

Московский авиационный институт
(национальный исследовательский университет)

**Институт информационных технологий и
прикладной математики**

Кафедра вычислительной математики и программирования

**Курсовой проект по курсу “Объектно-
ориентированное программирование”**

Работу выполнила:

М8О-209Б-19 Офицера Т.И. _____

Группа

ФИО

Подпись

Руководитель: _____/Кузнецова С.В./

Подпись:

Дата: 22 декабря 2020

Тема курсового проекта

Двухточечный кроссовер (алгоритм №24)

Постановка задачи

Реализация алгоритма двухточечного кроссовера на языке C# и его визуализация с использованием Windows Forms.

Программное и аппаратное обеспечение

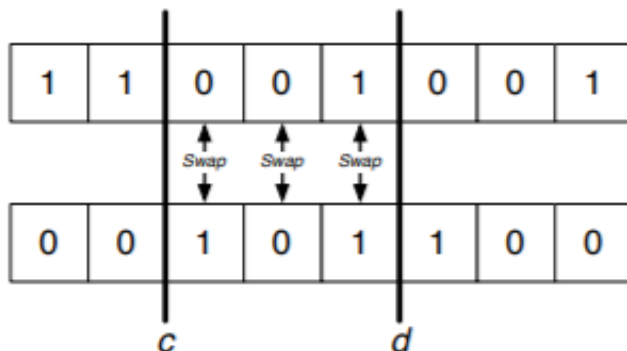
ОС: Windows 10

Среда разработки: Microsoft Visual Studio (Community) 2019

Описание алгоритма

Кроссовер – это генетический оператор, за счет которого производится обмен генетическим материалом между особями. Он моделирует процесс скрещивания особей.

Пусть имеются две родительские хромосомы длины l . Кроссовер случайным образом выбирает два числа c и d из промежутка от 1 до l включительно и меняет индексы между c и d , как показано на рисунке:



В случае, когда $c=d$, происходит пустой кроссовер (с вероятностью $\frac{1}{l}$)

Псевдокод:

$v \leftarrow$ first vector $\langle v_1, v_2, \dots, v_l \rangle$ to be crossed over

$w \leftarrow$ second vector $\langle w_1, w_2, \dots, w_l \rangle$ to be crossed over

$c \leftarrow$ random integer chosen uniformly from 1 to l inclusive

$d \leftarrow$ random integer chosen uniformly from 1 to l inclusive

if $c > d$ then

swap c and d

if $c \neq d$ then

for i from c to $d-1$ do

swap the values of v_i and w_i
return v and w

Реализация алгоритма на C#

Модель кроссовера:

CrossoverModel.cs:

```
public class CrossoverModel
{
    private int size;
    private int point1;
    private int point2;
    private int[] chromosome1;
    private int[] chromosome2;
    private ArrayList listners;

    public CrossoverModel()
    {
        this.size = 0;
        this.point1 = 0;
        this.point2 = 0;
        this.chromosome1 = null;
        this.chromosome2 = null;
        this.listners = new ArrayList();
    }

    public void Swap(ref int a, ref int b)
    {
        int temp;
        temp = a;
        a = b;
        b = temp;
    }

    public void DoCrossover(int p1, int p2, int[] ch1, int[] ch2)
    {
        this.chromosome1 = ch1;
        this.chromosome2 = ch2;
        this.point1 = p1;
        this.point2 = p2;
        if (p1 > p2) Swap(ref p1, ref p2);
        if (p1 != p2)
        {
            for (int i = p1; i < p2; i++)
            {
                Swap(ref ch1[i], ref ch2[i]);
            }
        }
    }

    public void Register(IObserver o)
    {
        this.listners.Add(o);
    }

    public void ClearListners()
    {
        this.listners.Clear();
    }
}
```

Observer.cs:

```
public interface IObserver
```

```

{
    void UpdateState();
}

```

Реализация интерфейса на C#

Приложение имеет 3 окна. Главная форма позволяет посмотреть информацию об алгоритме и информацию о самой работе, а также дает выбор в каком формате пользователь хочет воспользоваться алгоритмом: можно выбрать вариант с полной настройкой алгоритма или анимированный вариант.

Program.cs:

```

static class Program
{
    [STAThread]
    static void Main()
    {
        MainForm view3 = new MainForm();
        Application.Run(view3);
    }
}

```

MainForm.cs:

```

public partial class MainForm : Form
{
    //change language
    public void Change()
    {
        this.Text = rs.GetString("mainform");
        button1.Text = rs.GetString("Mainbutton1.Text");
        button2.Text = rs.GetString("Mainbutton2.Text");
        button3.Text = rs.GetString("button6.Text");
        languageToolStripMenuItem.Text =
rs.GetString("languageToolStripMenuItem.Text");
        algorithmToolStripMenuItem.Text =
rs.GetString("algorithmToolStripMenuItem.Text");
        courseWorkInfoToolStripMenuItem1.Text =
rs.GetString("courseWorkInfoToolStripMenuItem1.Text");
    }
    ResourceManager rs = null;
    public MainForm()
    {
        InitializeComponent();
        rs = new ResourceManager("course_work.Eng", typeof(MainForm).Assembly);
        Change();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        CrossoverModel model = new CrossoverModel();
        Form1 view1 = new Form1(model);
        view1.Show();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Close();
    }
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    CrossoverModel model = new CrossoverModel();
    Form2 view2 = new Form2(model);
    view2.Show();
}

private void algorithmToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show(rs.GetString("algoinfo"));
}

private void courseWorkInfoToolStripMenuItem1_Click(object sender, EventArgs e)
{
    MessageBox.Show(rs.GetString("workinfo"));
}

private void englishToolStripMenuItem_Click(object sender, EventArgs e)
{
    rs = new ResourceManager("course_work.Eng", typeof(Form1).Assembly);
    Change();
}

private void русскийToolStripMenuItem_Click(object sender, EventArgs e)
{
    rs = new ResourceManager("course_work.Rus", typeof(Form1).Assembly);
    Change();
}
}

```

При выборе первого варианта открывается окно работы с алгоритмом. При вызове помощи в меню появится MessageBox, содержащий информацию о работе данной формы. Программа позволяет ввести хромосомы самому или указать длину необходимых хромосом, и они будут сгенерированы случайным образом. В любом случае хромосомы отобразятся в textbox. Далее дается возможность выбрать способ обозначения точек кроссовера: случайные точки или же вводимые пользователем. После нажатия на кнопку кроссовера результат работы алгоритма выводится в richtextbox.

