### Кам’янець-Подільський національний університет імені Івана Огієнка

### КАФЕДРА КОМП’ЮТЕРНИХ НАУК

### Навчальна дисципліна «Об’єктно-орієнтоване програмування»

### ЛАБОРАТОРНА РОБОТА #0102

## Тема:

Класи, структури та простори імен

Варіант №9

**Виконав**:  
студент 2-го курсу  
групи KN1-B23  
Серебрянський Артем О.

**Прийняв**:  
доцент,  
Слободянюк О.В.

### Кам’янець-Подільський – 2025

1. **Короткі теоретичні відомості:**

#### **1. Клас**

* Клас – це опис безлічі об'єктів, які мають однакові атрибути та поведінку. Він є абстрактним типом, на основі якого створюються об'єкти. Клас містить поля (атрибути) і методи (функції, що виконують дії з цими полями)
* Створення класу. Щоб створити клас у C#, необхідно оголосити його за допомогою ключового слова class, визначити поля (атрибути) та методи (функції).
* Синтаксис створення класу  
  class Ім'яКласу

{

// Поля (змінні класу)

тип ім'яПоля;

// Конструктор

public Ім 'яКласу()

{

// Ініціалізація

}

// Методи (функції)

public void Ім'яМетоду()

{

// Логіка методу

}

}

#### **2. Конструктор**

* Конструктор – це спеціальний метод класу, який автоматично викликається при створенні об'єкта і ініціалізує його поля. Його ім’я збігається з іменем класу, він не повертає значення і може мати параметри.

#### **3. Перевантаження методів**

* Перевантаження методів дозволяє створювати кілька методів з однаковим ім'ям, але з різними параметрами (за кількістю чи типом). Це дозволяє використовувати один метод для різних типів вхідних даних, що підвищує гнучкість коду.

#### **4. Статичні методи**

#### Статичні методи належать класу, а не об'єктам. Вони викликаються через ім'я класу (ІМ'Я\_КЛАСУ.МЕТОД()). Такі методи зазвичай використовуютьс для операцій, що не залежать від конкретного екземпляра класу, наприклад, математичні обчислення. Статичні методи належать класу, а не об'єктам. Вони викликаються через ім'я класу (ІМ'Я\_КЛАСУ.МЕТОД()). Такі методи зазвичай використовуються для операцій, що не залежать від конкретного екземпляра класу, наприклад, математичні обчислення. **5. Статичні поля**

Статичні поля є спільними для всіх об'єктів класу. Вони зберігають єдине значення для всіх екземплярів класу. Використовуються для збереження глобальних параметрів або лічильників створених об'єктів.

1. **Повні умови завдань:**

**Задача 1. Ініціалізація за замовчуванням:**

Створіть клас Car з властивістю Speed. Ініціалізуйте значення за замовчуванням у

конструкторі.

**Задача 2. Перевантаження методів:**

Створіть клас Printer з методами Print для виведення різних типів даних (рядок, ціле число, число з плаваючою комою).

**Задача 3. Статичні поля та методи для рахунку:**

Створіть клас Counter з приватним статичним полем. Реалізуйте метод для збільшення значення лічильника та виведення його на екран.

1. **Послідовність виконання завдань:**
2. **Виконання завдання. Ініціалізація за замовчуванням:**

* **Створити клас Car.**
* **Додати властивість Speed.**
* **Реалізувати конструктор для ініціалізації значення за замовчуванням.**

1. **Виконання завдання. Перевантаження методів:**

* **Створити клас Printer.**
* **Реалізувати метод Print з трьома перевантаженими варіантами для різних типів даних (рядок, ціле число, число з плаваючою комою).**

1. **Виконання завдання. Статичні поля та методи для рахунку:**

* **Створити клас Counter.**
* **Додати приватне статичне поле.**
* **Реалізувати статичні методи для збільшення значення лічильника та його виведення на екран.**

**4. Лістинги програм:**

**Задача 1. Ініціалізація за замовчуванням:**class Car

{

public int Speed { get; set; }

public Car()

{

Speed = 100; // Значення за замовчуванням

}

}

class Program

{

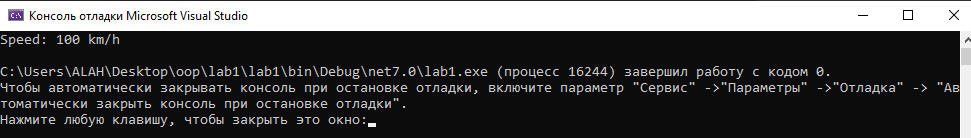
static void Main(string[] args)

{

Car car = new Car();

Console.WriteLine($"Speed: {car.Speed} km/h");

}

}  


**Задача 2. Перевантаження методів:**class Printer

{

public void Print(string message)

{

Console.WriteLine($"String: {message}");

}

public void Print(int number)

{

Console.WriteLine($"Integer: {number}");

}

public void Print(double number)

{

Console.WriteLine($"Double: {number}");

}

}

class Program

{

static void Main(string[] args)

{

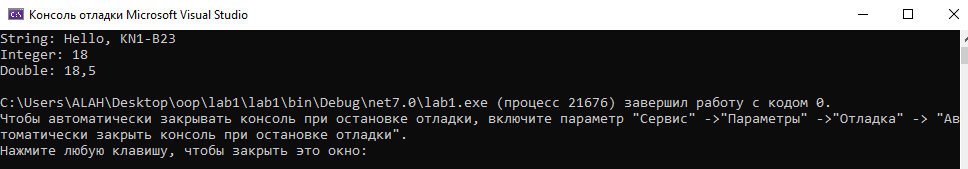
Printer printer = new Printer();

printer.Print("Hello, KN1-B23");

printer.Print(18);

printer.Print(18.5);

}

}  
 **Задача 3. Статичні поля та методи для рахунку:**class Counter

{

// Приватне статичне поле для лічильника

private static int count = 0;

// Метод для збільшення лічильника

public static void Increment()

{

count++;

}

// Метод для виведення лічильника на екран

public static void DisplayCount()

{

Console.WriteLine($"Current count: {count}");

}

}

class Program

{

static void Main(string[] args)

{

Counter.Increment();

Counter.Increment();

Counter.Increment();

Counter.DisplayCount(); // буде 3 тому, що Counter.Increment(); викликається 3 рази"

}

}

