Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення

**Звіт**

Про виконання лабораторної роботи №6

**На тему:**

«Опис предметної області з використанням UML»

з дисципліни

«Вступ до Інженерії Програмного Забезпечення»

**Лектор:**

Доц. каф. ПЗ

Левус Є. В.

**Виконав:**

ст. гр. ПЗ-11

Солтисюк Д. А.

**Прийняла:**

Доц. каф. ПЗ

Левус Є. В.

« \_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑ = \_\_\_\_\_ \_\_\_\_\_\_\_\_\_ .

Львів – 2022

**Тема:** Опис предметної області з використанням UML.

**Мета:** Навчитися створювати об’єктну модель програмної системи.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

**Мовою C++ написати фрагмент коду для агрегації та композиції (в чому відмінність зазначити в комментарі)**

#include <string>

**using** std::string;

class Plane {

public:

int id;

string name;

time\_t departs\_at;

time\_t arrives\_at;

};

class Human {

public:

string name;

int age;

};

class Passenger : Human {

public:

int planes\_taken;

void ask\_for\_plane();

Plane \*planes;

};

class Income {

public:

float amount;

float calculate\_income();

};

class System {

public:

Human \*\*people; *// array of pointers - aggregation*

Plane \*\*planes; *// array of pointers - aggregation*

Income income; *// object - composition*

string location;

System(Human \*\*people, Plane \*\*planes) {

**this**->people = people;

**this**->planes = planes;

**this**->income = Income();

}

};

В даному випадку клас “System” включає в себе інші класи (Human, Plane). Вони служать будівельними блоками для даного класу (проте не є наслідуванням) та є самостійними сутностями. Така асоціація називається агрегацією і в мові C++ вона відображається вказівником.

В свою чергу, відношення між класом “System” та “Income” називається композицією, адже він також є складовою частиною, проте не може існувати самостійно, у C++ це є звичайний екземпляр класу.

**ПОСТАНОВКА ЗАВДАННЯ**

**Варіант №24**

Провести аналіз предметної області «**RyanAir**».

1. Скласти словник предметної області
2. Побудувати UML-діаграму класів на концептуальному рівні. Зобразити коментарі на схемі. Вказати відношення між сутностями із обов’язковим зазначенням їх характеристик.
3. Побудувати UML-діаграму конкретних класів на рівні реалізації. Чітко вказати усі поля та методи класів з відповідними модифікаторами доступу, а також усі необхідні відношення між класами.

**ОТРИМАНІ РЕЗУЛЬТАТИ**

**Словник предметної області**:

1. **Адміністратор** – працівник-адміністратор, має можливість виконувати потрібну роботу в застосунку - work() та додати до системи новго Оператора - add\_operator(). Спецілізована сутність **Працівник**
2. **Пасажир** – людина, яка має має наступні властивості: к-сть польотів та літаки, на яких літав до цього. Має можливість купити переліт (включає в себе оплату). Спеціалізовна сутність **Людина**
3. **Оператор** – працівник системи, має можливість надати політ **Відвідувачу** та працювати (work() - верифікувати квитки, тощо). Спеціалізовна сутність **Працівник**.
4. **Пілот** – працівник системи, який має можливість керувати **Літаком**. Спеціалізовна сутність **Працівник**.
5. **Людина** – абстрактна сутність, яка має імя та вік. Привязано до сутності **Система**
6. **Працівник** – абстрактна сутність, яка має заробітну плату та досвід роботи, а також абстрактний метод work(). Спеціалізовна сутність **Людина**
7. **Прибуток** – грошовий прибуток **Системи** (частина-ціле, 1 до 1 композиція), має властивість **amount**. Має **залежність** від к-сті відвідувачів.
8. **Система** – головна сутність. Має властивість **location** (місце знаходження). Та зберігає в собі всіх **людей** та **літаки**, які з нею повязані.
9. **Літак** – сутність (частина-ціле для **Системи** - агрегація 1 to 1..\*). Має наступні властивості: час відправки та прильоту, а також унікальний номер польоту.

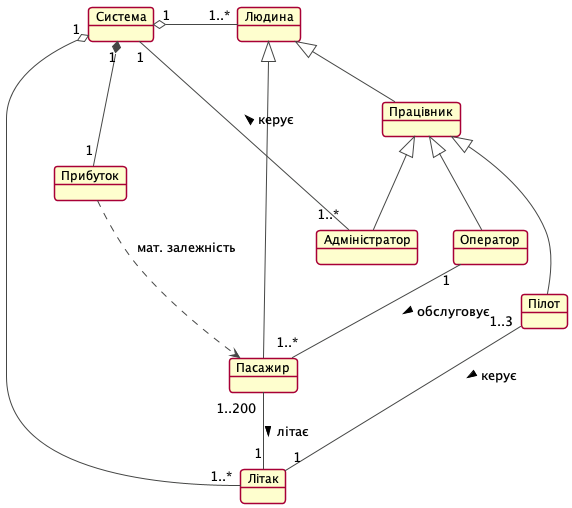
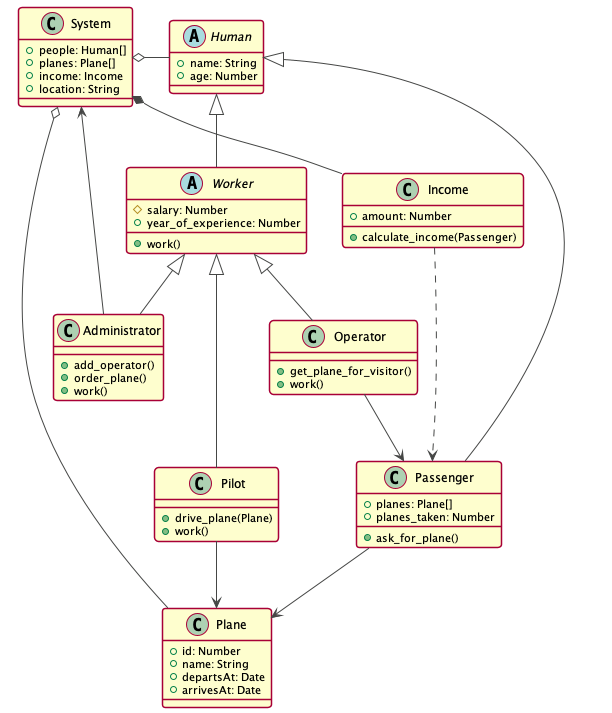


Рис. 1. UML-діаграма на концептуальному рівні

Рис. 2. UML-діаграма на рівні реалізації.

**ВИСНОВОК**

Виконуючи лабораторну роботу №6, я навчився описувати предметну сутність за допомогою мови UML на різних рівнях (концептуальному, реалізації). Побудував UML-діаграми та словник предметної області до інформаційної системи «RyanAir».