**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені І.І. МЕЧНИКОВА**

Факультет математики, фізики та інформаційних технологій

Кафедра комп’ютерних систем та технологій

**ІНДИВІДУАЛЬНА РОБОТА**

з дисципліни «Програмування (Об’єктні Технології)»

**Створення ієрархії класів на тему «Транспортні засоби»**

(варіант 16а)

Виконав: **Шестов Нікіта Сергійович**

Студент I курсу, денна форма навчання

Спеціальність 123 Комп’ютерна інженерія

ОДЕСА – 2023

**ЗМІСТ**

1. **ПОСТАНОВКА ЗАДАЧІ………………………………………3**
2. **ПРОЄКТНЕ РІШЕННЯ………………………………………4**
   1. **Діаграма класів**
   2. **Опис структури класів**
3. **ІНТЕРФЕЙС……………………………………………………**
4. **ВИСНОВКИ……………………………………………………**

**Постановка задачі**

1. Реалізувати клас “Транспортний засіб" Transport з полями: вага weight, максимальна швидкість maxSpeed, витрата пального в літрах на 100 км fuelConsumption.
2. Реалізувати класи-спадкоємці від Transport для різних типів транспортних засобів з визначенням їхніх характеристик:
   1. Клас "Легковий Автомобіль" Car з додатковими полями: максимально можливий багаж maxCargo, клас/сегмент segment, тип кузова (седан, універсал, кабріолет) bodyType, кількість пасажирів passengerCount, тип оббивки сидінь seatType.
   2. Клас "Мотоцикл" з додатковим полем: наявність коляски hasSidecar.
   3. Клас "Вантажний автомобіль" з полем: вантажність cargoCapacity.
   4. Клас "Автобус" з полями: кількість пасажирів passengerCount, наявність кондиціонера hasAirConditioning, наявність відкидних сидінь hasRecliningSeats, зручність видів hasGoodVisibility, максимальний можливий багаж кожного пасажира maxPassengerCargo.
3. Реалізувати метод обчислення комфорту перевезення пасажирів для обраних транспортних засобів: мотоцикл, легковий автомобіль, автобус.
4. Реалізувати такі можливості:
   1. Додавання нових транспортних засобів.
   2. Підбір автомобілів для перевезення пасажирів та їх вантажів — може здійснюватись за принципом ручного розподілу, оптимізації комфорту та/або вартості, в залежності від вибору студента.

**Проєктне рішення**

В ході індивідуальної роботи була розроблена ієрархія класів для представлення різних типів транспортних засобів. Основною метою роботи було створення класів, які відображають характеристики і функціональні можливості різних транспортних засобів.

У створеній ієрархії класів використовувалися наслідування і поліморфізм для забезпечення загальної структури і можливості роботи з різними типами транспортних засобів через їх спільний базовий клас `Transport`. Кожен конкретний тип транспортного засобу було представлено окремим похідним класом, що містить додаткові характеристики і методи, специфічні для цього типу транспортного засобу.

Задля логіки у реалізації підбору транспортних засобів до класу Transport було додано додаткову властивість number — номер дає можливість користувачу ефективно відрізняти транспортні засоби. У класі створено два віртуальних методи — для реалізації підбору транспортних засобів calculateValue() та для додаткової можливості переглянути повну інформацію про транспортні засоби getInfo().

У той час як метод calculateValue() для класів Car, Motorcycle та Bus розраховує комфорт за формулами, що були виведені інтуїтивно, — для класу Truck цей метод повертає вказану вартість оренди вантажного авто на місяць.

Для роботи з колекцією транспортних засобів був створений клас `Transports`, який забезпечує функціонал додавання нових транспортних засобів, виведення списків транспортних засобів за різними критеріями (наприклад, комфортність чи вартість), а також звільнення пам'яті при завершенні роботи з колекцією.

Під час розробки було дотримано принципів об'єктно-орієнтованого програмування, таких як спадкування, інкапсуляція, поліморфізм, абстракція. Це дозволило створити модульну і легко розширювану структуру класів, що може бути використана для подальшого розроблення та розширення функціональності.

Використання функціонала реалізовано через консольний інтерфейс.

Для користувача доступні такі можливості:

1. Додавання транспортного засобу за категорією.
2. Підбір найкомфортнішого транспортного засобу (легкове авто, мото, автобус).
3. Підбір найдоступнішої вантажівки для перевезення вантажу.
4. Перегляд повної інформації про транспортні засоби за категорією.

Реалізовано перевірку введених користувачем даних під час додавання нового транспортного засобу.

**Діаграма класів**

(мал. 1)

**Опис структури класів**

1. Основний клас **Transport**.

