



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

Институт \_\_\_\_\_ Информатики и кибернетики  
Кафедра \_\_\_\_\_ Программных систем

## **ОТЧЁТ**

### **по лабораторной работе**

**№7 «Язык программирования C#: Windows-приложения»**

**по дисциплине «Языки программирования и структуры данных»**

Выполнил \_\_\_\_\_ **Фадеев А.С. 6101**

Проверил \_\_\_\_\_ **Котенёва С.Э.**

Самара

2024

## **ЗАДАНИЕ**

Задание 1.

Прочитать теоретический материал.

Задание 2.

Реализовать лабораторную работу 5 как Windows-приложение.

Разработать графический интерфейс пользователя, адекватно отражающий функциональность приложения.

Отлавливать и обрабатывать все возможные исключения.

Протестировать работу приложения.

Задание 5.

Подготовить отчет о работе.

## КОД ПРОГРАММЫ

```
using System;
using Lab07;

namespace Lab07
{
    [Serializable]
    public class ArrayVector : IVectorable, IComparable, ICloneable
    {
        private int[] vector;

        public ArrayVector(int length)
        {
            var r = new Random();

            vector = new int[length];
            for (int i = 0; i < vector.Length; i++)
            {
                vector[i] = r.Next(100);
            }
        }

        public ArrayVector()
        {
            var r = new Random();

            vector = new int[5];
            for (int i = 0; i < vector.Length; i++)
            {
                vector[i] = r.Next(1000);
            }
        }

        public int this[int idx]
        {
            get
            {
                if (idx < 0 || idx >= Length)
                {
                    throw new IndexOutOfRangeException("Индекс за пределами вектора");
                }
            }
        }
    }
}
```

```

        }

        return vector[idx];
    }

    set
    {
        if (idx < 0 || idx >= Length)
        {
            throw new IndexOutOfRangeException("Индекс за границами
вектора");
        }

        vector[idx] = value;
    }
}

public int Length
{
    get { return vector.Length; }
}

public double GetNorm()
{
    double acc = 0;
    for (int i = 0; i < Length; i++)
    {
        acc += Math.Pow(vector[i], 2);
    }

    return Math.Sqrt(acc);
}

public int SumPositivesWithEvenIndex()
{
    int acc = 0;
    for (int i = 1; i < Length; i += 2)
    {
        if (vector[i] > 0)
        {
            acc += vector[i];
        }
    }

    return acc;
}

```

```

    }

    public int SumLessAverageAbsoluteWithOddIndex()
    {
        if (Length == 0)
        {
            return 0;
        }

        int average = 0;
        for (int i = 0; i < Length; i++)
        {
            average += Math.Abs(vector[i]);
        }

        average /= Length;

        int acc = 0;
        for (int i = 0; i < Length; i += 2)
        {
            if (vector[i] < average)
            {
                acc += vector[i];
            }
        }

        return acc;
    }

    public int MultiplyEven()
    {
        int result = 0;
        for (int i = 1; i < Length; i += 2)
        {
            if (vector[i] > 0 && vector[i] % 2 == 0)
            {
                if (result == 0) result = 1;
                result *= vector[i];
            }
        }

        return result;
    }

```

```

public int MultiplyOdd()
{
    int result = 0;
    for (int i = 0; i < Length; i += 2)
    {
        if (vector[i] % 2 != 0 && vector[i] % 3 != 0)
        {
            if (result == 0) result = 1;
            result *= vector[i];
        }
    }

    return result;
}

public void SortUp()
{
    int n = Length;
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (vector[j] > vector[j + 1])
            {
                int tmp = vector[j];
                vector[j] = vector[j + 1];
                vector[j + 1] = tmp;
            }
        }
    }
}

public void SortDown()
{
    int n = Length;
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (vector[j] < vector[j + 1])
            {
                int tmp = vector[j];
                vector[j] = vector[j + 1];

```

```

        vector[j + 1] = tmp;
    }
}

}

}

public void Log(string message = "")
{
    if (message != "")
    {
        Console.Write(message + ": ");
    }

    Console.WriteLine(ToString());
}

public override string ToString()
{
    string s = Length.ToString() + ' ';

    for (int i = 0; i < Length; i++)
    {
        s += this[i].ToString();
        if (i != Length - 1) s += ' ';
    }

    return s;
}

public static ArrayVector GetFromUserInput()
{
    int length;
    do
    {
        Console.Write("Введите длину вектора: ");
    } while (!int.TryParse(Console.ReadLine(), out length) || length
<= 0);

    ArrayVector vec = new ArrayVector(length);

    Random r = new Random();
    for (int i = 0; i < length; i++)
    {

```

```

        vec[i] = r.Next(100);
    }

    return vec;
}

public int CompareTo(object obj)
{
    if (!(obj is IVectorable))
    {
        throw new Exception("Можно сравнить только объекты типа
IVectorable");
    }

    IVectorable other = obj as IVectorable;

    if (Length < other.Length) return -1;
    if (Length > other.Length) return 1;
    return 0;
}

public override bool Equals(object obj)
{
    if (!(obj is IVectorable))
    {
        throw new Exception("Можно сравнивать только объекты типа
IVectorable");
    }

    IVectorable other = obj as IVectorable;

    if (Length != other.Length) return false;

    for (int i = 0; i < Length; i++)
    {
        if (this[i] != other[i]) return false;
    }

    return true;
}

public object Clone()
{
    ArrayVector clone = new ArrayVector(Length);

