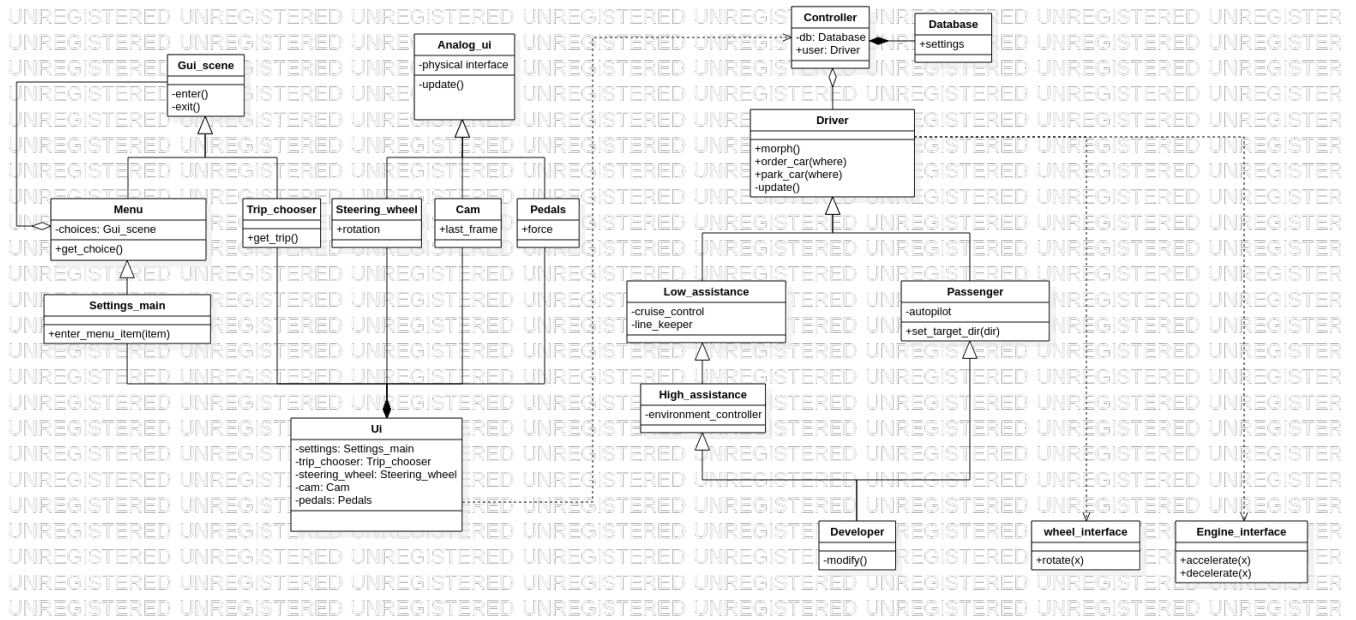
Тема. Генерація коду.

Мета. Ознайомитися із взаємозв'язком діаграм та етапами генерації коду. Отримати практичні навички генерації коду для UML-моделі.

### **Постановка задачі згідно завдання.**

Для проекту, який розроблявся протягом лабораторних робіт № 1 – 6, виконати генерацію коду. Визначити відповідності між кодом та елементами моделі: на прикладах коду продемонструвати всі застосовані специфікатори до елементів класу, відношення асоціації (композиції, агрегації), спадкування. Доповнити модель компонентами для створення, видалення об'єктів, доступу до елементів.

## Згенерований код з поясненнями.



*Рисунок 1 (Діаграма класів з ЛР2)*

“Menu.py”

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
```

```
from Gui_scene import Gui_scene
```

```
class Menu(Gui_scene):
    def __init__(self):
        self.choices = None

    def get_choice(self, ):
        pass
```

“Analog\_ui.py”

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
```

```
class Analog_ui:
    def __init__(self):
        self.physical_interface = None

    def update(self, ):
        pass
```

## “Controller.py”

```
#!/usr/bin/python
#-*- coding: utf-8 -*-

class Controller:
    def __init__(self):
        self.db = None
        self.user = None
```

## “Пояснення”

Через те, що в обраній мові не існує вказівників також відсутня й різниця між композицією та агрегацією.

Наслідування демонструється наступним синтаксисом:

```
class Ім'я_класу(Ім'я_базового_класу).
```

Залежність ніяк не вказана через те, що в обраній мові відсутні модифікатори доступу, тож завжди є можливість отримати доступ до потрібних методів та атрибутів будь яких класів та об'єктів.

## **Аналіз результатів та висновки щодо відповідності елементів моделі коду.**

Код відповідає моделі, загалом такий спосіб проєктування дуже зручний тому що набагато зручніше редагувати зв'язки між класами в одному файлі, ніж в багатьох, а результат той самий. Тож цей метод проєктування зберігає дуже багато часу, який можна витратити наприклад на реалізацію методів.