Form1Controller.cs:

```

class Form1Controller
{
    private Form1 view;
    private CrossoverModel model;
    public Form1Controller(Form1 view, CrossoverModel model)
    {
        this.view = view;
        this.model = model;
    }
    public void Go(int p1, int p2, int[] ch1, int[] ch2)
    {
        this.model.DoCrossover(p1, p2, ch1, ch2);
    }
}

```

Form1.cs:

```

public partial class Form1 : Form, IObservable
{
    //change the language of program
    public void Change()
    {
        label11.Text = rs.GetString("label11.Text");
        helpToolStripMenuItem.Text = rs.GetString("helpToolStripMenuItem.Text");
        languageToolStripMenuItem.Text =
rs.GetString("languageToolStripMenuItem.Text");
        label13.Text = rs.GetString("label13.Text");
        label14.Text = rs.GetString("label14.Text");
        label15.Text = rs.GetString("label15.Text");
        label17.Text = rs.GetString("label17.Text");
        button1.Text = rs.GetString("button1.Text");
        button2.Text = rs.GetString("button2.Text");
        button4.Text = rs.GetString("button4.Text");
        button5.Text = rs.GetString("button5.Text");
        button6.Text = rs.GetString("button6.Text");
        this.Text = rs.GetString("Form1Name");
    }
    ResourceManager rs = null;
    private Form1Controller controller;
    private CrossoverModel model;
    Random rand = new Random();
    int a, b, size;
    public Form1(CrossoverModel model)
    {
        InitializeComponent();
        rs = new ResourceManager("course_work.Eng", typeof(Form1).Assembly);
        Change();
        this.model = model;
        this.model.Register(this);
        this.controller = InjectController();
    }
    private Form1Controller InjectController()
    {
        return new Form1Controller(this, this.model);
    }
    public void UpdateState()
    {
    }

    private void button6_Click(object sender, EventArgs e)
    {
        Close();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        groupBox1.Visible = true;
    }

    private void button4_Click(object sender, EventArgs e)
    {
        //errors if something wrong
        try
        {
            //creating chromosomes
            int[] array1 = textBox1.Text.Split(' ').Select(s =>
Convert.ToInt32(s)).ToArray();
            int[] array2 = textBox2.Text.Split(' ').Select(s =>
Convert.ToInt32(s)).ToArray();
            //crossover
            controller.Go(a, b, array1, array2);
        }
        catch { }
    }
}

```

```

        //print
        for (int i = 0; i < array2.Length; i++)
        {
            richTextBox1.Text += array1[i].ToString() + " ";
        }
        richTextBox1.Text += "\n";
        for (int i = 0; i < array1.Length; i++)
        {
            richTextBox1.Text += array2[i].ToString() + " ";
        }
        richTextBox1.Text += "\n";
        for (int i = 0; i < array1.Length; i++)
        {
            //error if something no 1 or 0
            if ((array1[i] != 1 && array1[i] != 0) || (array2[i] != 1 &&
array2[i] != 0))
            {
                MessageBox.Show(rs.GetString("text6"));
                richTextBox1.Text = "";
                textBox1.Text = "";
                textBox2.Text = "";
                break;
            }
        }
        //error if different length
        catch (IndexOutOfRangeException)
        {
            MessageBox.Show(rs.GetString("text4"));
            richTextBox1.Text = "";
        }
        //other errors
        catch (Exception ex)
        {
            MessageBox.Show(rs.GetString("text5"));
            richTextBox1.Text = "";
        }
    }

private void button7_Click(object sender, EventArgs e)
{
    try
    {
        size = (int)numericUpDown1.Value;
        //stop if size ==0
        for (int i = 0; i < size; i++)
        {
            textBox1.Text += rand.Next(0, 2).ToString() + " ";
            textBox2.Text += rand.Next(0, 2).ToString() + " ";
        }
        textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - 1);
        textBox2.Text = textBox2.Text.Remove(textBox2.Text.Length - 1);
    }
    catch (Exception ex) { MessageBox.Show(rs.GetString("text3")); }
}

//clean all
private void button5_Click(object sender, EventArgs e)
{
    numericUpDown1.Value = 0;
    textBox1.Text = "";
    textBox2.Text = "";
    size = 0;
    a = 0;
    b = 0;
    textBox3.Text = "";
}

```

```

        textBox4.Text = "";
        richTextBox1.Text = "";
        groupBox1.Visible = false;
    }

    private void englishToolStripMenuItem_Click(object sender, EventArgs e)
    {
        rs = new ResourceManager("course_work.Eng", typeof(Form1).Assembly);
        Change();
    }

    private void russianToolStripMenuItem_Click(object sender, EventArgs e)
    {
        rs = new ResourceManager("course_work.Rus", typeof(Form1).Assembly);
        Change();
    }

    private void helpToolStripMenuItem_Click_1(object sender, EventArgs e)
    {
        MessageBox.Show(rs.GetString("text1"));
    }

    private void button1_Click(object sender, EventArgs e)
    {
        int[] array1 = textBox1.Text.Split(' ').Select(s =>
Convert.ToInt32(s)).ToArray();
        numericUpDown1.Value = array1.Length;
        size = (int)numericUpDown1.Value;
        textBox3.Text = rand.Next(0, size+1).ToString();
        textBox4.Text = rand.Next(0, size+1).ToString();
        a = int.Parse(textBox3.Text);
        b = int.Parse(textBox4.Text);
    }

    private void button3_Click(object sender, EventArgs e)
    {
        try
        {
            int[] array1 = textBox1.Text.Split(' ').Select(s =>
Convert.ToInt32(s)).ToArray();
            numericUpDown1.Value = array1.Length;
            size = (int)numericUpDown1.Value;
            a = int.Parse(textBox3.Text);
            b = int.Parse(textBox4.Text);
            if (a>size || b > size)
            {
                MessageBox.Show(rs.GetString("text7"));
            }
            else groupBox1.Visible = false;
        }
        catch (Exception ex) { MessageBox.Show(rs.GetString("text2")); }
    }
}