```



```

        for (int i = 0; i < Length; i++)
        {
            clone[i] = this[i];
        }

        return clone;
    }
}
}
using System;

namespace Lab07
{
    [Serializable]
    public class LinkedListVector : IVectorable, IComparable, ICloneable
    {
        private Node head;

        [Serializable]
        private class Node
        {
            public int value = 0;
            public Node next = null;

            public Node(int value)
            {
                this.value = value;
                next = null;
            }
        }

        public LinkedListVector()
        {
            var r = new Random();

            head = new Node(r.Next(100));
            Node cur = head;

            for (int i = 0; i < 5; i++)
            {
                cur.next = new Node(r.Next(100));
                cur = cur.next;
            }
        }
    }
}

```

```

    }

    public LinkedListVector(int length)
    {
        var r = new Random();

        head = new Node(r.Next(100));
        Node cur = head;

        for (int i = 0; i < length; i++)
        {
            cur.next = new Node(r.Next(100));
            cur = cur.next;
        }
    }

    public int this[int idx]
    {
        get
        {
            if (0 <= idx && idx <= Length)
            {
                Node cur = head;
                for (int i = 0; i < idx; i++)
                {
                    cur = cur.next;
                }

                return cur.value;
            }
            else
            {
                throw new IndexOutOfRangeException("Индекс за пределами  
связного списка");
            }
        }
        set
        {
            if (0 <= idx && idx <= Length)
            {
                Node cur = head;
                for (int i = 0; i < idx; i++)
                {
                    cur = cur.next;

```

```

        }

        cur.value = value;
    }
    else
    {
        throw new IndexOutOfRangeException("Индекс за пределами
связного списка");
    }
}

public int Length
{
    get
    {
        if (head == null)
        {
            return -1;
        }

        int length = 0;
        Node cur = head;
        while (cur.next != null)
        {
            cur = cur.next;
            length++;
        }

        return length;
    }
}

public double GetNorm()
{
    double acc = 0;
    Node cur = head;
    for (int i = 0; i < Length; i++)
    {
        acc += Math.Pow(cur.value, 2);
        cur = cur.next;
    }

    return Math.Sqrt(acc);
}

```

```

    }

    public void InsertByIndex(int idx, int value)
    {
        if (idx < 0 || idx > Length) throw new
IndexOutOfRangeException("Индекс за границами связного списка");

        Node node = new Node(value);

        if (idx == 0)
        {
            node.next = head;
            head = node;
            return;
        }

        Node cur = head;
        int curIndex = 0;
        while (cur != null && curIndex < idx - 1)
        {
            cur = cur.next;
            curIndex++;
        }

        if (cur == null) throw new IndexOutOfRangeException("Индекс за
границами связного списка");

        node.next = cur.next;
        cur.next = node;
    }

    public void InsertToStart(int value)
    {
        InsertByIndex(0, value);
    }

    public void InsertToEnd(int value)
    {
        InsertByIndex(Length, value);
    }

    public void DeleteByIndex(int idx)
    {
        if (head == null) throw new Exception("Связный список пуст");

```

```

        if (idx < 0 || idx >= Length) throw new
IndexOutOfRangeException("Индекс за пределами связного списка");

        Node cur = head;

        if (idx == 0)
        {
            head = cur.next;
            return;
        }

        for (int i = 0; cur != null && i < idx - 1; i++)
        {
            cur = cur.next;
        }

        if (cur == null || cur.next == null) return;

        cur.next = cur.next.next;
    }

    public void DeleteFromStart()
    {
        DeleteByIndex(0);
    }

    public void DeleteFromEnd()
    {
        DeleteByIndex(Length - 1);
    }

    public void Log(string message = "")
    {
        if (message != "") Console.Write($"{message}: ");

        Console.WriteLine(ToString());
    }

    public override string ToString()
    {
        string s = Length.ToString() + ' ';

        for (int i = 0; i < Length; i++)
        {

```

```

        s += this[i].ToString();
        if (i != Length - 1) s += ' ';
    }

    return s;
}

public int CompareTo(object obj)
{
    if (!(obj is IVectorable))
    {
        throw new Exception("Можно сравнить только объекты типа
IVectorable");
    }

    IVectorable other = obj as IVectorable;

    if (Length < other.Length) return -1;
    if (Length > other.Length) return 1;
    return 0;
}

public override bool Equals(object obj)
{
    if (!(obj is IVectorable))
    {
        throw new Exception("Можно сравнивать только объекты типа
IVectorable");
    }

    IVectorable other = obj as IVectorable;

    if (Length != other.Length) return false;

    for (int i = 0; i < Length; i++)
    {
        if (this[i] != other[i]) return false;
    }

    return true;
}

public object Clone()
{

```

```

        LinkedListVector clone = new LinkedListVector(Length);

        for (int i = 0; i < Length; i++)
        {
            clone[i] = this[i];
        }

        return clone;
    }
}

using System.ComponentModel;

namespace Lab07
{
    partial class Stream
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }

            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>

