Чернівецький національний університет імені Юрія Федьковича

Навчально-науковий інститут фізико-технічних та комп’ютерних наук

Кафедра математичних проблем управління і кібернетики

**Звіт**

про виконання лабораторної роботи №3

«Об'єкти й класи (конструктори, деструктори, методи класу,

спадкування).»

з дисципліни

«Крос-платформне програмування»

Виконав: студент 241 групи

Фрасинюк О.Б.

Перевірив: канд.фіз.-мат. наук,

доцент Лазорик В.В.

Оцінка:

Дата захисту:

Чернівці 2024

Варіант 17

<https://github.com/VLazorykOOP/lab3csharp25-Oleh-Frasyniuk>

Завдання 1. Для розв’язання задачі згідно варіанту

створити клас із полями, конструкторами, методами та властивостями. До

запропонованих полів, методів та властивосте можна додавати власні.

1.7. Задано масив трапецій. Вивести в консоль інформацію про

трапеції та їх площу і периметр. Визначити скільки є квадратів.

Створити клас Trapeze (трапеція), розробивши такі елементи класу:

 Поля:

 основи та висота (int a, b , h);

 колір трапеції (int с);

 Конструктор, що дозволяє створити екземпляр класу з

заданими довжинами.

 Методи, що дозволяють:

 вивести довжини на екран;

 розрахувати периметр трапеції;

 розрахувати площу трапеції

 дозволяє встановити, чи є даний трапеція квадратом

(доступні тільки для читання).

 Властивості:

 отримати-встановити довжини (доступні для читання і

запису);

 отримати колір (доступна тільки для читання).

using System;

public class Trapeze

{

// Поля

private int a; // довжина першої основи

private int b; // довжина другої основи

private int h; // висота трапеції

private int c; // колір трапеції

// Конструктор

public Trapeze(int a, int b, int h, int c)

{

this.a = a;

this.b = b;

this.h = h;

this.c = c;

}

// Властивості для доступу до полів

public int A

{

get { return a; }

set { a = value; }

}

public int B

{

get { return b; }

set { b = value; }

}

public int H

{

get { return h; }

set { h = value; }

}

public int C

{

get { return c; }

}

// Метод для виведення довжин на екран

public void DisplayDimensions()

{

Console.WriteLine($"Основа 1: {a}, Основа 2: {b}, Висота: {h}, Колір: {c}");

}

// Метод для обчислення периметра трапеції

public double CalculatePerimeter()

{

double side = Math.Sqrt(Math.Pow(h, 2) + Math.Pow((a - b) / 2.0, 2));

return a + b + 2 \* side;

}

// Метод для обчислення площі трапеції

public double CalculateArea()

{

return ((a + b) / 2.0) \* h;

}

// Метод для перевірки чи є трапеція квадратом

public bool IsSquare

{

get { return a == b && h == a; }

}

// Статичний метод для підрахунку кількості квадратів у масиві трапецій

public static int CountSquares(Trapeze[] trapezes)

{

int count = 0;

foreach (var trapeze in trapezes)

{

if (trapeze.IsSquare)

{

count++;

}

}

return count;

}

}

class Program

{

static void Main()

{

// Створюємо масив трапецій

Trapeze[] trapezes = new Trapeze[]

{

new Trapeze(5, 5, 5, 1),

new Trapeze(10, 10, 10, 2),

new Trapeze(5, 7, 4, 3),

new Trapeze(3, 5, 4, 4)

};

// Виводимо інформацію про трапеції, їх площу, периметр

foreach (var trapeze in trapezes)

{

trapeze.DisplayDimensions();

Console.WriteLine($"Периметр: {trapeze.CalculatePerimeter():F2}");

Console.WriteLine($"Площа: {trapeze.CalculateArea():F2}");

Console.WriteLine($"Чи є квадратом: {trapeze.IsSquare}");

Console.WriteLine();

