

Московский государственный технический университет
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №3

ИИ

Выполнил:
студент группы ИУ5-36Б
Германенко И. М.
Преподаватель:
Гапанюк Ю.Е.

Москва, 2025

Описание задания:

Разработать консольное приложение на C#, которое:

- Создать обобщённые функции с кортежами разных типов;
- Реализовать специализированные функции сложения для int, float, string с использованием **двухэтапного вывода типов**;
- Применить **list comprehension** для генерации кортежей (число, квадрат, куб) для чётных чисел от 1 до 10;
- Реализовать обработку списка (квадраты элементов) через Head/Tail с if и с сопоставлением с образцом;
- Последовательно применить map, filter, sort, fold, zip, агрегирующие функции в трёх комбинациях;
- Выполнить то же с операторами |> (forward pipe) и >> (композиция).

Листинг кода:

```
using System;
using System.Collections;
using System.Collections.Generic;

#region Классы из Лаб. №2 (модифицированные)

// ИСПРАВЛЕНИЕ: Добавляем обычный интерфейс IComparable для работы ArrayList
public abstract class GeometricFigure : IComparable<GeometricFigure>, IComparable
{
    public abstract double Area();

    // Метод для List<GeometricFigure> (современный)
    public int CompareTo(GeometricFigure other)
    {
        if (other == null) return 1;
        return this.Area().CompareTo(other.Area());
    }

    // Метод для ArrayList (старый, принимает object)
    public int CompareTo(object obj)
    {
        if (obj == null) return 1;

        // Проверяем, является ли объект фигурой
        GeometricFigure otherFigure = obj as GeometricFigure;
        if (otherFigure != null)
        {
            return this.CompareTo(otherFigure); // Вызываем современный метод
        }
        else
        {
            throw new ArgumentException("Object is not a GeometricFigure");
        }
    }
}
```

```

public interface IPrint
{
    void Print();
}

public class Rectangle : GeometricFigure, IPrint
{
    public double Width { get; set; }
    public double Height { get; set; }
    public Rectangle(double w, double h) { Width = w; Height = h; }
    public override double Area() => Width * Height;
    public override string ToString() => $"Прямоугольник ({Width}x{Height}),
S={Area():F2}";
    public void Print() => Console.WriteLine(this);
}

public class Square : Rectangle
{
    public Square(double side) : base(side, side) { }
    public override string ToString() => $"Квадрат ({Width}x{Width}),
S={Area():F2}";
}

public class Circle : GeometricFigure, IPrint
{
    public double Radius { get; set; }
    public Circle(double r) { Radius = r; }
    public override double Area() => Math.PI * Radius * Radius;
    public override string ToString() => $"Круг (R={Radius}), S={Area():F2}";
    public void Print() => Console.WriteLine(this);
}

```

#endregion

#region Пункт 7: SimpleStack на основе односвязного списка

```

public class SimpleStack<T>
{
    private class Node
    {
        public T Data { get; }
        public Node Next { get; }
        public Node(T data, Node next) { Data = data; Next = next; }
    }

    // Инициализируем null, чтобы убрать предупреждение CS8618
    private Node _head = null;

    public void Push(T element)
    {
        _head = new Node(element, _head);
    }
}

```

```

    }

    public T Pop()
    {
        if (_head == null)
        {
            throw new InvalidOperationException("Стек пуст.");
        }
        T data = _head.Data;
        _head = _head.Next;
        return data;
    }

    public bool IsEmpty => _head == null;
}

#endregion

class Program
{
    static void Main(string[] args)
    {
        var rect = new Rectangle(5, 10);    // S = 50
        var square = new Square(8);         // S = 64
        var circle = new Circle(5);         // S ~ 78.5

        // --- Пункт 4: Работа с необобщенной коллекцией ArrayList ---
        Console.WriteLine("--- Тест ArrayList (необобщенная коллекция) ---");
        var arrayList = new ArrayList { rect, circle, square };

        // Теперь это сработает, так как мы добавили CompareTo(object obj)
        arrayList.Sort();

        foreach (var item in arrayList)
        {
            ((IPrint)item).Print();
        }
        Console.WriteLine();

        // --- Пункт 5: Работа с обобщенной коллекцией List<T> ---
        Console.WriteLine("--- Тест List<GeometricFigure> (обобщенная коллекция) ---");
        var list = new List<GeometricFigure> { rect, circle, square };
        list.Sort();
        foreach (var figure in list)
        {
            ((IPrint)figure).Print();
        }
        Console.WriteLine();

        // --- Пункт 8: Пример работы SimpleStack ---
    }
}

```

```

        Console.WriteLine("--- Тест SimpleStack ---");
        var stack = new SimpleStack<GeometricFigure>();
        stack.Push(rect);
        stack.Push(circle);
        stack.Push(square);

        Console.WriteLine("Извлекаем элементы из стека (LIFO):");
        while(!stack.IsEmpty)
        {
            var figure = stack.Pop();
            ((IPrint)figure).Print();
        }
        Console.WriteLine();
    }
}

```

Скриншоты:

```

● root@ILGERPC:/home/PCPL/Lab3# dotnet run
/home/PCPL/Lab3/Program.cs(13,16): warning CS8767: Nullability of 'other' doesn't match implicitly implemented member 'int butes). [/home/PCPL/Lab3/Lab3.csproj]
/home/PCPL/Lab3/Program.cs(20,16): warning CS8767: Nullability of 'n't match implicitly implemented member 'int IComparable.
/home/PCPL/Lab3/Program.cs(81,26): warning CS8625: Cannot
/home/PCPL/Lab3/Program.cs(25,39): warning CS8600: Conversion
--- Тест ArrayList (необобщенная коллекция) ---
Прямоугольник (5x10), S=50.00
Квадрат (8x8), S=64.00
Круг (R=5), S=78.54

--- Тест List<GeometricFigure> (обобщенная коллекция) ---
Прямоугольник (5x10), S=50.00
Квадрат (8x8), S=64.00
Круг (R=5), S=78.54

--- Тест SimpleStack ---
Извлекаем элементы из стека (LIFO):
Квадрат (8x8), S=64.00
Круг (R=5), S=78.54
Прямоугольник (5x10), S=50.00
○ root@ILGERPC:/home/PCPL/Lab3#

```