```

```

private void InitializeComponent()
{
    this.button2 = new System.Windows.Forms.Button();
    this.button1 = new System.Windows.Forms.Button();
    this.label2 = new System.Windows.Forms.Label();
    this.label1 = new System.Windows.Forms.Label();
    this.button3 = new System.Windows.Forms.Button();
    this.button4 = new System.Windows.Forms.Button();
    this.SuspendLayout();
    //
    // button2
    //
    this.button2.Location = new System.Drawing.Point(29, 35);
    this.button2.Margin = new System.Windows.Forms.Padding(2);
    this.button2.Name = "button2";
    this.button2.Size = new System.Drawing.Size(256, 37);
    this.button2.TabIndex = 16;
    this.button2.Text = "Байтовый поток";
    this.button2.UseVisualStyleBackColor = true;
    this.button2.Click += new
System.EventHandler(this.button2_Click);
    //
    // button1
    //
    this.button1.Location = new System.Drawing.Point(29, 76);
    this.button1.Margin = new System.Windows.Forms.Padding(2);
    this.button1.Name = "button1";
    this.button1.Size = new System.Drawing.Size(256, 37);
    this.button1.TabIndex = 17;
    this.button1.Text = "Символьный поток";
    this.button1.UseVisualStyleBackColor = true;
    this.button1.Click += new
System.EventHandler(this.button1_Click);
    //
    // label2
    //
    this.label2.AutoSize = true;
    this.label2.Font = new System.Drawing.Font("Microsoft Sans
Serif", 13.8F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte) 204));
    this.label2.Location = new System.Drawing.Point(118, 9);
    this.label2.Margin = new System.Windows.Forms.Padding(2, 0, 2,
0);
    this.label2.Name = "label2";

```



```

        this.label2.Size = new System.Drawing.Size(76, 24);
        this.label2.TabIndex = 32;
        this.label2.Text = "Запись";
        //
        // label1
        //
        this.label1.AutoSize = true;
        this.label1.Font = new System.Drawing.Font("Microsoft Sans
Serif", 13.8F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.label1.Location = new System.Drawing.Point(119, 139);
        this.label1.Margin = new System.Windows.Forms.Padding(2, 0, 2,
0);

        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(75, 24);
        this.label1.TabIndex = 35;
        this.label1.Text = "Чтение";
        //
        // button3
        //
        this.button3.Location = new System.Drawing.Point(30, 206);
        this.button3.Margin = new System.Windows.Forms.Padding(2);
        this.button3.Name = "button3";
        this.button3.Size = new System.Drawing.Size(256, 37);
        this.button3.TabIndex = 34;
        this.button3.Text = "Символьный поток";
        this.button3.UseVisualStyleBackColor = true;
        this.button3.Click += new
System.EventHandler(this.button3_Click);
        //
        // button4
        //
        this.button4.Location = new System.Drawing.Point(30, 165);
        this.button4.Margin = new System.Windows.Forms.Padding(2);
        this.button4.Name = "button4";
        this.button4.Size = new System.Drawing.Size(256, 37);
        this.button4.TabIndex = 33;
        this.button4.Text = "Байтовый поток";
        this.button4.UseVisualStyleBackColor = true;
        this.button4.Click += new
System.EventHandler(this.button4_Click);
        //
        // Stream
        //

```

```

        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.AutoSize = true;
        this.ClientSize = new System.Drawing.Size(325, 277);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.button3);
        this.Controls.Add(this.button4);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.button1);
        this.Controls.Add(this.button2);
        this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedSingle;
        this.MaximizeBox = false;
        this.MaximumSize = new System.Drawing.Size(331, 306);
        this.MinimizeBox = false;
        this.MinimumSize = new System.Drawing.Size(331, 306);
        this.Name = "Stream";
        this.Text = "Stream";
        this.ResumeLayout(false);
        this.PerformLayout();
    }

    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Button button3;
    private System.Windows.Forms.Button button4;

    private System.Windows.Forms.Label label2;

    private System.Windows.Forms.Button button2;
    private System.Windows.Forms.Button button1;

    #endregion
}
}
using System;
using System.Collections.Generic;
using System.IO;
using System.Windows.Forms;

namespace Lab07
{
    public partial class Stream : Form
    {
        public Stream()

```

```

{
    InitializeComponent();
}

private void button2_Click(object sender, EventArgs e)
{
    Form1 owner = (Form1)this.Owner;
    string path = "./vectors.bin";

    List<IVectorable> vectors = owner.vectors;

    if (File.Exists(path)) File.Delete(path);

    using (FileStream fs = new FileStream(path, FileMode.Append,
FileAccess.Write))
    {
        Vectors.WriteVectors(fs, vectors);
    }

    MessageBox.Show($"Запись векторов в файл `{path}` выполнена",
"Успех!");

    owner.vectors = new List<IVectorable>();
}

private void button4_Click(object sender, EventArgs e)
{
    Form1 owner = (Form1)this.Owner;
    string path = "./vectors.bin";

    try
    {
        using (FileStream fs = new FileStream(path, FileMode.Open))
        {
            owner.vectors = Vectors.ReadVectors(fs);
        }
    }
    catch
    {
        MessageBox.Show($"Файл пуст!", "Ошибка!");
        return;
    }
}

