Міністерство освіти і науки України

Харківський національний університет ім. В. Н. Каразіна

Факультет комп’ютерних наук

**Курсова робота**

з дисципліни «Об’єктно-орієнтоване програмування»

Тема: «Розробка прототипу програмної системи управління реєстром автомобілів (модифікація 1.1)»

|  |  |
| --- | --- |
| Оцінка \_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Члени комісії:  \_\_\_\_\_\_\_\_\_\_\_ Поклонський Є.В.  \_\_\_\_\_\_\_\_\_\_\_ Нагорний К.А.  \_\_\_\_\_\_\_\_\_\_\_ Богучарський С.І. | Виконала:  студентка 2 курсу, групи КС- 21 |
| Спеціальності:  122 «Комп’ютерні науки»  Дібцева Анна Миколаївна |
|  |

Харків – 2019

# ЗМІСТ

[ЗМІСТ 2](#_Toc27532279)

[ВСТУП 3](#_Toc27532280)

[РОЗДІЛ 1 КОНЦЕПЦІЯ 4](#_Toc27532281)

[1.1 Загальна характеристика продукту 4](#_Toc27532282)

[1.2 Основні можливості та обмеження 4](#_Toc27532283)

[РОЗДІЛ 2 АНАЛІЗ 7](#_Toc27532284)

[2.1 Склад системи 7](#_Toc27532285)

[2.2 Вимоги до даних (Використовувані файли) 7](#_Toc27532286)

[2.3 Інші вимоги (Основні вимоги програмної системи) 7](#_Toc27532287)

[РОЗДІЛ 3 ПРОЕКТУВАННЯ 8](#_Toc27532288)

[3.1 Опис предметної області 8](#_Toc27532289)

[3.2 Класи предметної області 9](#_Toc27532290)

[3.3 UML-діаграма класів 9](#_Toc27532291)

[РОЗДІЛ 4 ТЕСТУВАННЯ 10](#_Toc27532292)

[РОЗДІЛ 5 ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАЧА 11](#_Toc27532293)

[ВИСНОВКИ 12](#_Toc27532294)

[ДОДАТОК А ФОРМАТ ФАЙЛІВ 13](#_Toc27532295)

[ДОДАТОК Б ФРАГМЕНТИ ВИХІДНОГО КОДУ 14](#_Toc27532296)

# ВСТУП

В цій роботі опрацьовується прототип програмної системи (ПС) управління автомобільним салоном, яка надає можливість користувачу маніпулювати реєстром автомобілів. Ця тема була обрана з розрахунку, що в результаті створення ПС ми будемо мати просту систему зберігання та обробки даних конкретної галузі користування, тож це як невеликий тренувальний проект, пов’язаний з нашою майбутньою професією.

Розробка програмної системи проходила в IDE Eclipse, де знаходиться інтерфейс користувача – консоль, де реалізовано меню для зручнішого користування функціями реєстру.

Створення та форматування пояснювальної записки до курсової роботи проходило в текстовому редакторі MS Word, бо він має зручний інтерфейс для редагування тексту.

Пояснювальна записка складається з таких основних частин: концепція програмної системи, аналіз поставленої задачі та вихідного формату файлу, загальний опис архітектури та класів предметної області (автомобільного салону), наведення результатів тестування окремих класів та програми загалом, інструкція для користувача.

Треба звернути увагу на основні можливості та обмеження програми (див. 1.2) , та уважно ознайомитись з інструкцією для використання кінцевого продукту.

# РОЗДІЛ 1 КОНЦЕПЦІЯ

## 1.1 Загальна характеристика продукту

Мета моєї програми це розробка прототипу програмної системи на базі основних принципів об’єктно-орієнтованого дизайну та сучасних практик програмування із використанням об’єктно-орієнтованої мови програмування Java. Її призначення це забезпечити взаємодію між автосалоном, клієнтами та реєстром автомобілів.

## 1.2 Основні можливості та обмеження

Основні можливості системи:

1. Додати транспорт до реєстру;

2. Видалити транспортний засіб із реєстру;

3. Отримати список усіх транспортних засобів, що містяться у реєстрі;

4. Отримати список усіх авто із реєстру;

5. Отримати список усіх мотоциклів із реєстру;

6. Продати обране авто покупцеві магазину, при цьому транспортний засіб видаляється із реєстру та записується ціна продажу;

7. Роздрукувати суму продажів за «день»

8. Зберегти реєстр до сховища (текстовий файл);

9. Отримати реєстр із сховища (текстового файлу).

Основні принципи та конструкції, що мають бути реалізовані:

1. Агрегація. Магазин – сутність, яка містить колекцію видань;

2. Наслідування. Автомобіль та Мотоцикл повинні мати базову абстракцію Транспортний Засіб (Vehicle);

3. Поліморфізм. Кожний Транспортний Засіб має поліморфний метод getType, який використовується під час вибірки та відображення тварини за типом;

4. Типи транспортних засобів є фіксованою множиною та мають належати до перерахування (Enumeration): автомобіль, мотоцикл;

5. Асоціативна сутність Продаж (Транспортний Засіб, Покупець, Ціна), для реалізації продажу транспортного засобу покупцеві магазину.

Основні можливості системи:

1. Додати транспортний засіб до реєстру;

2. Видалити транспортний засіб із реєстру;

3. Отримати список усіх транспортних засобів, що містяться у реєстрі;

4. Отримати список усіх транспортних засобів, тип палива яких відповідає заданому;

5. Продати обране авто покупцеві магазину, при цьому транспортний засіб видаляється із реєстру та записується ціна продажу;

6. Роздрукувати суму продажів за «день»

7. Зберегти реєстр до сховища (текстовий файл);

8. Отримати реєстр із сховища (текстового файлу).

