**Практична робота №6.**

Наслідування. Абстрактні класи. Інтерфейси.

**Мета завдання**

Навчитися працювати з класами, які утворюють ієрархію наслідування. Навчитися описувати сутності програми за допомогою графічних нотацій

**Постановка завдання**

Ознайомитися з теоретичним матеріалом лекції та створити проект, що відповідає поставленому завданню.

**Завдання**

**Частина 1**

Створити ієрархію класів засобів пересування і сервер, який керує рухом усіх транспортних засобів, що виїхали в місто. Базовим класом є абстрактний клас (абстрактний клас - це клас об'єкт якого не можна створити) **Vehicle**.

*public abstract class Vehicle {*

*…*

*}*

з одним абстрактним методом (абстрактний метод - це метод, який обов'язково повинен бути перевизначений в класі нащадку) з параметром **int id**.

*public abstract void move(int id);*

Також є чотири види засобів пересування:

- автомобіль (**Car**);

- велосипед (**Bicycle**);

- роликові ковзани (**RollerSkates**);

- джампери (**Jumpers**).

Кожний засіб пересування - це окремий клас, який успадковується від базового класу Vehicle і кожен з них перевизначає метод move () по-своєму, наприклад:

*public class Bicycle extends Vehicle {*

*public void move(int id) {*

*System.out.println("Велосипед №" + id + "виїхав.");*

*}*

*}*

Клас Server може мати наступний вигляд

*public class Server {*

*// кількість транспортних засобів (ТЗ) кожного виду, що виїхали в місто*

*private int[] countVehicles;*

*// загальна кількість ТЗ, що виїхали в місто*

*private int commonCountVehicles;*

*// обмеження на кількість ТЗ, що одночасно виїхали в місто*

*private int limitVehicle;*

*// масив, що зберігає ТЗ будь-якого виду*

*private Vehicle[] vehicles = new Vehicle[limitVehicle];*

*// конструктор класу, параметр size – кількість видів ТЗ*

*public Server (int size) {*

*countVehicles = new int[size];*

*}*

*// повертає обмеження на кількість ТЗ, що одночасно виїхали в місто*

*public int getLimitVehicle () {*

*return limitVehicle;*

*}*

*// повертає загальну кількість ТЗ, що виїхали в місто*

*public int getCurrentCountVehicle () {*

*return commonCountVehicles;*

*}*

*// додає ТЗ до віртуальної бази*

*public void addTransport (int numberVehicles, Vehicle vehicle) {*

*countVehicles[numberVehicles]++;*

*commonCountVehicles ++;*

*vehicles [commonCountVehicles] = vehicle;*

*vehicles.move(countVehicles[numberVehicles]);*

*}*

*}*

У функції **main ()** тестового класу в циклі створювати об'єкти нових транспортних засобів (випадковим чином і з використанням оператора switch) до того моменту, поки не буде перевищено ліміт загальної кількості транспортних засобів. Після цього роздрукувати загальну кількість кожного транспортного засобу, що виїхав на дорогу.

**Частина 2**

За основу взяти попереднє завдання та доповнити його наступними класами. Замість абстрактного класу Vehicle використовувати інтерфейс Run.

Створити інтерфейс Run, для опису транспорту. Інтерфейс повинен містити в собі такі методи:

* метод виведення імені транспортного засобу;
* метод отримання максимальної швидкості пересування транспорту;
* метод отримання маси транспорту;
* метод move (), який дає команду рухатися.

Створити абстрактний клас Machine, який реалізує інтерфейс Run. В даному класі описати відповідні поля (назва, маса, максимальна швидкість пересування, кількість пасажирів), реалізувати конструктор для ініціалізації відповідних полів і перевизначити методи інтерфейсу Run.

Створити клас Bus, що розширює абстрактний клас Machine. Визначити в ньому конструктор, поле кількість пасажирів і метод для повернення кількості пасажирів.

Створити абстрактний клас Traсtor, який розширює абстрактний клас Machine. Визначити в ньому поле String type (колісний або гусеничний), конструктор Traсtor (String name, int speed, int weight, String type) для ініціалізації полів і метод який повертає тип трактора.

Створити два прямих нащадка TraсtorWheeled і TraсtorCrawler від класу Traсtor, в яких визначити конструктори.

У функції **main ()** тестового класу в циклі створювати об'єкти нових транспортних засобів (випадковим чином і з використанням оператора switch) до того моменту, поки не буде перевищено ліміт загальної кількості транспортних засобів. Після цього роздрукувати загальну кількість кожного транспортного засобу, що виїхав на дорогу.

**Частина 3**

Для частини 1 та частини 2 відповідно створити UML – діаграми класів, які опишуть. Скористатися можна як:

* Плагіном для Ultimate Intelij IDEA;
* Microsoft Visio;
* online – засоби;
* Paint тощо.

Для довідки: <https://www.tutorialspoint.com/uml/uml_class_diagram.htm>