```

```

        MessageBox.Show($"Чтение векторов из файла `{path}` выполнено",
"Успех!");
    }

    private void button1_Click(object sender, EventArgs e)
    {
        // write
        Form1 owner = (Form1)this.Owner;
        string path = "./vectors.txt";

        if (File.Exists(path)) File.Delete(path);

        using (TextWriter w = File.AppendText(path))
        {
            for (int i = 0; i < owner.vectors.Count; i++)
            {
                Vectors.WriteVector(w, owner.vectors[i]);
            }

            MessageBox.Show($"Запись векторов в файл `{path}` выполнена",
"Успех!");
        }

        owner.vectors = new List<IVectorable>();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        // read
        Form1 owner = (Form1)this.Owner;
        string path = "./vectors.txt";

        TextReader r = File.OpenText(path);
        int n = r.ReadToEnd().Split('\n').Length;
        r = File.OpenText(path);
        List<IVectorable> vectorsRead = new List<IVectorable>();
        for (int i = 0; i < n - 1; i++)
        {
            vectorsRead.Add(Vectors.ReadVector(r));
        }
        r.Close();

        owner.vectors = vectorsRead;
    }

```

```

        MessageBox.Show($"Чтение векторов из файла `{path}` выполнено",
"Успех!");
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Lab07
{
    public partial class Form1 : Form
    {
        public List<IVectorable> vectors = new List<IVectorable>();
        public Form1()
        {
            InitializeComponent();

            private void button1_Click(object sender, EventArgs e)
            {
                try
                {
                    IVectorable vec = new
LinkedListVector(Convert.ToInt32(textBox1.Text));
                    vectors.Add(vec);
                    listBox1.Items.Add(vectors.Count + "\tLinkedListVector\t" +
vec.ToString());
                }
                catch
                {
                    MessageBox.Show("Длина вектора введена неправильно",
"Ошибка!");
                }
            }

            private void button11_Click(object sender, EventArgs e)
            {
                try

```

```

        {
            IVectorable vec = new
ArrayVector(Convert.ToInt32(textBox1.Text));
            vectors.Add(vec);
            listBox1.Items.Add(vectors.Count + "\tArrayVector\t" +
vec.ToString());
        }
        catch
        {
            MessageBox.Show("Длина вектора введена неправильно",
"Ошибка!");
        }
    }

private void button2_Click(object sender, EventArgs e)
{
    if (vectors.Count == 0)
    {
        MessageBox.Show("Список векторов пуст!", "Внимание!");
        return;
    }

    try
    {
        int idx = listBox1.SelectedIndex;
        vectors.RemoveAt(idx - 1);
        listBox1.Items.RemoveAt(idx);
        for (int i = idx; i <= vectors.Count; i++)
        {
            string str = (string)listBox1.Items[i];
            int k = Convert.ToInt32((str.Split())[0]) - 1;
            str = Convert.ToString(k) + str.Substring(1);
            listBox1.Items.RemoveAt(i);
            listBox1.Items.Insert(i, str);
        }
    }
    catch
    {
        MessageBox.Show("Выберете вектор для удаления", "Ошибка!");
    }
}

private void button3_Click(object sender, EventArgs e)
{

```

```

        if (vectors.Count == 0)
        {
            MessageBox.Show("Список векторов пуст!", "Внимание!");
            return;
        }

        try
        {
            int n = listBox1.SelectedIndex;
            if (vectors[n - 1] is ArrayVector)
            {
                vectors.Add((IVectorable) ((ArrayVector)vectors[n - 1]).Clone());
                listBox1.Items.Add(vectors.Count + "\tArrayVector\t" + vectors[n - 1].ToString());
            }
            else
            {
                vectors.Add((IVectorable) ((LinkedListVector)vectors[n - 1]).Clone());
                listBox1.Items.Add(vectors.Count + "\tLinkedListVector\t" + vectors[n - 1].ToString());
            }
        }
        catch
        {
            MessageBox.Show("Выберете вектор для удаления", "Ошибка!");
        }
    }

    private void button6_Click(object sender, EventArgs e)
    {
        if (vectors.Count == 0)
        {
            MessageBox.Show("Список векторов пуст!", "Внимание!");
            return;
        }

        for (int i = 0; i < vectors.Count - 1; i++)
        {
            for (int j = i + 1; j < vectors.Count; j++)
            {
                int compareResult;

```

```

        if (vectors[i] is ArrayVector)
        {
            compareResult = (vectors[i] as
ArrayVector).CompareTo(vectors[j]);
        }
        else
        {
            compareResult = (vectors[i] as
LinkedListVector).CompareTo(vectors[j]);
        }

        if (compareResult > 0)
        {
            IVectorable tmp = vectors[i];
            vectors[i] = vectors[j];
            vectors[j] = tmp;
        }
    }

    ClearVectorsBox();
    ShowVectorsBox();
}

private void button4_Click(object sender, EventArgs e)
{
    if (vectors.Count == 0)
    {
        MessageBox.Show("Список векторов пуст!", "Внимание!");
        return;
    }

    IVectorable tmp;
    VectorsComparer comparer = new VectorsComparer();

    for (int i = 0; i < vectors.Count - 1; i++)
    {
        for (int j = i + 1; j < vectors.Count; j++)
        {
            if (comparer.Compare(vectors[i], vectors[j]) > 0)
            {
                tmp = vectors[j];
                vectors[j] = vectors[i];
                vectors[i] = tmp;
            }
        }
    }
}

