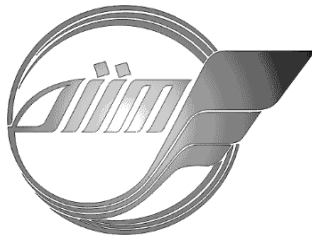


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



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

Кафедра «Комп'ютерні інформаційні технології»

Лабораторна робота №2

з дисципліни «Архітектура та проектування програмних засобів»

на тему: «Основи структурного моделювання. Діаграма класів.»

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

Дніпро, 2020

Тема. Основи структурного моделювання. Діаграма класів.

Мета. Вивчити: способи подання класів і базових зав'язків між ними; прийоми моделювання сутностей і зав'язків між ними. Отримати практичні навички з моделювання предметної області за допомогою діаграми класів.

Постановка задачі згідно загального та індивідуального завдання.

Виконати структурне моделювання ПЗ відповідно завданню лабораторної роботи №1(Автопілот для легкового автомобіля на дорогах загального користування).

Визначити сутності предметної області, їх атрибути, операції. Специфікувати класи сутностей у вигляді CRC-карток. Визначити та обґрунтувати зв'язки між класами. Побудувати діаграму класів.

Виконати аналіз застосованих проектних рішень.

Опис сутностей з визначенням їх атрибутів та операцій.

Ui:

```
private settings: Settings_main  
private trip_chooser: Trip_chooser  
private steering_wheel: Steering_wheel  
private cam: Cam  
private pedals: Pedals
```

Gui:

-

Scene extends Gui:

```
private enter()  
private exit()
```

Trip_chooser extends Scene:

-

Menu extends Scene:

```
private choices: menu_item  
public get_choice()
```

Settings_main extends Menu:

```
public enter_menu_item(item)
```

Menu_item extends Menu:

-

Analog_ui:

private physical interface

private update()

Steering_wheel extends Analog_ui:

-

Cam extends Analog_ui:

-

Pedals extends Analog_ui:

-

Data_transfer:

public event

public event_handler()

Controller:

private db: Database

public user: User

Database:

public settings

User:

public morph()

public order_car(where)

public park_car(where)

Passenger extends User:

```
public set_target_dir(dir)
private autopilot: Autopilot
```

High_assistance extends Low_assistance:

-

Low_assistance extends User:

```
private cruise_control: Cruise_control
private line_keeper: Line_keeper
```

Driver extends User:

-

Cruise_control:

-

Line_keeper:

-

Autopilot:

-

Специфікація класів сутностей у вигляді CRC-карток.

Ui

Базовий клас	Похідні класи (нащадки)
-	-
Обов'язки	Зв'язки
Об'єднати усі класи інтерфейсу користувача	data_transfer

Gui

Базовий клас	Похідні класи (нащадки)
-	Scene
Обов'язки	Зв'язки
Базовий клас графічного інтерфейсу	-

Scene

Базовий клас	Похідні класи (нащадки)
Gui	Menu, Trip_chooser
Обов'язки	Зв'язки
Базова сцена графічного інтерфейсу	-

Menu

Базовий клас	Похідні класи (нащадки)
Scene	Menu_item, Settings_main
Обов'язки	Зв'язки
Повертати обраний пункт меню	-

Menu_item

Базовий клас	Похідні класи (нащадки)
Menu	-
Обов'язки	Зв'язки
Базовий клас пункту меню	-

Trip_chooser

Базовий клас	Похідні класи (нащадки)
Scene	-
Обов'язки	Зв'язки
Вибір подорожі	-

Settings_main

Базовий клас	Похідні класи (нащадки)
Menu	-
Обов'язки	Зв'язки
Перехід за обраним пунктом меню	-

Analog_ui

Базовий клас	Похідні класи (нащадки)
-	Steering_wheel, Cam, Pedals
Обов'язки	Зв'язки
Повертати релевантне значення фізичного інтерфейсу	-

Steering_wheel

Базовий клас	Похідні класи (нащадки)
Analog_ui	-
Обов'язки	Зв'язки
Повертати релевантне значення повороту руля	-

Cam

Базовий клас	Похідні класи (нащадки)
Analog_ui	-
Обов'язки	Зв'язки
Повертати релевантну картинку з камер	-

Pedals

Базовий клас	Похідні класи (нащадки)
Analog_ui	-
Обов'язки	Зв'язки
Повертати релевантну силу натиску на педалі	-

Data_transfer

Базовий клас	Похідні класи (нащадки)
-	-
Обов'язки	Зв'язки
Обробка подій	-

Controller

Базовий клас	Похідні класи (нащадки)
-	-
Обов'язки	Зв'язки
Зв'язок між User, Ui, Database	Data_transfer

User

Базовий клас	Похідні класи (нащадки)
-	Driver, Low_assistance, Passenger
Обов'язки	Зв'язки
Представлення типу користувача	-

Driver

Базовий клас	Похідні класи (нащадки)
User	Developer
Обов'язки	Зв'язки
Дозволити водію повністю контролювати авто	-

Low_assistance

Базовий клас	Похідні класи (нащадки)
User	High_assistance
Обов'язки	Зв'язки
Дозволити водію контролювати авто, та допомагати у простих ситуаціях	-

High_assistance

Базовий клас	Похідні класи (нащадки)
Low_assistance	Developer
Обов'язки	Зв'язки
Дозволити водію контролювати авто, та допомагати у простих ситуаціях, та забирати контроль у критичних	-

Passenger

Базовий клас	Похідні класи (нащадки)
User	Developer
Обов'язки	Зв'язки
Дозволити пасажиру обрати кінцеву зупинку	-

Developer

Базовий клас	Похідні класи (нащадки)
High_assistance, Driver, Passenger	-
Обов'язки	Зв'язки
Дозволити розробнику модифікувати та тестувати систему	-

Database

Базовий клас	Похідні класи (нащадки)
-	-
Обов'язки	Зв'язки
Зберігати налаштування	-

Опис та обґрунтування зв'язків між класами.

Data_transfer обробляє подію з Ui із допомогою методів Controller.user

Діаграма класів.

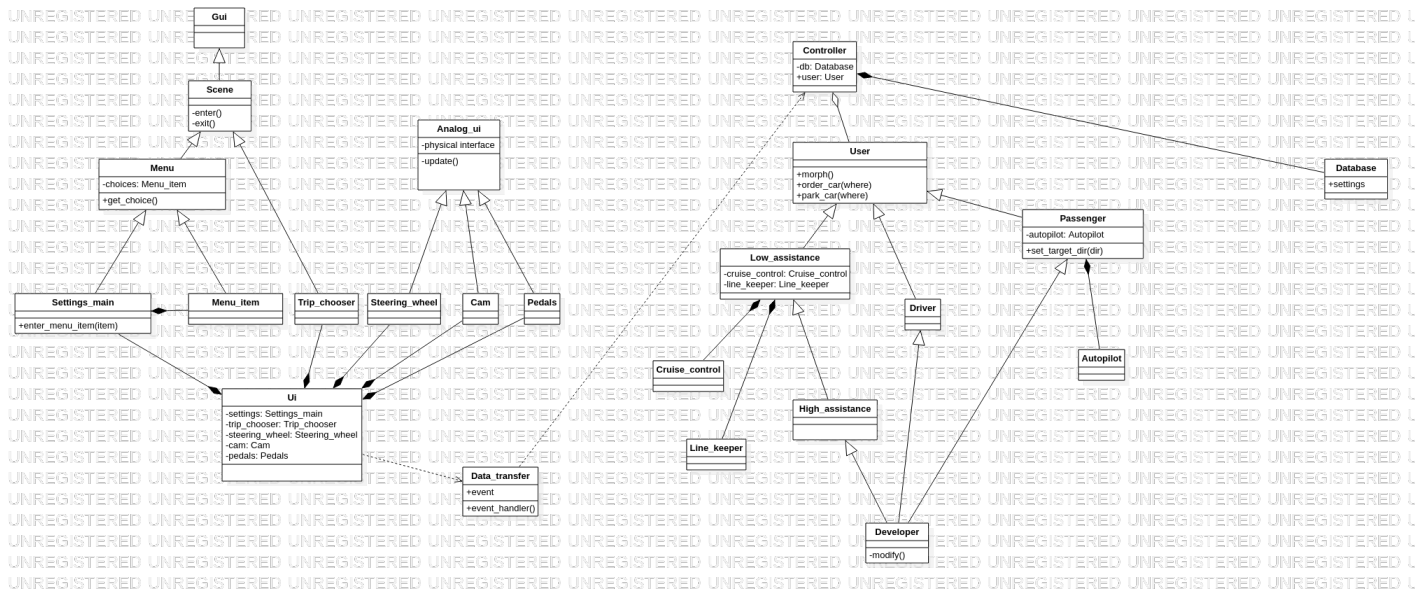


Рисунок 1

Аналіз результатів та висновки щодо переваг та недоліків у використанні засобів UML для специфікації ПЗ, а також відносно особливостей прийнятих проектних рішень.

Одна з найбільших переваг у використанні UML — дуже докладний розбір архітектури ПЗ. Найбільший недолік — час розробки. Але як і завжди це компроміс. Також до недоліків можна віднести те, що потрібно вчити нові правила. Щодо особливостей прийнятих рішень: я розділив програму на три частини: Інтерфейс користувача, контролер та маленька база даних для збереження налаштувань користувача. Усі події у інтерфейсі оброблюються через користувача контролера.