* Приватні поля: weight, maxSpeed, fuelConsumption, number;
* Конструктор з параметрами;
* Методи:
  + getWeight() - гетер ваги;
  + getMaxSpeed() - гетер максимальної швидкості;
  + getFuelConsumption() - гетер витрати пального;
  + getNumber() - гетер номера;
  + calculateValue() - віртуальний метод;
  + getInfo() - віртуальний метод;

1. Клас **Car.**

* Приватні поля: int maxCargo, segment, bodyType, passengerCount, seatType;
* Конструктор з параметрами;
* Методи:
  + calculateValue - метод обчислює значення комфорту автомобіля;
  + getInfo() - метод повертає повну інформацію про авто;

1. Клас **Motorcycle.**

* Приватне поле: hasSidecar;
* Конструктор з параметрами;
* Методи:
  + calculateValue - метод обчислює значення комфорту мотоцикла;
  + getInfo() - метод повертає повну інформацію про мотоцикл;

1. Клас **Truck.**

* Приватні поля: cargoCapacity, cost;
* Конструктор з параметрами;
* Методи:
  + calculateValue - повертає вартість оренди вантажного авто;
  + getInfo() - метод повертає повну інформацію про вантажне авто;

1. Клас **Bus.**

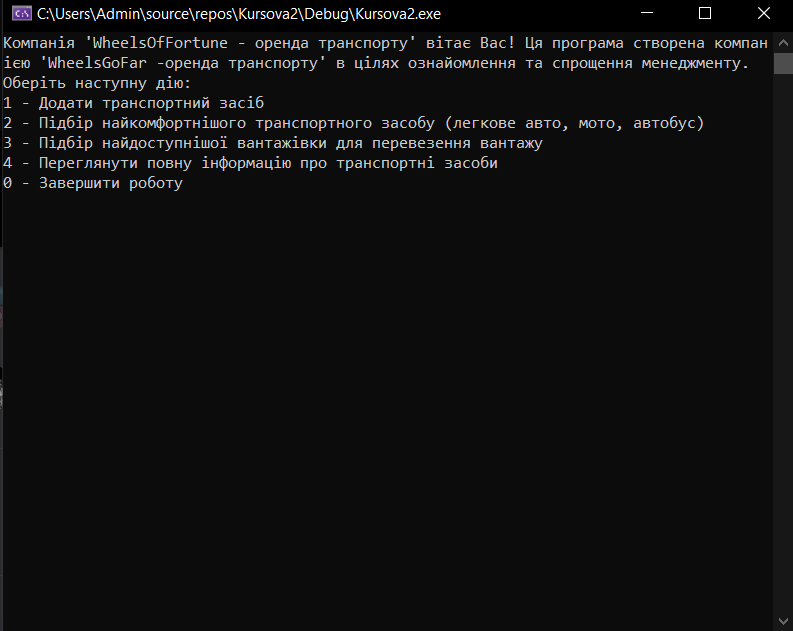
* Приватні поля: passengerCount, hasAirConditioning, hasRecliningSeats, hasGoodVisibility, maxPassengerCargo;
* Конструктор з параметрами;
* Методи:
  + calculateValue - метод обчислює значення комфорту автобуму;
  + getInfo() - метод повертає повну інформацію про автобус;

1. Клас **Transports**.

* Приватне поле: transports - вектор вказівників на об'єкти класу Transport;
* Деструктор ~Transports;
* Методи:
  + addTransport() - метод для додавання “шаблонів” транспорту через код програми;
  + addUserTransport() - метод для додавання транспорту через консоль;
  + findMostComfortableVehicles() - метод повертає вектор транспортних засобів, відсортованих за спаданням комфортності;
  + findMostAffordableVehicles() - метод повертає вектор транспортних засобів (вантажівок), відсортованих за зростанням вартості
  + getList() - метод повертає вектор транспортних засобів заданого типу.