```



```

        }
    }
}

ClearVectorsBox();
ShowVectorsBox();
}

private void button5_Click(object sender, EventArgs e)
{
    if (vectors.Count == 0)
    {
        MessageBox.Show("Список векторов пуст!", "Внимание!");
        return;
    }

    int idx = listBox1.SelectedIndex;

    if (idx == 0)
    {
        MessageBox.Show("Выберете вектор для расчета модуля",
"Ошибка!");
        return;
    }

    try
    {
        MessageBox.Show($"Вектор: {vectors[idx - 1]}, модуль:
{vectors[idx - 1].GetNorm()}", "Ответ!");
    }
    catch
    {
        MessageBox.Show("Выберете вектор для удаления", "Ошибка!");
    }
}

private void button12_Click(object sender, EventArgs e)
{
    if (vectors.Count == 0)
    {
        MessageBox.Show("Список векторов пуст!", "Внимание!");
        return;
    }
}

```

```

        int idx = listBox1.SelectedIndex;

        if (idx == 0)
        {
            MessageBox.Show("Выберете вектор для расчета хэш-кода",
"Ошибка!");
            return;
        }

        try
        {
            int hashCode = vectors[idx - 1].GetHashCode();
            MessageBox.Show($"Вектор: {vectors[idx - 1]}, хэш-код (hex):
0x{Convert.ToString(hashCode, 16)}", "Ответ!");
        }
        catch
        {
            MessageBox.Show("Выберете вектор для удаления", "Ошибка!");
        }
    }

    private void button10_Click(object sender, EventArgs e)
    {
        Stream form = new Stream();
        form.Owner = this;
        form.ShowDialog();

        ClearVectorsBox();
        ShowVectorsBox();
    }

    private void ClearVectorsBox()
    {
        while (listBox1.Items.Count > 1)
        {
            listBox1.Items.RemoveAt(1);
        }
    }

    private void ShowVectorsBox()
    {
        for (int i = 0; i < vectors.Count; i++)
        {
            if (vectors[i] is ArrayVector)

```

```

        {
            listBox1.Items.Add((i + 1).ToString() + "\tArrayVector\t"
+ vectors[i].ToString());
        }
        else
        {
            listBox1.Items.Add((i + 1).ToString() +
"\tLinkedListVector\t" + vectors[i].ToString());
        }
    }
}
}
namespace Lab07
{
    partial class Form1
    {
        /// <summary>
        /// Требуется переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен
        быть удален; иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Код, автоматически созданный конструктором форм Windows

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {

```

```

this.listBox1 = new System.Windows.Forms.ListBox();
this.label5 = new System.Windows.Forms.Label();
this.label11 = new System.Windows.Forms.Label();
this.button2 = new System.Windows.Forms.Button();
this.button3 = new System.Windows.Forms.Button();
this.button6 = new System.Windows.Forms.Button();
this.button9 = new System.Windows.Forms.Button();
this.button10 = new System.Windows.Forms.Button();
this.label7 = new System.Windows.Forms.Label();
this.label8 = new System.Windows.Forms.Label();
this.label9 = new System.Windows.Forms.Label();
this.textBox1 = new System.Windows.Forms.TextBox();
this.label2 = new System.Windows.Forms.Label();
this.button11 = new System.Windows.Forms.Button();
this.button1 = new System.Windows.Forms.Button();
this.button4 = new System.Windows.Forms.Button();
this.button5 = new System.Windows.Forms.Button();
this.button12 = new System.Windows.Forms.Button();
this.SuspendLayout();
//
// listBox1
//
this.listBox1.FormattingEnabled = true;
this.listBox1.Items.AddRange(new object[] { "№\tТип\t\tВектор"
});

this.listBox1.Location = new System.Drawing.Point(29, 224);
this.listBox1.Margin = new System.Windows.Forms.Padding(2);
this.listBox1.Name = "listBox1";
this.listBox1.Size = new System.Drawing.Size(294, 186);
this.listBox1.TabIndex = 6;
//
// label5
//
this.label5.AutoSize = true;
this.label5.Font = new System.Drawing.Font("Microsoft Sans
Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.label5.Location = new System.Drawing.Point(27, 64);
this.label5.Margin = new System.Windows.Forms.Padding(2, 0, 2,
0);

this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(237, 20);
this.label5.TabIndex = 10;
this.label5.Text = "Добавление вектора в список";

```

```

//
// label1
//
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Microsoft Sans
Serif", 12F);
this.label1.Location = new System.Drawing.Point(27, 202);
this.label1.Margin = new System.Windows.Forms.Padding(2, 0, 2,
0);

this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(138, 20);
this.label1.TabIndex = 12;
this.label1.Text = "Список векторов";
//
// button2
//
this.button2.Location = new System.Drawing.Point(327, 224);
this.button2.Margin = new System.Windows.Forms.Padding(2);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(89, 31);
this.button2.TabIndex = 15;
this.button2.Text = "Удалить";
this.button2.UseVisualStyleBackColor = true;
this.button2.Click += new
System.EventHandler(this.button2_Click);
//
// button3
//
this.button3.Location = new System.Drawing.Point(327, 259);
this.button3.Margin = new System.Windows.Forms.Padding(2);
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(89, 31);
this.button3.TabIndex = 16;
this.button3.Text = "Клонировать";
this.button3.UseVisualStyleBackColor = true;
this.button3.Click += new
System.EventHandler(this.button3_Click);
//
// button6
//
this.button6.Location = new System.Drawing.Point(29, 414);
this.button6.Margin = new System.Windows.Forms.Padding(2);
this.button6.Name = "button6";
this.button6.Size = new System.Drawing.Size(145, 36);