}

// Підраховуємо скільки трапецій є квадратами

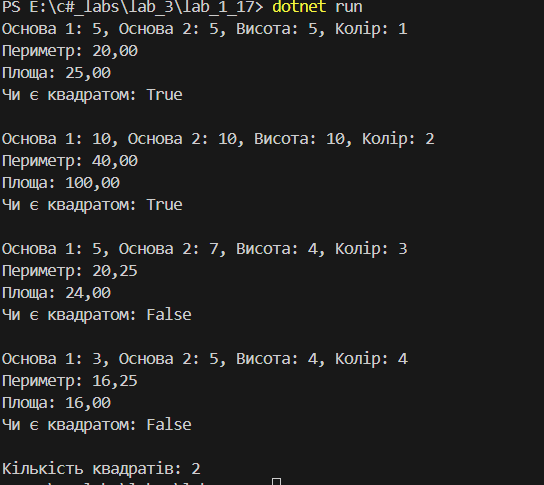
int squareCount = Trapeze.CountSquares(trapezes);

Console.WriteLine($"Кількість квадратів: {squareCount}");

}

}

Результат:



Завдання 2. Побудувати ієрархію класів відповідно до варіанта завдання. Згідно завдання вибрати базовий клас та похідні. В класах задати поля, які характерні для кожного класу. Для всіх класів розробити метод Show(), який виводить дані про об’єкт класу. Створити масив базового класу та написати функцію наповняє масив різними об’єктами похідних класів. Вивести масив впорядкований за деяким критерієм який характеризує всі об’єкти масиву. 2.2. Службовець, персона, робітник, інженер.

using System;

public class Person

{

// Поля базового класу

public string Name { get; set; }

public int Age { get; set; }

// Конструктор

public Person(string name, int age)

{

Name = name;

Age = age;

}

// Метод для виведення даних про об'єкт

public virtual void Show()

{

Console.WriteLine($"Ім'я: {Name}, Вік: {Age}");

}

}

public class Employee : Person

{

// Поле для класу Службовець

public string Position { get; set; }

// Конструктор

public Employee(string name, int age, string position) : base(name, age)

{

Position = position;

}

// Перевизначення методу Show()

public override void Show()

{

base.Show();

Console.WriteLine($"Посада: {Position}");

}

}

public class Worker : Person

{

// Поле для класу Робітник

public string Specialty { get; set; }

// Конструктор

public Worker(string name, int age, string specialty) : base(name, age)

{

Specialty = specialty;

}

// Перевизначення методу Show()

public override void Show()

{

base.Show();

Console.WriteLine($"Спеціальність: {Specialty}");

}

}

public class Engineer : Person

{

// Поле для класу Інженер

public string AreaOfWork { get; set; }

// Конструктор

public Engineer(string name, int age, string areaOfWork) : base(name, age)

{

AreaOfWork = areaOfWork;

}

// Перевизначення методу Show()

public override void Show()

{

base.Show();

Console.WriteLine($"Область роботи: {AreaOfWork}");

}

}

class Program

{

// Функція для заповнення масиву різними об'єктами

public static Person[] FillArray()

{

Person[] people = new Person[5];

// Створення об'єктів різних типів

people[0] = new Employee("Іван Петренко", 35, "Менеджер");

people[1] = new Worker("Олексій Сидоренко", 40, "Слюсар");

people[2] = new Engineer("Марина Коваль", 30, "Програміст");

people[3] = new Employee("Аліна Кравченко", 28, "Бухгалтер");

people[4] = new Worker("Петро Іванов", 45, "Електрик");

return people;

}

// Функція для впорядкування масиву за віком

public static void SortByAge(Person[] people)

{

Array.Sort(people, (x, y) => x.Age.CompareTo(y.Age));

}

static void Main()

{

// Заповнення масиву

Person[] people = FillArray();

// Виведення початкового масиву

Console.WriteLine("До сортування:");

foreach (var person in people)

{

person.Show();

Console.WriteLine();

}

// Сортуємо масив за віком

SortByAge(people);

// Виведення відсортованого масиву

Console.WriteLine("Після сортування за віком:");

foreach (var person in people)

{

person.Show();

Console.WriteLine();

}

}

}

Результат:

