****

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»**

Институт (факультет) Информатики

Кафедра Программных систем

ОТЧЁТ

по лабораторной работе №01

Программы с использованием цикловпо дисциплине «Основы программирования»

Выполнил Боряков Никита Сергеевич, группа 6102-020302D

Проверил Котенёва Светлана Эдуардовна

Самара  
2021

ЗАДАНИЯ

Задание 1.

Описать класс «вектор» с именем “ArrayVector” со следующей структурой:

* поле массив элементов целого типа (координаты вектора в пространстве);
* конструктор с параметром – длиной массива;
* конструктор без параметров, создающий массив из 5 элементов;
* индексатор для организации доступа к элементам массива, выбрасывающий исключения при некорректном индексе;
* метод получения модуля (длины) вектора GetNorm;
* свойство для чтения числа координат вектора.

Задание 2.

Описать класс «список» с именем «LinkedListVector», содержащий массив элементов целого типа в виде динамического односвязного списка. Каждый элемент массива представляет собой отдельный объект «узел» класса «Node», класс Node является внутренним для класса LinkedListVector.

Структура класса «Node»:

* поле – элемент целого типа (по умолчанию = 0);
* поле – ссылка на элемент класса Node (по умолчанию = null).

Структура класса «LinkedListVector»:

* поле – ссылка на начало списка (на объект класса Node);
* конструктор с параметром – длиной списка;
* конструктор без параметра, задающий длину списка 5;
* индексатор для организации доступа к элементам массива, выбрасывающий исключение при некорректном индексе;
* метод получения модуля вектора GetNorm;
* свойство для чтения числа координат вектора.

Задание 3.

Добавить класс с именем «Vectors», содержащий публичные статические методы:

* сложения двух векторов Sum (принимает в качестве параметра 2 объекта ArrayVector, возвращает новый объект ArrayVector);
* скалярного произведения двух векторов Scalar (принимает в качестве параметра 2 объекта ArrayVector, возвращает вещественное число);
* умножения вектора на число NumberMult (принимает в качестве параметра объект ArrayVector и вещественное число, возвращает новый объект ArrayVector);
* получения модуля (длины) вектора GetNorm (принимает в качестве

параметра объект ArrayVector, возвращает вещественное число).

Выбрасывать исключения в методах Sum() и Scalar() в случае невозможности проведения указанных действий над векторами (возможно, FormatException). (попробовать по желанию).

Задание 4.

Проверить функциональность классов в методе Main класса Program. Разработать адекватный пользовательский интерфейс. Отлавливать все возможные исключения – некорректный ввод пункта меню, некорректный ввод данных, несовпадение длин векторов в статических методах класса Vectors и т.д. То есть сделать программу правильно реагирующей на предсказуемые ошибки.

Задание 5.

Подготовить отчет о работе.

КОД ПРОГРАММЫ

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_work\_0201

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Лабораторная работа №1.\n" +

"Выполнил студент группы 6102 Боряков Никита\n ");

Console.WriteLine("Выберете режим работы\n" +

"1 - Array Vector\n" +

"2 - Vectors\n" +

"3 - Выход из программы\n");

int m = int.Parse(Console.ReadLine());

while (m != 3)

{

switch (m)

{

case 1:

{

Console.Clear();

Console.Write("Введите кол-во пространств:");

int count = int.Parse(Console.ReadLine());

// Экземпляр класса

ArrayVektor v = new ArrayVektor(count);

v.EnterVector();

Console.Clear();

Console.WriteLine(v.ToString());

Console.WriteLine(" 1 - \"Установка элемнта по индексу\"\n " +

"2 - \"Чтение элемента массива\"\n " +

"3 - \"Получение модуля вектора\"\n " +

"4 - \"Подсчет суммы всех положительных элементов с четными номерами\"\n " +

"5 - \"Подсчёт суммы тех элементов массива, которые имеют нечетные номера и\"\n " +

"\"одновременно меньше среднего значения всех модулей элементов массива\"\n " +

"6 - \"Подсчёт произведения всех четных положительных элементов\"\n " +

"7 - \"Подсчёт всех нечетных элементов (по значению), не делящихся на три\"\n " +

"8 - \"Сортировка иассива\"\n " +

"9 - \"Выход из программы\"\n ");

int k = int.Parse(Console.ReadLine());

while (k != 9)

{

switch (k)

{

case 1:

{

Console.Write("Введите индекс элемента массива: ");

int i = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите значение элемента массива: ");

int value = Convert.ToInt32(Console.ReadLine());

v.SetElement(i, value);

Console.WriteLine(v.ToString());

Console.ReadLine();

break;

}

case 2:

{

Console.WriteLine("Введите индекс элемента массива: ");

int ind = Convert.ToInt32(Console.ReadLine());

v.GetElement(ind);

Console.ReadLine();

break;

}

case 3:

{

Console.WriteLine(v.ToString());

Console.WriteLine("Длина вектора = " + v.GetNorm());

Console.ReadLine();

break;

}

case 4:

{

v.SumPositivesFromChetIndex();

Console.WriteLine(v.ToString());

Console.ReadLine();

break;

}

case 5:

{

v.SumLessFromNechetIndex();

Console.WriteLine(v.ToString());

Console.ReadLine();

break;

}

case 6:

{

v.MultiChet();

Console.WriteLine(v.ToString());

Console.ReadLine();

break;

}

case 7:

{

v.MultiNechet();

Console.WriteLine(v.ToString());

Console.ReadLine();

break;

}

case 8:

{

v.Sort();

Console.WriteLine(v.ToString());

Console.ReadLine();

break;

}

}

Console.Clear();

Console.WriteLine(" 1 - \"Установка элемнта по индексу\"\n " +

"2 - \"Чтение жлемента массива\"\n " +

"3 - \"Получение модуля вектора\"\n " +

"4 - \"Подсчет суммы всех положительных элементов с четными номерами\"\n " +

"5 - \"Подсчёт суммы тех элементов массива, которые имеют нечетные номера и\"\n " +

"\"одновременно меньше среднего значения всех модулей элементов массива\"\n " +

"6 - \"Подсчёт произведения всех четных положительных элементов\"\n " +

"7 - \"Подсчёт всех нечетных элементов (по значению), не делящихся на три\"\n " +

"8 - \"Сортировка иассива\"\n " +

"9 - \"Выход из программы\"\n ");

k = int.Parse(Console.ReadLine());

}

break;

}

case 2:

{

Console.Clear();

Console.Write("Введите кол-во пространств первого вектора:");

int count1 = int.Parse(Console.ReadLine());

// Экземпляр класса

ArrayVektor v1 = new ArrayVektor(count1);

v1.EnterVector();

Console.Write("Введите кол-во пространств второго вектора:");

int count2 = int.Parse(Console.ReadLine());

// Экземпляр класса

ArrayVektor v2 = new ArrayVektor(count1);

v2.EnterVector();

Console.Clear();

Console.WriteLine(v1.ToString());

Console.WriteLine(v2.ToString());

Console.WriteLine(" 1 - \"Сложение векторов\"\n " +

"2 - \"Скалярное произведение векторов\"\n " +

"3 - \"Умножение вектора 1 на число\"\n " +

"4 - \"Умножение вектора 2 на число\"\n " +

"5 - \"Получение модуля вектора 1\"\n " +

"6 - \"Получение модуля вектора 2\"\n " +

"7 - \"Выход в главное меню\"\n ");

int k = int.Parse(Console.ReadLine());

while (k != 9)

{

switch (k)

{

case 1:

{

if (Vectors.Сontrol(v1, v2))

{

Console.WriteLine("Сумма = " + Vectors.Sum(v1, v2));

}

else

{

Console.WriteLine("Данные векторы нельзя сложить!");

}

Console.ReadLine();

break;

}

case 2:

{

if (Vectors.Сontrol(v1, v2))

{

Console.WriteLine("Произведение = " + Vectors.Scalar(v1, v2));

}

else

{

Console.WriteLine("Данные векторы нельзя перемножить!");

}

Console.ReadLine();

break;

}

case 3:

{

Console.Write("Введите число, на которое хотите умножить массив: ");

int value = Convert.ToInt32(Console.ReadLine());

Console.WriteLine(v1 = Vectors.NumberMult(v1, value));

Console.ReadLine();

break;

}

case 4:

{

Console.Write("Введите число, на которое хотите умножить массив: ");

int value = Convert.ToInt32(Console.ReadLine());

Console.WriteLine(v2 = Vectors.NumberMult(v2, value));

Console.ReadLine();

break;

}

case 5:

{

Console.WriteLine("Модуль вектора 1 = " + Vectors.GetNorm(v1));

Console.ReadLine();

break;

}

case 6:

{

Console.WriteLine("Модуль вектора 2 = " + Vectors.GetNorm(v2));

Console.ReadLine();

break;

}

}

Console.Clear();

Console.WriteLine(" 1 - \"Сложение векторов\"\n " +

"2 - \"Скалярное произведение векторов\"\n " +

"3 - \"Умножение вектора 1 на число\"\n " +

"4 - \"Умножение вектора 2 на число\"\n " +

"5 - \"Получение модуля вектора 1\"\n " +

"6 - \"Получение модуля вектора 2\"\n " +

"7 - \"Выход в главное меню\"\n ");

k = int.Parse(Console.ReadLine());

}

break;

}

}

Console.Clear();

Console.WriteLine("Выберете режим работы\n" +

"1 - Array Vector\n" +

"2 - Vectors\n" +

"3 - Выход из программы\n");

m = int.Parse(Console.ReadLine());

}

}

}

}

Изображение выглядит как текст

Автоматически созданное описание Рисунок 1 – Главное меню программы

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 – Задание 1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – Задание 1

Изображение выглядит как текст

Автоматически созданное описание Рисунок 3 – Задание 2

Изображение выглядит как текст

Автоматически созданное описание Рисунок 4 – Задание 2

ВЫВОДЫ

В лабораторной работе были использованы конструкции языка:

* ветвление;
* конструктор;
* свойства;
* форматированный вывод информации на консоль;
* цикл с условием;
* switch;
* классы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОНИКОВ

1. Павловская Т.А. C#. Программирование на языке высокого уровня. Учебник для вузов [Текст]/Т.А. Павловская. – СПб.: Питер, 2007. – 432 с.
2. Шилдт, Герберт. С# 4.0 : полное руководство / Герберт Шилдт ; [пер. с англ. и ред. И. В. Берштейна]. – Москва [и др.] : Вильямс, 2015. – 1056 с.