|  |  |  |
| --- | --- | --- |
| Защищено:  Гапанюк Ю.Е.    "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |  | Демонстрация:  Гапанюк Ю.Е.    "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

**Отчет по лабораторной работе № 3**

**по курсу**

**Базовые компоненты интернет-технологий**

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы ИУ5Ц-52Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Михалев С. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

Москва, МГТУ - 2020

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ ОТЧЕТА**

[1. Задание 3](#_Toc53994319)

[2. Листинг программы 3](#_Toc53994320)

[3. Результаты работы программы 5](#_Toc53994321)

# Задание

Разработать программу, реализующую работу с коллекциями.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создать объекты классов «Прямоугольник», «Квадрат», «Круг».
3. Для реализации возможности сортировки геометрических фигур для класса «Геометрическая фигура» добавить реализацию интерфейса IComparable. Сортировка производится по площади фигуры.
4. Создать коллекцию класса ArrayList. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.
5. Создать коллекцию класса List<Figure>. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.
6. Модифицировать класс разреженной матрицы (проект SparseMatrix) для работы с тремя измерениями – x,y,z. Вывод элементов в методе ToString() осуществлять в том виде, который Вы считаете наиболее удобным. Разработать пример использования разреженной матрицы для геометрических фигур.
7. Реализовать класс «SimpleStack» на основе односвязного списка. Класс SimpleStack наследуется от класса SimpleList (проект SimpleListProject). Необходимо добавить в класс методы:
   1. public void Push(T element) – добавление в стек;
   2. public T Pop() – чтение с удалением из стека.
8. Пример работы класса SimpleStack реализовать на основе геометрических фигур.

# Листинг программы

# using System;

# using System.Collections;

# using System.Collections.Generic;

# using System.Net.NetworkInformation;

# using Collection\_class\_lab3.Figures;

# using Collection\_class\_lab3.classes;

# namespace Collection\_class\_lab3

# {

# class Program

# {

# static void PrintArrayList (ArrayList array)

# {

# foreach (object obj in array)

# {

# string type = obj.GetType().Name;

# if (type == "Rectangle")

# {

# Rectangle rect = obj as Rectangle;

# rect.Print();

# }

# else if (type == "Square")

# {

# Square sq = obj as Square;

# sq.Print();

# }

# else if (type == "Circle")

# {

# Circle circ = obj as Circle;

# circ.Print();

# }

# else

# Console.WriteLine("Unknown type");

# }

# }

# 

# static void PrintList (List<GeometryFigure> figures)

# {

# foreach (object figure in figures)

# {

# string type = figure.GetType().Name;

# if (type == "Rectangle")

# {

# Rectangle rect = figure as Rectangle;

# rect.Print();

# }

# else if (type == "Square")

# {

# Square sq = figure as Square;

# sq.Print();

# }

# else if (type == "Circle")

# {

# Circle circ = figure as Circle;

# circ.Print();

# }

# else

# Console.WriteLine("Unknown type");

# }

# }

# static void Main()

# {

# ArrayList array = new ArrayList();

# Rectangle rectangle = new Rectangle(5,4);

# Square square = new Square(25);

# Circle circle = new Circle(5);

# array.Add(rectangle);

# array.Add(square);

# array.Add(circle);

# Console.WriteLine("Вывод ArrayList до сортировки: ");

# PrintArrayList(array);

# array.Sort();

# 

# Console.WriteLine("\nПосле сортировки");

# PrintArrayList(array);

# List<GeometryFigure> figures = new List<GeometryFigure>

# {

# rectangle,

# square,

# circle

# };

# Console.WriteLine("\nВывод List до сортировки: ");

# PrintList(figures);

# figures.Sort();

# Console.WriteLine("\nПосле сортировки: ");

# PrintList(figures);

# Console.WriteLine("\n Использование разреженной матрицы для геометрических фигур: ");

# SparseMatrix<GeometryFigure> x = new SparseMatrix<GeometryFigure>(2, 2, 2, rectangle);

# x[0, 0, 0] = circle;

# x[1, 1, 1] = square;

# x[0, 1, 0] = square;

# x[0, 0, 1] = square;

# try

# {

# x[100, 100, 100] = rectangle;

# }

# catch (Exception e)

# {

# Console.WriteLine(e.Message);

# }

# Console.WriteLine(x);

# Console.WriteLine("\n Использование класса SimpleStack:");

# SimpleStack<GeometryFigure> stack\_figures = new SimpleStack<GeometryFigure>();

# stack\_figures.Push(rectangle);

# stack\_figures.Push(circle);

# stack\_figures.Push(square);

# while (stack\_figures.Count > 0)

# {

# GeometryFigure figure = stack\_figures.Pop();

# Console.WriteLine(figure);

# }

# Console.WriteLine("\n Проверка заполненности стека: ");

# if (stack\_figures.Count == 0)

# {

# Console.WriteLine("Стек пуст");

# }

# }

# }

# }

# Результаты работы программы