**Інтерфейс**

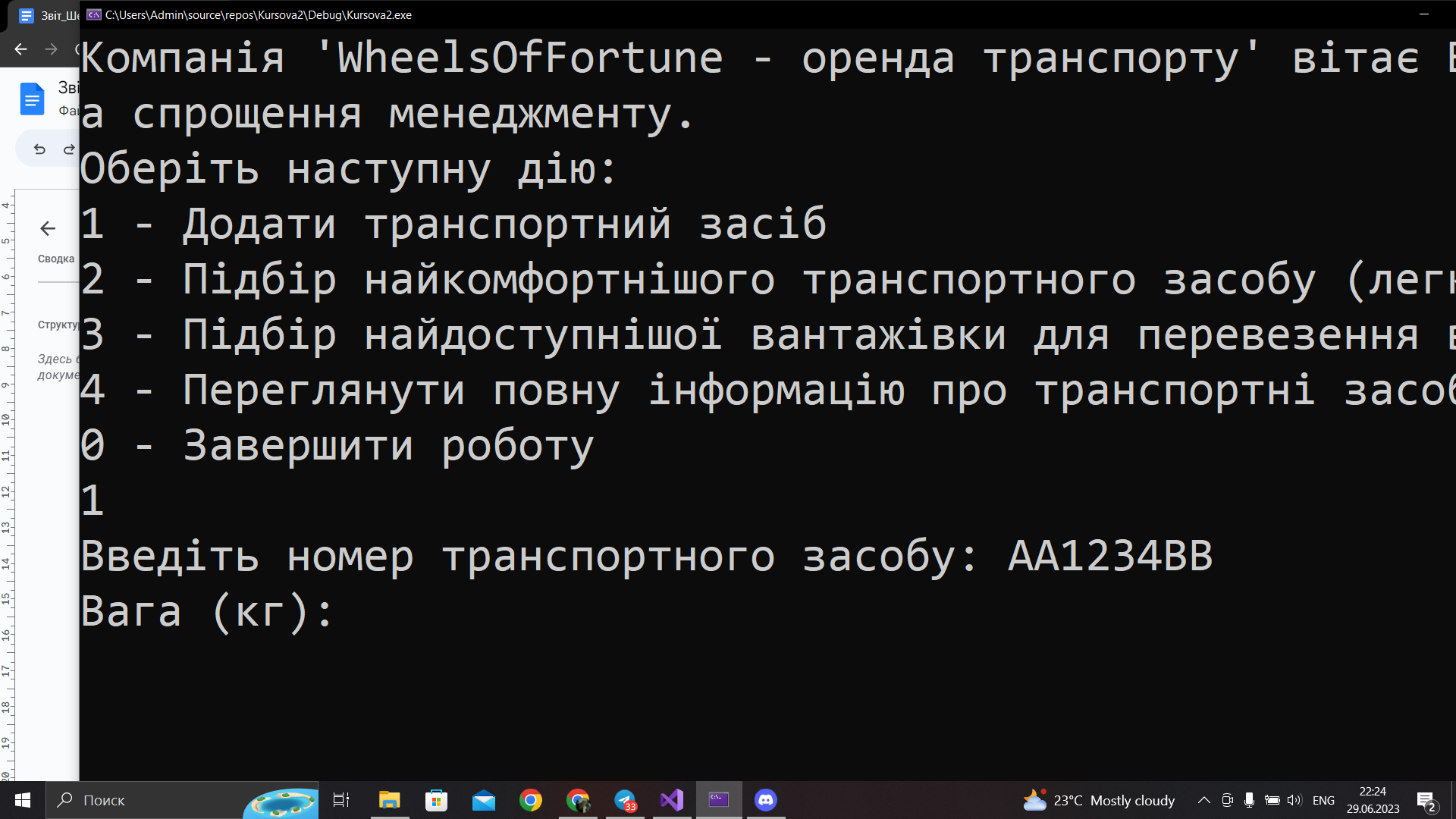
Під час запуску програми користувач бачить наступне (Мал. 2.1.):



(мал. 2. 1)

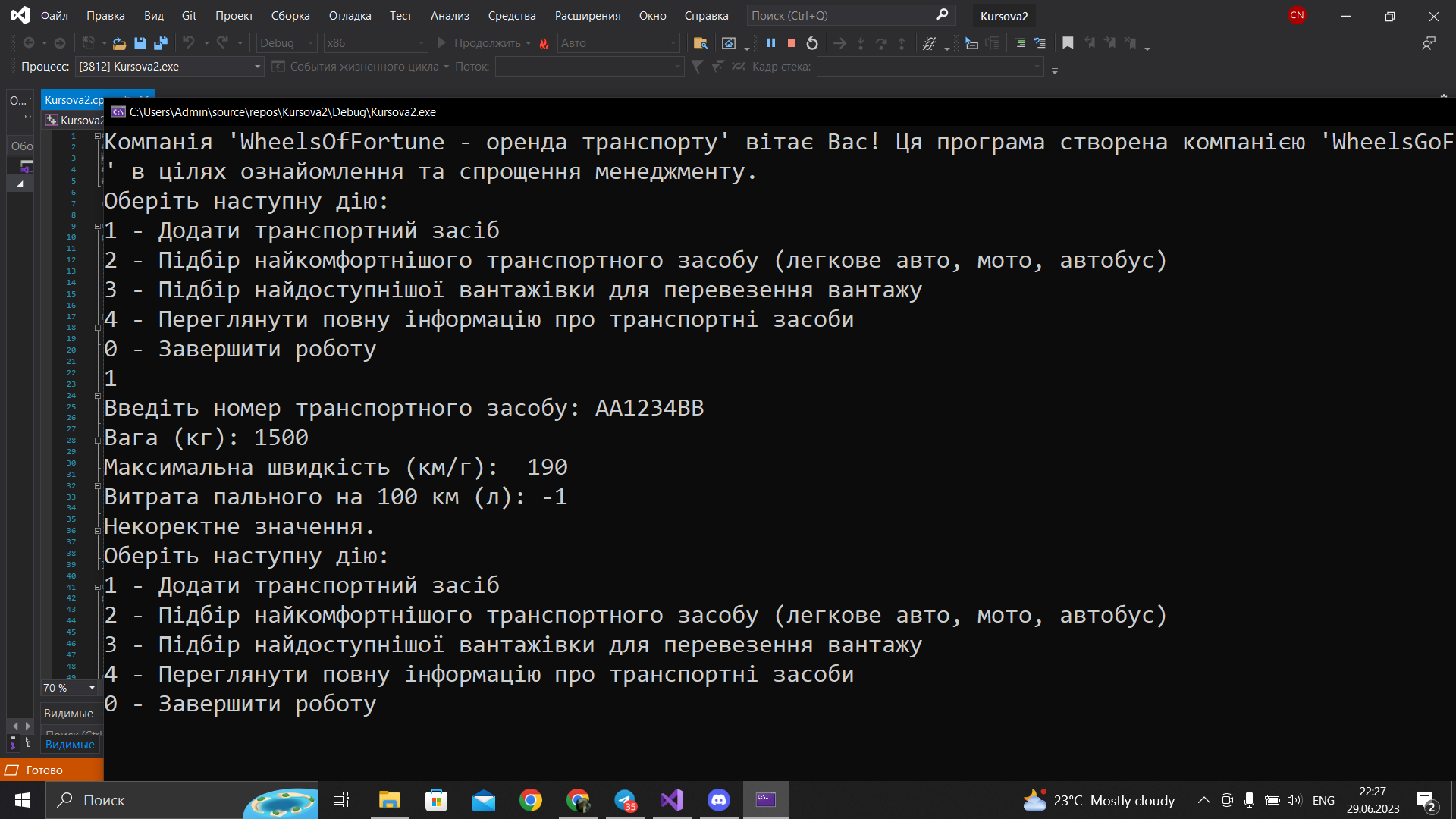
Меню реалізовано через цикл while() та за допомогою структури switch() case:{} — це дає умовному користувачу можливість зручно взаємодіяти з програмою через консоль.

При виборі варіанта «1» — ми починаємо процес додавання нового транспортного засобу та вводимо потрібні дані, спираючись на підказки у дужках. (Мал. 2.2):



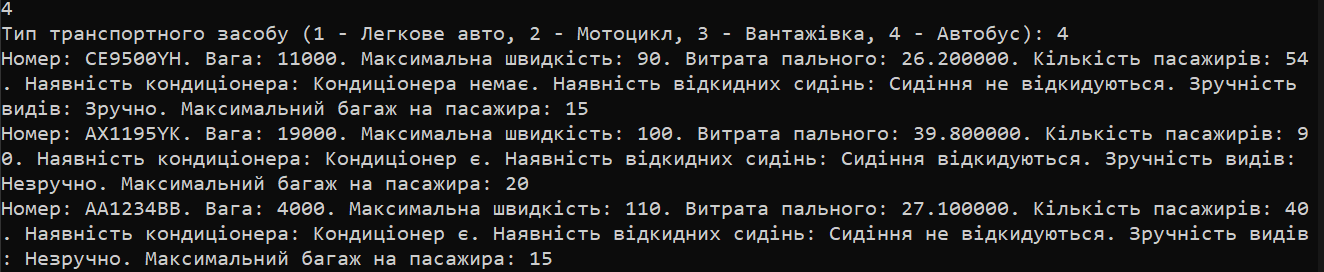
(мал. 2. 2)

При непроходженні даними перевірки виводиться відповідне повідомлення (Мал. 2.3):



(мал. 2. 3)

Після введення всіх даних нас повертає в меню. Аби перевірити успішність додавання транспорту — переходимо за варіантом «4» та обираємо потрібний тип транспорту (вибір зроблено зі структурою switch() case:{}). (Мал. 2.4):



(мал. 2. 4)

**Висновки**

1. Розроблена ієрархія класів дозволяє моделювати різні типи транспортних засобів та працювати з ними у єдиному інтерфейсі.

2. Використання поліморфізму дозволяє здійснювати операції з різними типами транспортних засобів, використовуючи спільний базовий клас.

3. Клас `Transports` надає зручний інтерфейс для додавання, виведення та фільтрації транспортних засобів у колекції.

4. Розроблені класи дотримуються принципів об'єктноорієнтованого програмування, забезпечуючи модульність і легку розширюваність.

Загалом, робота над створенням ієрархії класів на тему "Транспортні засоби" дозволила практично застосувати концепції об'єктноорієнтованого програмування і створити функціональну структуру для роботи з різними типами транспортних засобів.