**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**імені ІГОРЯ СІКОРСЬКОГО”**

**ЗВІТ**

**з лабораторної роботи №5**

**з навчальної дисципліни “ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ”**

**Тема:**

Побудова діаграми прецедентів інформаційної системи, діаграми класів, діаграми кооперації (взаємодії), діаграми послідовності засобами СASE системи Rational Rose

**Варіант 20**

**Виконав студент групи ТР–12**

Руденко Владислав\_\_\_\_\_\_\_\_\_

Лабораторну роботу захищено

з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Київ 2024**

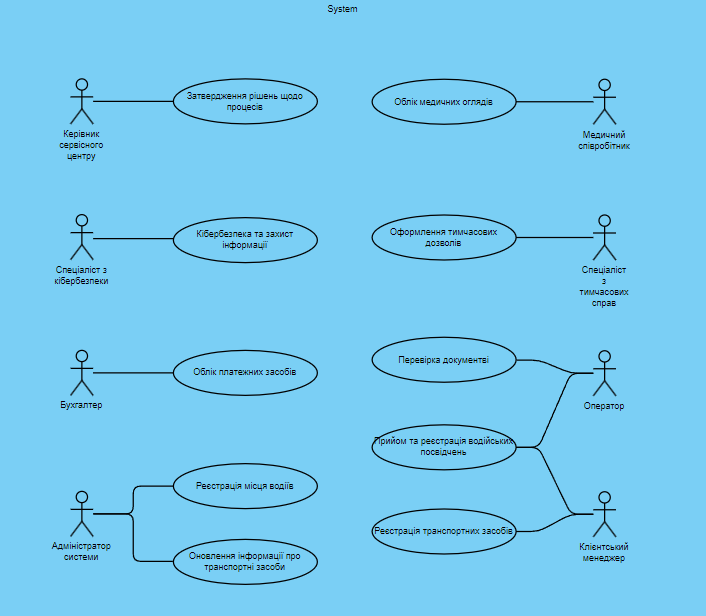
**Завдання:**

навчитися будувати діаграму прецедентів та діаграму класів при проектуванні інформаційної системи.

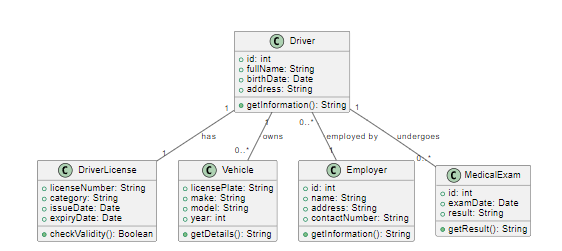
**Тема:** ІС обліку водіїв у сервісному центрі поліції

**Виконання:**

1. Розробити діаграму прецедентів інформаційної системи



1. Розробити діаграму класів



1. Згенерувати в Rational Rose код програмної системи С++ опису класів.

#include <iostream>

#include <string>

#include <vector>

using namespace std;

// Клас Driver

class Driver {

public:

int id;

string fullName;

string birthDate;

string address;

Driver(int id, string fullName, string birthDate, string address)

: id(id), fullName(fullName), birthDate(birthDate), address(address) {}

string getInformation() {

return "Driver: " + fullName + ", Address: " + address;

}

};

// Клас DriverLicense

class DriverLicense {

public:

string licenseNumber;

string category;

string issueDate;

string expiryDate;

DriverLicense(string licenseNumber, string category, string issueDate, string expiryDate)

: licenseNumber(licenseNumber), category(category), issueDate(issueDate), expiryDate(expiryDate) {}

bool checkValidity() {

// Простий приклад перевірки валідності (умовно)

return expiryDate > "2024-12-01";

}

};

// Клас Vehicle

class Vehicle {

public:

string licensePlate;

string make;

string model;

int year;

Vehicle(string licensePlate, string make, string model, int year)

: licensePlate(licensePlate), make(make), model(model), year(year) {}

string getDetails() {

return "Vehicle: " + make + " " + model + " (" + to\_string(year) + ")";

}

};

// Клас Employer

class Employer {

public:

int id;

string name;

string address;

string contactNumber;

Employer(int id, string name, string address, string contactNumber)

: id(id), name(name), address(address), contactNumber(contactNumber) {}

string getInformation() {

return "Employer: " + name + ", Contact: " + contactNumber;

}

};

// Клас MedicalExam

class MedicalExam {

public:

int id;

string examDate;

string result;

MedicalExam(int id, string examDate, string result)

: id(id), examDate(examDate), result(result) {}

string getResult() {

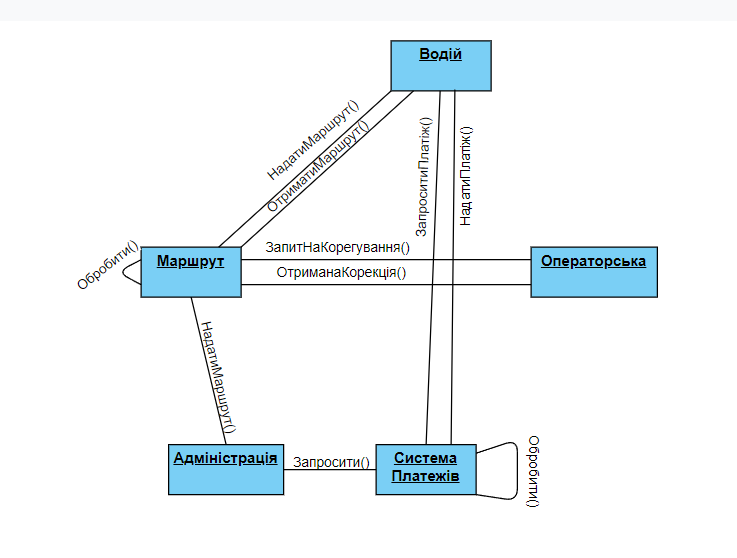
return "Exam Date: " + examDate + ", Result: " + result;

}

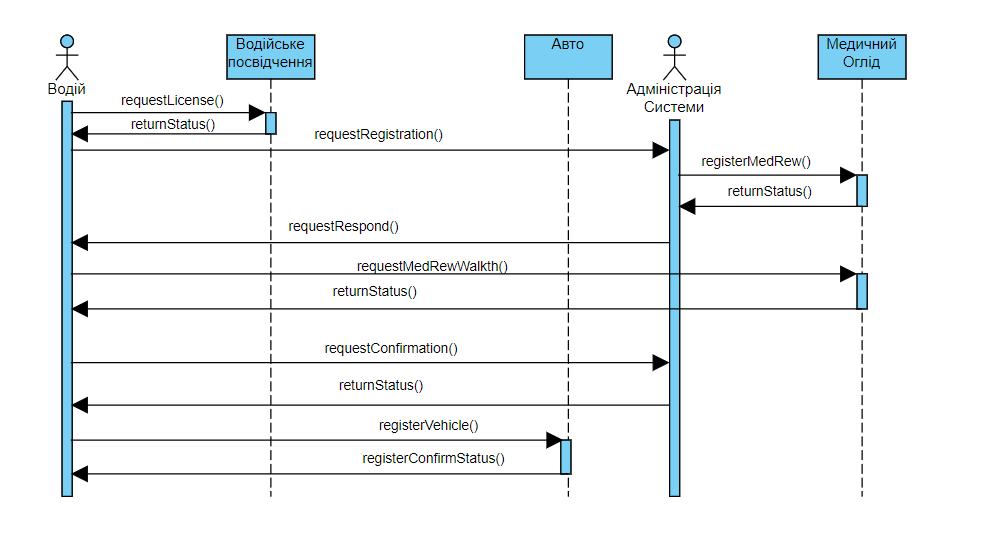
};

}

1. Розробити діаграму кооперації (взаємодії)



1. Розробити діаграму послідовності.



**Висновок:**

У результаті виконання роботи було освоєно побудову основних UML-діаграм, зокрема діаграму прецедентів, класів, кооперації та послідовності. Це дало змогу детально проаналізувати взаємодію компонентів проєктованої системи. Згенерований код підтверджує коректність розробленої моделі, а використання Rational Rose спростило створення й візуалізацію структури системи. Отримані навички сприяють підвищенню якості програмних рішень та їх моделюванню.