Основні принципи та конструкції, що мають бути реалізовані:

1. Агрегація. Магазин – сутність, яка містить колекцію видань.
2. Наслідування. Автомобіль та Мотоцикл повинні мати базову абстракцію Транспортний Засіб (Vehicle);
3. Поліморфізм. Кожний Транспортний Засіб має поліморфний метод getType, який використовується під час вибірки та відображення тварини за типом (п.4, п.5 основних можливостей системи).
4. Типи палива транспортних засобів є фіксованою множиною та мають належати до перерахування (Enumeration): бензин, дизель, електрика, ядерний реактор.
5. Асоціативна сутність Продаж (Транспортний Засіб, Покупець, Ціна), для реалізації продажу транспортного засобу покупцеві магазину.

# РОЗДІЛ 2 АНАЛІЗ

## 2.1 Склад системи

У комплект поставки входять файли Database.jar та database.csv.

Database.jar – Java-архів. Основне завдання цього архіву - зберігати файли з класами. А database.csv – текстовий файл, який виступає в ролі реєстру транспорту.

## 2.2 Вимоги до вхідних даних

Програма приймає на вхід текстовий файл “database.csv” як реєстр для транспорту, звідки вичитує інформацію (при запуску програми) та куди записує інформацію (при завершенні програми). У цьому файлі зберігається інформація про транспорт різних типів. Типи, що зберігаються: Авто (AVTO) та мотоцикл (MOTO).

Приклад форматування цього файлу надано у додатку А

## 2.3 Основні вимоги програмної системи

Розробити прототип програмної системи (ПС) управління автомобільним салоном, яка надає можливість користувачу маніпулювати реєстром автомобілів.

Інтерфейс користувача – консоль у яку має бути виведений список усіх можливих команд.

Програмна система має працювати до введення команди виходу із системи.

Перед завершенням роботи програмної системи, реєстр має бути збережений у постійному сховищі (persistence storage), у форматі CSV (coma separated values). Описати відповідну структуру класів розробленого програмного рішення в нотації UML у термінах діаграми класів (class diagram).

# РОЗДІЛ 3 ПРОЕКТУВАННЯ

## 3.1 Опис предметної області

Предметна область цієї програми: автомобільний салон.

Основні сутності предметної області:

1. Магазин. Має назву, адресу, містить реєстр авто;
2. Автомобіль – транспортний засіб, який має назву, рік виробництва. Транспортний засіб має відповідний тип палива, та ємність паливного сховища. Авто в магазині характеризується ціною, за яку воно може бути куплена.
3. Мотоцикл – транспортний засіб, який має назву, рік виробництва. Транспортний засіб має відповідний тип палива, та ємність паливного сховища. Авто в магазині характеризується ціною, за яку воно може бути куплена.
4. Модель – характеризується назвою та номером.
5. Покупець магазину – людина, яка має ім’я та прізвище, рік народження та ідентифікаційний код.

Основні сутності предметної області:

1. Магазин. Має назву, адресу, містить реєстр авто;
2. Автомобіль – транспортний засіб, який має назву, рік виробництва, модель, та виробника. Транспортний засіб має відповідний тип палива, та ємність паливного сховища. Авто в магазині характеризується ціною, за яку воно може бути куплена.
3. Мотоцикл – транспортний засіб, який має назву, рік виробництва, модель, та виробника. Транспортний засіб має відповідний тип палива, та ємність паливного сховища. Авто в магазині характеризується ціною, за яку воно може бути куплена.
4. Модель – характеризується назвою та номером.
5. Покупець магазину – людина, яка має ім’я та прізвище, рік народження та ідентифікаційний код.

## 3.2 Класи предметної області

Програма містить такі класи:

1. Main;
2. Shop;
3. Client;
4. Vehicle;
5. Type;
6. Model;
7. Avto;
8. Moto.

## 3.3 UML-діаграма класів

## 3.4 Опис застосованих класів

Main – клас, де розташоване головне меню. Користувач має змогу додати транспортний засіб до реєстру, видалити транспортний засіб із реєстру, роздрукувати перелік усіх транспортних засобів, роздрукувати перелік усіх авто, роздрукувати перелік усіх мотоциклів, продати обране авто покупцеві магазину, роздрукувати суму продажів за «день».

Shop – клас, котрий викликає методи для рішення задач, які користувач обирає у класі Main.

Client – клас, що відповідає за клієнтів, котрі можуть купувати транспорт.

Vehicle – клас, що містить основні характеристики та властивості (поля та методи) для транспорту.

Type – перерахування, що допомагає вводити тип транспорту, де авто/мотоцикл є фіксованою множиною.

Model – клас, що містить модель та номер транспорту.

Avto – клас, що є спадкоємцем класу Vehicle, він приймає характеристики, котрі властиві автомобілю.

Moto - клас, що є спадкоємцем класу Vehicle, він приймає характеристики, котрі властиві молоциклу.

# РОЗДІЛ 4 ТЕСТУВАННЯ

# РОЗДІЛ 5 ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАЧА

# ВИСНОВКИ

# ДОДАТОК А ФОРМАТ ФАЙЛІВ

# ДОДАТОК Б ФРАГМЕНТИ ВИХІДНОГО КОДУ