```

При выборе анимированной работы алгоритма открывается новое окно для работы с алгоритмом. При вызове помощи в меню появится MessageBox, содержащий информацию о работе данной формы. Пользователю требуется ввести число – длину хромосом, которые будут сгенерированы и нарисованы на форме после нажатия кнопки ок. хромосомы представлены в виде последовательностей пустых или

закрашенных кругов (1 и 0), двух цветов для большей наглядности при совершении кроссовера. Каждое нажатие на кнопку кроссовера производит случайный кроссовер. Также есть возможность начать работу заново и вернуться к вводу длины хромосом. Ширина окна подстраивается под введенную длину.

Form2Controller.cs:

```
class Form2Controller
{
    private Form2 view;
    private CrossoverModel model;
    public Form2Controller(Form2 view, CrossoverModel model)
    {
        this.view = view;
        this.model = model;
    }
    public void Go(int p1, int p2, int[] ch1, int[] ch2)
    {
        this.model.DoCrossover(p1, p2, ch1, ch2);
    }
}
```

Form2.cs:

```
public partial class Form2 : Form, IObservable
{
    //change language
    public void Change()
    {
        button1.Text = rs.GetString("button4.Text");
        button2.Text = rs.GetString("button6.Text");
        button4.Text = rs.GetString("button5.Text");
        languageToolStripMenuItem.Text =
rs.GetString("languageToolStripMenuItem.Text");
        infoToolStripMenuItem.Text = rs.GetString("helpToolStripMenuItem.Text");
        this.Text = rs.GetString("Form1Name");
    }
    Graphics context;
    private Form2Controller controller;
    private CrossoverModel model;
    static Random rand = new Random();
    bool randomBool;
    ResourceManager rs = null;
    private AbstractFactory factoryRed;
    private AbstractFactory factoryBlue;
    public Form2(CrossoverModel model)
    {
        InitializeComponent();
        rs = new ResourceManager("course_work.Eng", typeof(MainForm).Assembly);
        Change();
        this.model = model;
        this.model.Register(this);
        this.controller = InjectController();
    }
    private Form2Controller InjectController()
    {
        {
            return new Form2Controller(this, this.model);
        }
    }
    public void UpdateState()
    {
    }
}
```

```

List<AbstractCircle> list = new List<AbstractCircle>(); //список кругов
List<AbstractCircle> list1 = new List<AbstractCircle>();
int[] ch1; int[] ch2; int[] crossed;
//painter for random generated chromosomes
private void Painter()
{
    context = Graphics.FromHwnd(this.Handle);
    foreach (AbstractCircle krug in list)
    {
        {
            if (krug.F == true)
            {
                context.FillEllipse(new SolidBrush(krug.CircleColor()), krug.X,
krug.Y, 20, 20);
            }
            else context.DrawEllipse(new Pen(krug.CircleColor()), krug.X, krug.Y,
20, 20);
        }
    }
    foreach (AbstractCircle krug in list1)
    {
        if (krug.F == true)
        {
            context.FillEllipse(new SolidBrush(krug.CircleColor()), krug.X,
krug.Y, 20, 20);
        }
        else context.DrawEllipse(new Pen(krug.CircleColor()), krug.X, krug.Y, 20,
20);
    }
}

private void button2_Click(object sender, EventArgs e)
{
    ch1 = null; ch2 = null; crossed = null;
    list.Clear(); list1.Clear();
    Close();
}

//creating and painting chromosomes
private void button3_Click(object sender, EventArgs e)
{
    int N = (int)numericUpDown1.Value;
    ch1 = new int[N];
    ch2 = new int[N];
    crossed = new int[N];
    for (int i = 0; i < N; i++)
    {
        randomBool = rand.Next(2) == 1;
        list.Add(factoryRed.CreateCircle(randomBool, 20 + i * 50, 50));
        if (list[i].F == true) ch1[i] = 1;
        else ch1[i] = 0;
        randomBool = rand.Next(2) == 1;
        list1.Add(factoryBlue.CreateCircle(randomBool, 20 + i * 50, 80));
        if (list1[i].F == true) ch2[i] = 1;
        else ch2[i] = 0;
    }
    if (N > 5)
    {
        this.Width += (N-5)*50;
    }
    Painter();
    groupBox1.Visible = false;
    button1.Visible = true;
    button4.Visible = true;
}

//clear all for again