```

```

        this.button6.TabIndex = 19;
        this.button6.Text = "Сортировка по длине";
        this.button6.UseVisualStyleBackColor = true;
        this.button6.Click += new
System.EventHandler(this.button6_Click);
        //
        // button9
        //
        this.button9.Enabled = false;
        this.button9.Location = new System.Drawing.Point(29, 454);
        this.button9.Margin = new System.Windows.Forms.Padding(2);
        this.button9.Name = "button9";
        this.button9.Size = new System.Drawing.Size(145, 36);
        this.button9.TabIndex = 22;
        this.button9.Text = "Операции с 2 векторами";
        this.button9.UseVisualStyleBackColor = true;
        //
        // button10
        //
        this.button10.Location = new System.Drawing.Point(178, 454);
        this.button10.Margin = new System.Windows.Forms.Padding(2);
        this.button10.Name = "button10";
        this.button10.Size = new System.Drawing.Size(145, 36);
        this.button10.TabIndex = 23;
        this.button10.Text = "Работа с файлами";
        this.button10.UseVisualStyleBackColor = true;
        this.button10.Click += new
System.EventHandler(this.button10_Click);
        //
        // label7
        //
        this.label7.AutoSize = true;
        this.label7.Location = new System.Drawing.Point(416, 270);
        this.label7.Margin = new System.Windows.Forms.Padding(2, 0, 2,
0);
        this.label7.Name = "label7";
        this.label7.Size = new System.Drawing.Size(0, 13);
        this.label7.TabIndex = 26;
        //
        // label8
        //
        this.label8.AutoSize = true;

```

```

        this.label8.Font = new System.Drawing.Font("Microsoft Sans
Serif", 16.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.label8.Location = new System.Drawing.Point(25, 0);
        this.label8.Margin = new System.Windows.Forms.Padding(2, 0, 2,
0);

        this.label8.Name = "label8";
        this.label8.Size = new System.Drawing.Size(276, 26);
        this.label8.TabIndex = 29;
        this.label8.Text = "Лабораторная работа №7";
        //
        // label9
        //
        this.label9.AutoSize = true;
        this.label9.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
        this.label9.Font = new System.Drawing.Font("Microsoft Sans
Serif", 16.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.label9.Location = new System.Drawing.Point(25, 26);
        this.label9.Margin = new System.Windows.Forms.Padding(2, 0, 2,
0);

        this.label9.Name = "label9";
        this.label9.Size = new System.Drawing.Size(318, 26);
        this.label9.TabIndex = 30;
        this.label9.Text = "Фадеев Артем, 6101-020302D";
        //
        // textBox1
        //
        this.textBox1.Location = new System.Drawing.Point(129, 91);
        this.textBox1.Margin = new System.Windows.Forms.Padding(2);
        this.textBox1.Name = "textBox1";
        this.textBox1.Size = new System.Drawing.Size(91, 20);
        this.textBox1.TabIndex = 0;
        this.textBox1.Text = "5";
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Font = new System.Drawing.Font("Microsoft Sans
Serif", 9F);
        this.label2.Location = new System.Drawing.Point(27, 93);
        this.label2.Margin = new System.Windows.Forms.Padding(2, 0, 2,
0);

        this.label2.Name = "label2";

```

```

        this.label2.Size = new System.Drawing.Size(98, 15);
        this.label2.TabIndex = 5;
        this.label2.Text = "Длина вектора:";
        //
        // button11
        //
        this.button11.Location = new System.Drawing.Point(29, 143);
        this.button11.Margin = new System.Windows.Forms.Padding(2);
        this.button11.Name = "button11";
        this.button11.Size = new System.Drawing.Size(191, 24);
        this.button11.TabIndex = 35;
        this.button11.Text = "Добавить ArrayVector";
        this.button11.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
        this.button11.UseVisualStyleBackColor = true;
        this.button11.Click +=
new
System.EventHandler(this.button11_Click);
        //
        // button1
        //
        this.button1.BackColor =
System.Drawing.SystemColors.ControlLightLight;
        this.button1.Location = new System.Drawing.Point(29, 115);
        this.button1.Margin = new System.Windows.Forms.Padding(2);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(191, 24);
        this.button1.TabIndex = 34;
        this.button1.Text = "Добавить LinkedListVector";
        this.button1.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
        this.button1.UseVisualStyleBackColor = false;
        this.button1.Click +=
new
System.EventHandler(this.button1_Click);
        //
        // button4
        //
        this.button4.Location = new System.Drawing.Point(178, 414);
        this.button4.Margin = new System.Windows.Forms.Padding(2);
        this.button4.Name = "button4";
        this.button4.Size = new System.Drawing.Size(145, 36);
        this.button4.TabIndex = 36;
        this.button4.Text = "Сортировка по модулю";
        this.button4.UseVisualStyleBackColor = true;

```



```

        this.button4.Click                                     +=          new
System.EventHandler(this.button4_Click);
        //
        // button5
        //
        this.button5.Location = new System.Drawing.Point(327, 318);
        this.button5.Margin = new System.Windows.Forms.Padding(2);
        this.button5.Name = "button5";
        this.button5.Size = new System.Drawing.Size(89, 31);
        this.button5.TabIndex = 37;
        this.button5.Text = "Модуль";
        this.button5.UseVisualStyleBackColor = true;
        this.button5.Click                                     +=          new
System.EventHandler(this.button5_Click);
        //
        // button12
        //
        this.button12.Location = new System.Drawing.Point(327, 353);
        this.button12.Margin = new System.Windows.Forms.Padding(2);
        this.button12.Name = "button12";
        this.button12.Size = new System.Drawing.Size(89, 31);
        this.button12.TabIndex = 38;
        this.button12.Text = "HashCode";
        this.button12.UseVisualStyleBackColor = true;
        this.button12.Click                                     +=          new
System.EventHandler(this.button12_Click);
        //
        // Form1
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(442, 567);
        this.Controls.Add(this.button12);
        this.Controls.Add(this.button5);
        this.Controls.Add(this.button4);
        this.Controls.Add(this.button11);
        this.Controls.Add(this.button1);
        this.Controls.Add(this.label9);
        this.Controls.Add(this.label8);
        this.Controls.Add(this.label7);
        this.Controls.Add(this.button10);
        this.Controls.Add(this.button9);
        this.Controls.Add(this.button6);
        this.Controls.Add(this.button3);

```

```

        this.Controls.Add(this.button2);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.label5);
        this.Controls.Add(this.listBox1);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.textBox1);
        this.Margin = new System.Windows.Forms.Padding(2);
        this.Name = "Form1";
        this.Text = "Form1";
        this.ResumeLayout(false);
        this.PerformLayout();
    }

    private System.Windows.Forms.Button button12;

    private System.Windows.Forms.Button button5;

    private System.Windows.Forms.TextBox textBox1;
    private System.Windows.Forms.Label label2;

    private System.Windows.Forms.Button button11;

    #endregion

    private System.Windows.Forms.Label label5;
    private System.Windows.Forms.Button button1;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Button button2;
    private System.Windows.Forms.Button button3;
    private System.Windows.Forms.Button button4;
    private System.Windows.Forms.Button button6;
    private System.Windows.Forms.Button button9;
    private System.Windows.Forms.Button button10;
    private System.Windows.Forms.ListBox listBox1;
    private System.Windows.Forms.Label label7;
    private System.Windows.Forms.Label label8;
    private System.Windows.Forms.Label label9;
    }
}

using System;
using System.Windows.Forms;

namespace Lab07
{

```

```

static class Program
{
    /// <summary>
    /// Главная точка входа для приложения.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1());
    }
}

```

**Лабораторная работа №7**  
**Фадеев Артем, 6101-020302D**

**Добавление вектора**

Длина вектора:

**Список векторов**

№	Тип	Вектор
1	LinkedListVector	5 74 39 57 96 46

Рисунок 1 – Главное меню программы

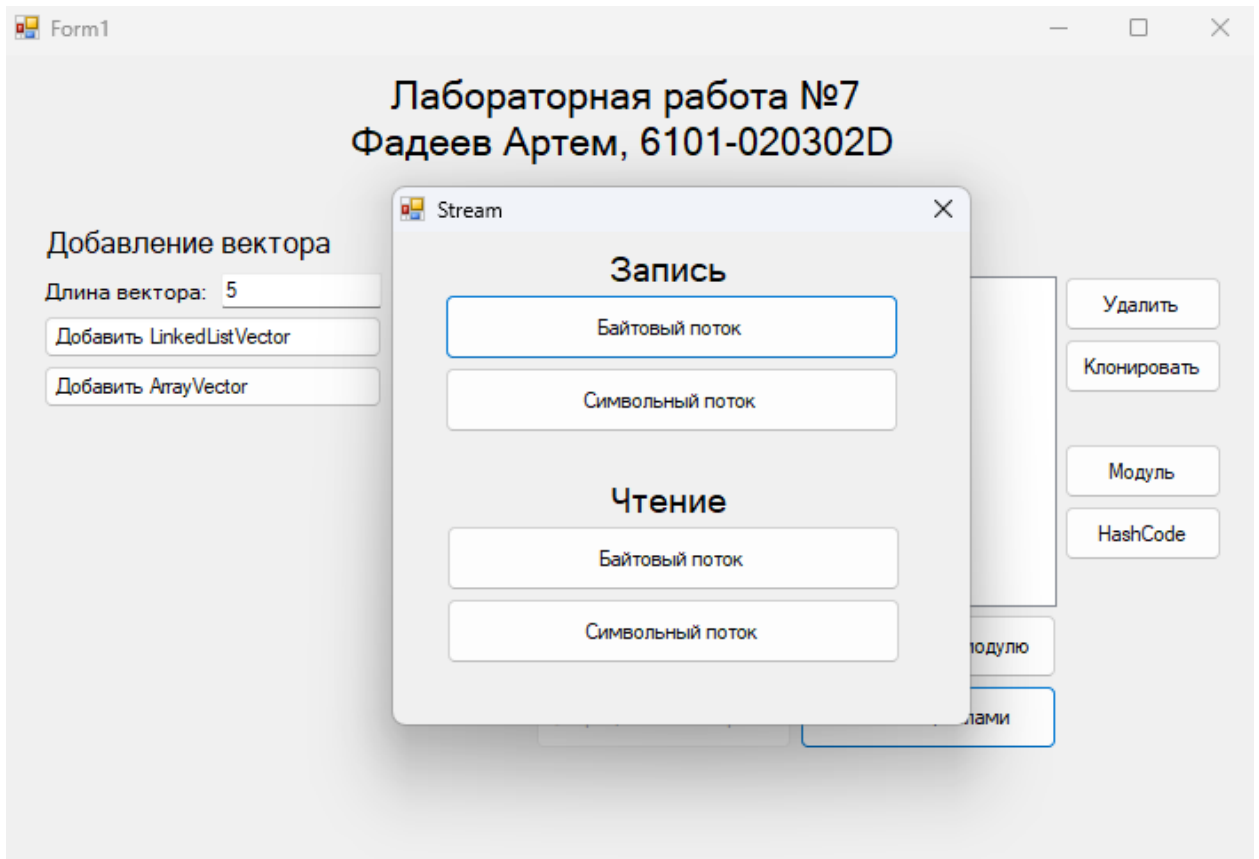


Рисунок 2 – Меню работы с потоками

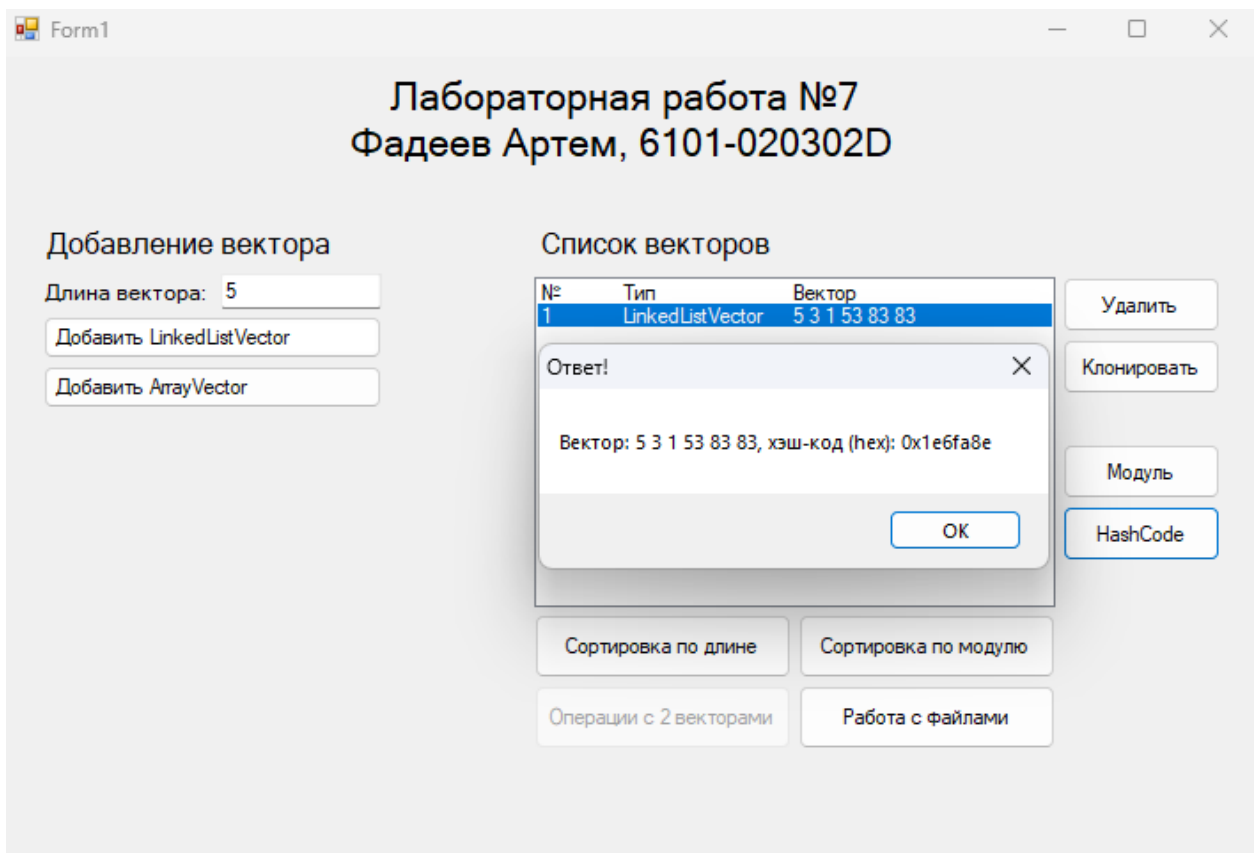


Рисунок 3 – Расчитанный хэш-код для вектора

## ВЫВОДЫ

В лабораторной работе были использованы конструкции языка:

- форматированный вывод информации на консоль;
- оператор switch;
- условные операторы;
- функции;
- классы;
- конструкторы класса;
- поля класса;
- статические и динамические методы класса;
- интерфейсы;
- индексаторы;
- байтовые и символьные потоки;
- делегаты;
- конструктор Windows-приложений Windows Forms;
- конструкция try-catch.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

- 1 Павловская Т.А. С#. Программирование на языке высокого уровня. Учебник для вузов [Текст]/Т.А. Павловская. – СПб.: Питер, 2007. – 432 с.