```

```

private void button4_Click(object sender, EventArgs e)
{
    numericUpDown1.Value = 0;
    ch1 = new int[0]; ch2 = new int[0]; crossed = new int[0];
    list.Clear(); list1.Clear();
    context.FillRectangle(new SolidBrush(this.BackColor), 0, 0, Width, Height);
    groupBox1.Visible = true;
    button1.Visible = false;
    button4.Visible = false;
    this.Width = 283;
}
//crossover
private void button1_Click(object sender, EventArgs e)
{
    int a = rand.Next(0, ch1.Length+1);
    int b = rand.Next(0, ch1.Length+1);
    controller.Go(a, b, ch1, ch2);
    for (int i = 0; i < ch1.Length; i++)
    {
        if ((i > a && i < b) || (i < a && i > b))
        {
            crossed[i] = 1;
        }
        else crossed[i] = 0;
    }
    this.Invalidate();
}
//animation
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);

    if (crossed != null)
    {
        for (int i = 0; i < ch1.Length; i++)
        {
            if (crossed[i] == 1)
            {
                for (int j = 1; j < 11; j++)
                {
                    if (list[i].F == true)
                    {
                        context.FillEllipse(new SolidBrush(this.BackColor),
list[i].X, list[i].Y + 3 * (j - 1), 20, 20);
                        context.FillEllipse(new
SolidBrush(list[i].CircleColor()), list[i].X, list[i].Y + 3 * j, 20, 20);
                    }
                    else
                    {
                        context.DrawEllipse(new Pen(this.BackColor), list[i].X,
list[i].Y + 3 * (j - 1), 20, 20);
                        context.DrawEllipse(new Pen(list[i].CircleColor()),
list[i].X, list[i].Y + 3 * j, 20, 20);
                    }
                    if (list1[i].F == true)
                    {
                        context.FillEllipse(new SolidBrush(this.BackColor),
list1[i].X, list1[i].Y - 3 * (j - 1), 20, 20);
                        context.FillEllipse(new
SolidBrush(list1[i].CircleColor()), list1[i].X, list1[i].Y - 3 * j, 20, 20);
                    }
                    else
                    {
                        context.DrawEllipse(new Pen(this.BackColor), list1[i].X,
list1[i].Y - 3 * (j - 1), 20, 20);

```

```

        context.DrawEllipse(new Pen(list1[i].CircleColor()),
list1[i].X, list1[i].Y - 3 * j, 20, 20);
    }
}
else
{
    if (list[i].F == true)
    {
        context.FillEllipse(new SolidBrush(list[i].CircleColor()),
list[i].X, list[i].Y, 20, 20);
    }
    else
    {
        context.DrawEllipse(new Pen(list[i].CircleColor()),
list[i].X, list[i].Y, 20, 20);
    }
    if (list1[i].F == true)
    {
        context.FillEllipse(new SolidBrush(list1[i].CircleColor()),
list1[i].X, list1[i].Y, 20, 20);
    }
    else
    {
        context.DrawEllipse(new Pen(list1[i].CircleColor()),
list1[i].X, list1[i].Y, 20, 20);
    }
}
}
}

private void englishToolStripMenuItem_Click(object sender, EventArgs e)
{
    rs = new ResourceManager("course_work.Eng", typeof(Form1).Assembly);
    Change();
}

private void русскийToolStripMenuItem_Click(object sender, EventArgs e)
{
    rs = new ResourceManager("course_work.Rus", typeof(Form1).Assembly);
    Change();
}

private void infoToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show(rs.GetString("textinfoform2"));
}

private void Form2_Load(object sender, EventArgs e)
{
    context = Graphics.FromHwnd(this.Handle);
    factoryRed = new FactoryRed();
    factoryBlue = new FactoryBlue();
}
}

```

Для создания кругов различных цветов использован паттерн абстрактная фабрика.

AbstractFactory.cs:

```
//creating of figures red or blue
public abstract class AbstractFactory
{
    public abstract AbstractCircle CreateCircle(bool fullen, int x, int y);
}
public class FactoryRed : AbstractFactory
{
    public override AbstractCircle CreateCircle(bool fullen, int x, int y)
    {
        return new RedCircle(fullen, x, y);
    }
}
public class FactoryBlue : AbstractFactory
{
    public override AbstractCircle CreateCircle(bool fullen, int x, int y)
    {
        return new BlueCircle(fullen, x, y);
    }
}
```

AbstractProduct.cs:

```
public abstract class AbstractCircle //abstract circle with fullness, x, y and color
{
    public bool F { set; get; }
    public int X { set; get; }
    public int Y { set; get; }
    public abstract Color CircleColor();
}
public class RedCircle : AbstractCircle
{
    public RedCircle(bool f, int x, int y)
    {
        this.F = f;
        this.X = x;
        this.Y = y;
    }
    public override Color CircleColor()
    {
        return Color.Red;
    }
}
public class BlueCircle : AbstractCircle
{
    public BlueCircle(bool f, int x, int y)
    {
        this.F = f;
        this.X = x;
        this.Y = y;
    }
    public override Color CircleColor()
    {
        return Color.Blue;
    }
}
```

Все приложение поддерживает два языка: русский и английский. Это сделано с помощью System.Resources.

Eng.resx:

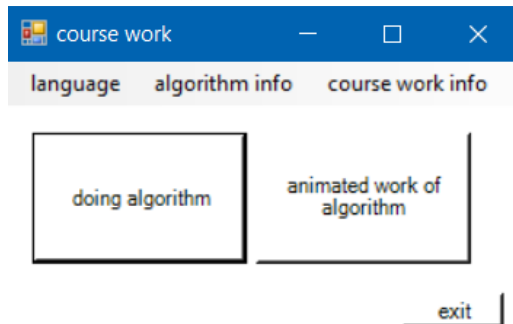
Имя	Значение	Комментарий
algorithinfo	Crossover is the Genetic Algorithm's distinguishing feature. It involves mixing and matching parts of two parents to form children. Let's say the vectors are of length l. Two-point crossover implies picking two numbers c and d, and swap the indexes between them. When c = d you get an empty crossover (with 1/l probability).	
algorithmToolStripMenuItem.T	algorithm info	
button1.Text	random	
button2.Text	with my points	
button4.Text	do crossover	
button5.Text	again	
button6.Text	exit	
courseWorkInfoToolStripMenu	course work info	
FormName	two-point crossover	
helpToolStripMenuItem.Text	help	
infoToolStripMenuItem.Text	info	
Имя	Значение	Комментарий
label1.Text	enter 1st and 2nd chromosome (1 and 0, delimited by spaces)	
label3.Text	enter length of chromosomes and press "ok" to generate random chromosome of this length	
label4.Text	new chromosomes are:	
label5.Text	1st and 2nd points	
label7.Text	choose type of crossover	
languageToolStripMenuItem.T	language	
Mainbutton1.Text	doing algorithm	
Mainbutton2.Text	animated work of algorithm	
mainform	course work	
text1	To do crossover you must enter 2 chromosomes (1 and 0, delimited by spaces) or generate them randomly (enter length and click "ok"), choose the way for generating crossover points (random or enter in special place) and press button "do crossover". The result chromosomes will appear lower. Press "again" to clear all and try next chromosomes. Press "exit" to	
Имя	Значение	Комментарий
text2	error in entered points	
text3	chromosomes can't be zero length	
text4	chromosomes must be of equal length	
text5	something wrong with format of chromosomes	
text6	chromosomes can contain only 0 and 1!	
text7	points must be less than length of chromosomes	
textinfoform2	To do crossover you must enter the length of chromosomes and press "ok". Then the program will generate random chromosomes and draw them (different colors for visibility). Every click to "do crossover" will cause new randomly generated animated crossover. to choose another length and generate new chromosomes click "again", to close the window press "exit".	
workinfo	This program is a course work on subject OOP with thesis "Realization of algorithm on C# and visualization of it". Done by Ofitserova Tatiana (student of group M8O-209B-19).	

Rus.resx:

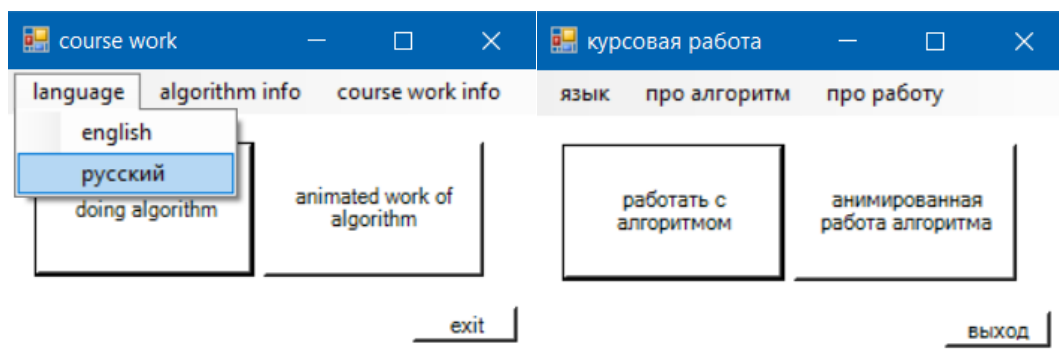
	Имя	Значение	Комментарий
▶	algoinfo	Кроссовер это отличительная черта генетического алгоритма. Он включает в себя перемешивания и сочетания частей хромосом двух родителей для получения хромосом детей. Скажем, что векторы имеют длину l. Кроссовер с двумя точками подразумевает взятие двух чисел c и d и обмен индексами между ними.в случае, если c=d происходит пустой кроссовер (с вероятностью 1/l).	
	algorithmToolStripMenuItem1	про алгоритм	
	button1.Text	случайный	
	button2.Text	с моими точками	
	button4.Text	кроссовер	
	button5.Text	еще раз	
	button6.Text	выход	
	courseWorkInfoToolStripMenuItem1	про работу	
	Form1Name	кроссовер с двумя точками	
	helpToolStripMenuItem1	помощь	
	Имя	Значение	Комментарий
	label1.Text	введите первую и вторую хромосому (1 и 0, разделенные пробелами)	
	label3.Text	введите длину хромосом и нажмите "ок" для генерации случайных хромосом заданной длины	
	label4.Text	итоговые хромосомы:	
	label5.Text	1я и вторая точки	
	label7.Text	выберите тип кроссовера	
	languageToolStripMenuItem1	язык	
	Mainbutton1.Text	работать с алгоритмом	
	Mainbutton2.Text	анимированная работа алгоритма	
	mainform	курсовая работа	
▶	text1	Для выполнения алгоритма требуется ввести две хромосомы, состоящие из нулей и единиц, разделенных пробелами (или сгенерировать случайные хромосомы путем выбора количества элементов в них и нажатия кнопки "ок"), выбрать способ генерации точек кроссовера (случайные или ввести собственные) и нажать кнопку "кроссовер". Результат появится в нижнем окне. Чтобы выйти из программы требуется нажать "выход". Чтобы очистить поля и повторить алгоритм - нажать "еще раз".	
	Имя	Значение	Комментарий
	text2	ошибка в введенных точках	
	text3	хромосома не может быть нулевой длины :(
	text4	хромосомы должны быть одной длины :(
	text5	что-то не так с форматом введенных данных	
	text6	в хромосоме могут быть только 0 и 1!	
	text7	точка не может выходить за границу хромосомы	
	textinfoform2	Для выполнения алгоритма необходимо ввести длину требуемых хромосом и нажать "ок", после чего отобразятся рандомно сгенерированные хромосомы (разных цветов для наглядности). При каждом нажатии кнопки "кроссовер" будет выполняться случайный анимированный кроссовер, при нажатии кнопки "еще раз" появится возможность выбрать новую длину и создать новые хромосомы. После нажатия "выход" окно закроется.	
▶	workinfo	Данная программа является курсовой работой по предмету Объектно-ориентированное программирование по теме "Реализация алгоритма на C# и его визуализация", выполненной Офицеровой Татьяной, студенткой группы М8О-209Б-19.	

Демонстрация функциональных возможностей

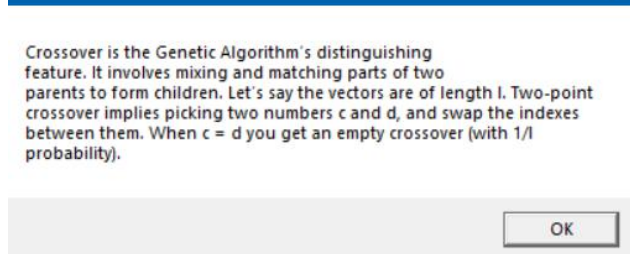
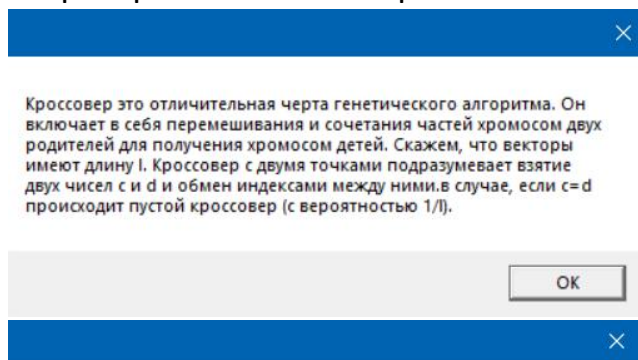
Главная форма:



Смена языка:



Инфо о работе и об алгоритме:



Данная программа является курсовой работой по предмету
Объектно-ориентированное программирование по теме
"Реализация алгоритма на C# и его визуализация", выполненной
Офицеровой Татьяной, студенткой группы М8О-209Б-19.

This program is a course work on subject OOP with tesis "Realization of
algorithm on C# and visualization of it". Done by Ofitserova Tatiana
(student of group M8O-209B-19).

OK

OK

Окно работы с алгоритмом:

two-point crossover

language help

enter 1st and 2nd chromosome (1 and 0, delimited by spaces)

choose type of crossover

☐ random ☐ with my points

new chromosomes are:

enter length of chromosomes and press "ok" to generate random chromosome of this length

Помощь:

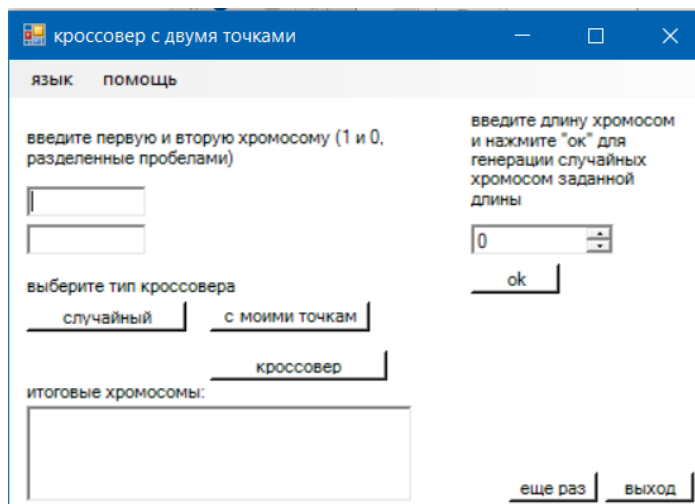
To do crossover you must enter 2 chromosomes (1 and 0, delimited by spaces) or generate them randomly (enter length and click "ok"), choose the way for generating crossover points (random or enter in special place) and press button "do crossover". The result chromosomes will appear lower. Press "again" to clear all and try next chromosomes. Press "exit" to close window.

OK

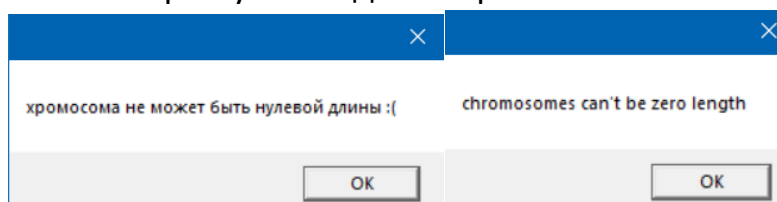
Для выполнения алгоритма требуется ввести две хромосомы, состоящие из нулей и единиц, разделенных пробелами (или сгенерировать случайные хромосомы путем выбора количества элементов в них и нажатия кнопки "ок"), выбрать способ генерации точек кроссовера (случайные или ввести собственные) и нажать кнопку "кроссовер". Результат появится в нижнем окне. Чтобы выйти из программы требуется нажать "выход". Чтобы очистить поля и повторить алгоритм - нажать "еще раз".

OK

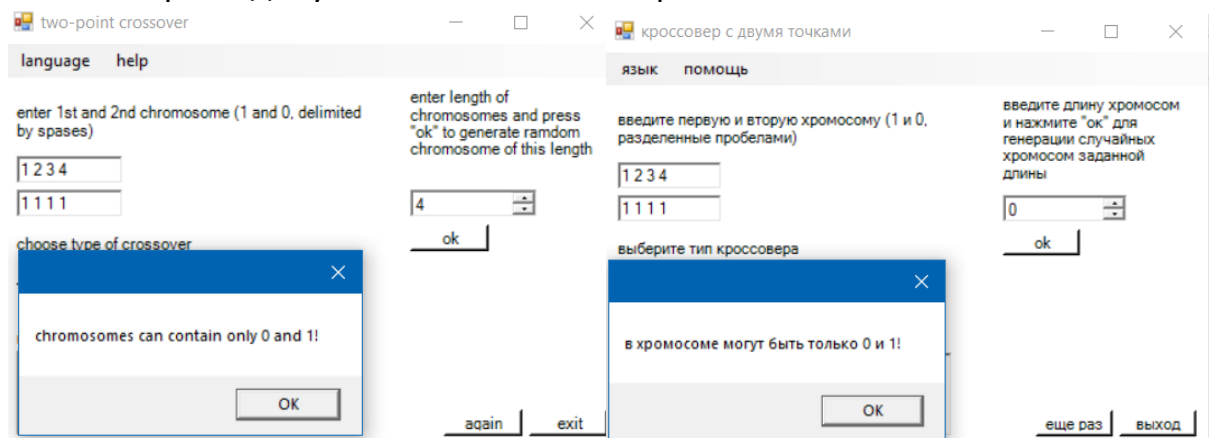
Смена языка:



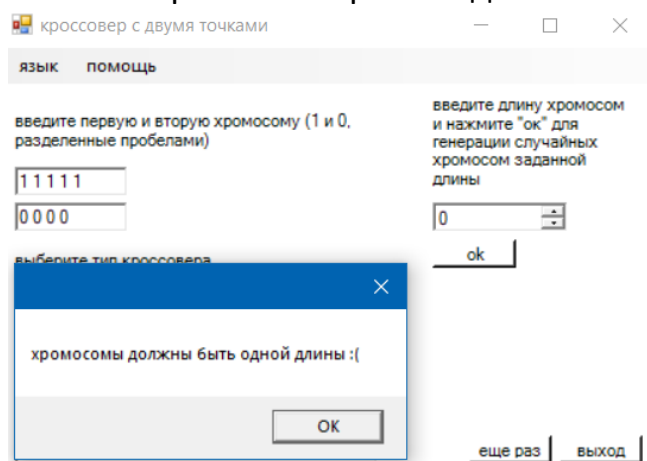
Ошибка при нулевой длине хромосом:



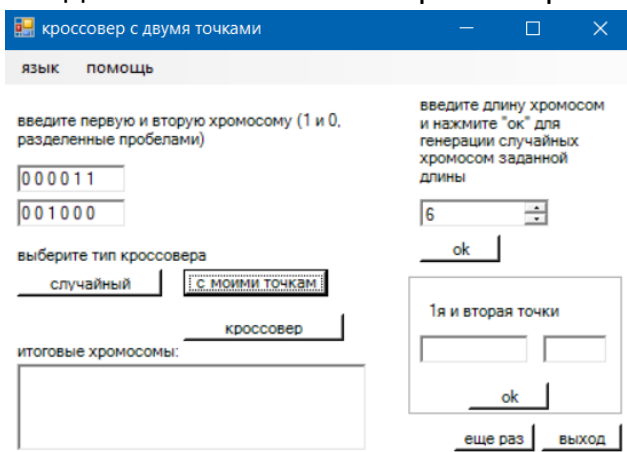
Ошибка при недопустимых символах в хромосомах:



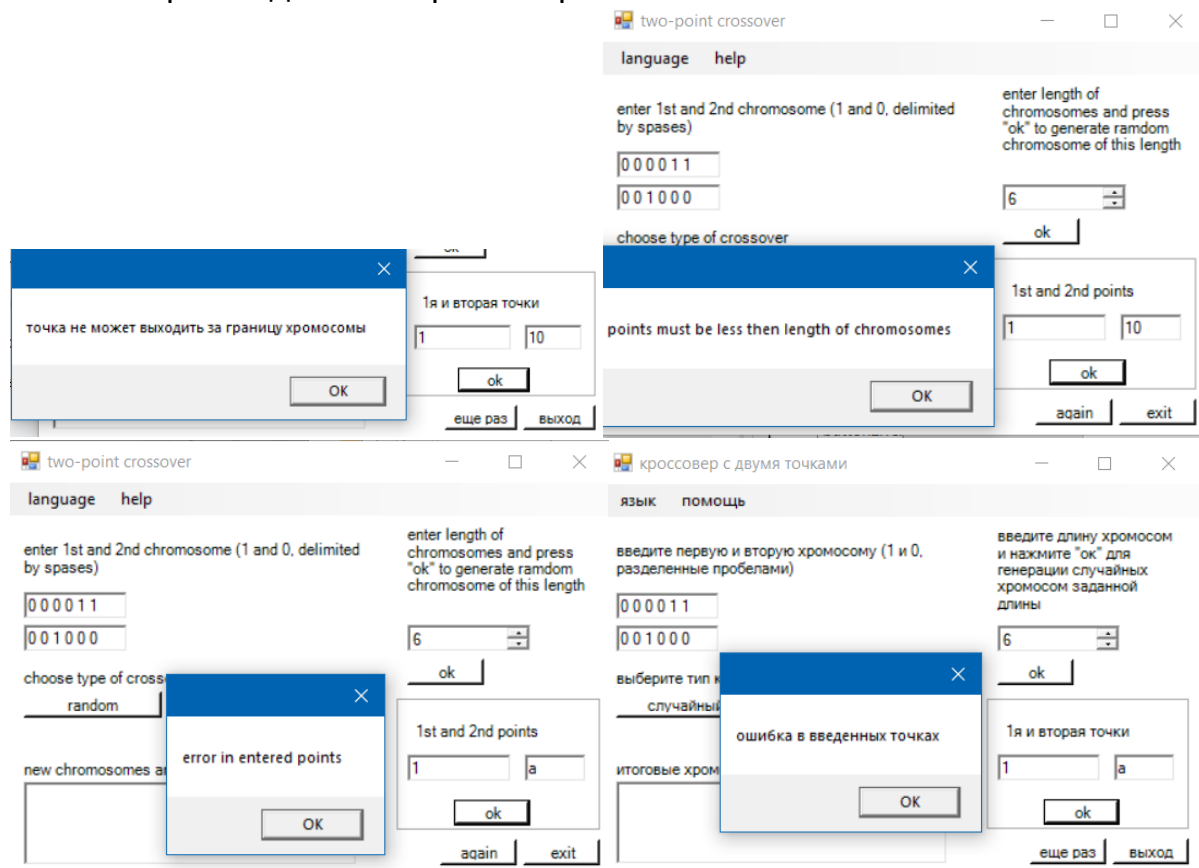
Ошибка: хромосомы разной длины:



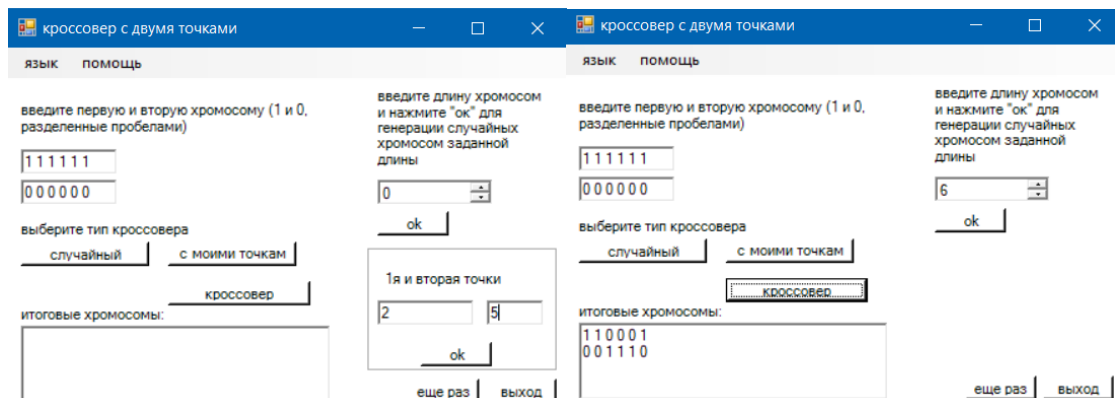
Ввод собственных точек кроссовера:



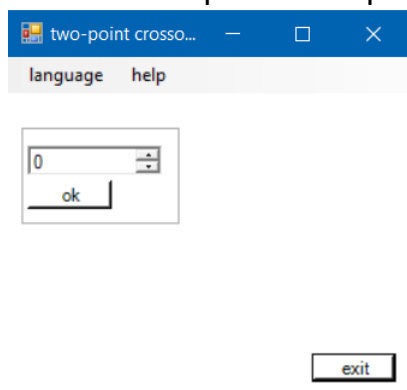
Ошибка при вводе точек кроссовера:



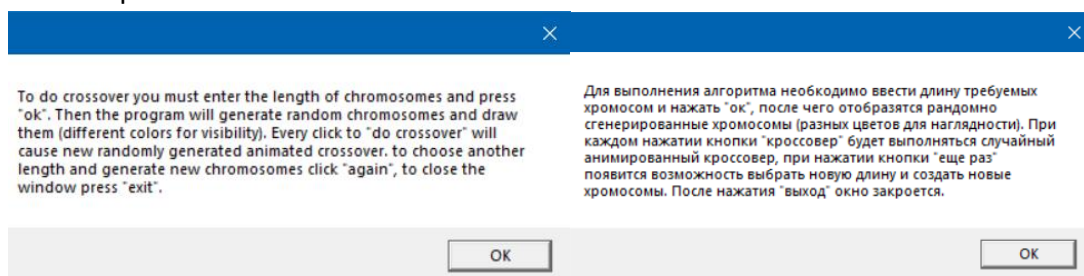
Результат работы кроссовера:



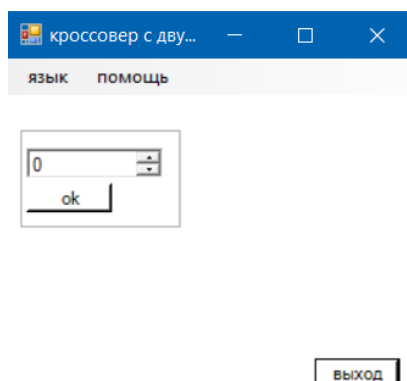
Окно анимированной работы с алгоритмом:



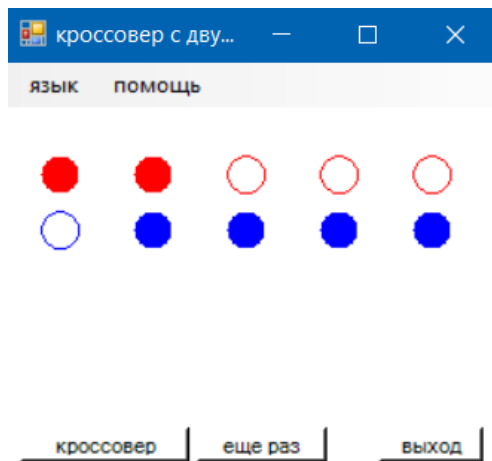
Помощь:



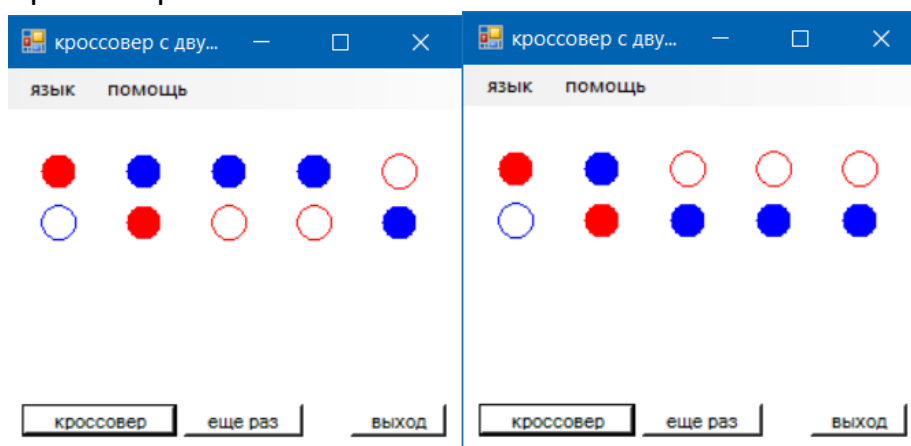
Смена языка:



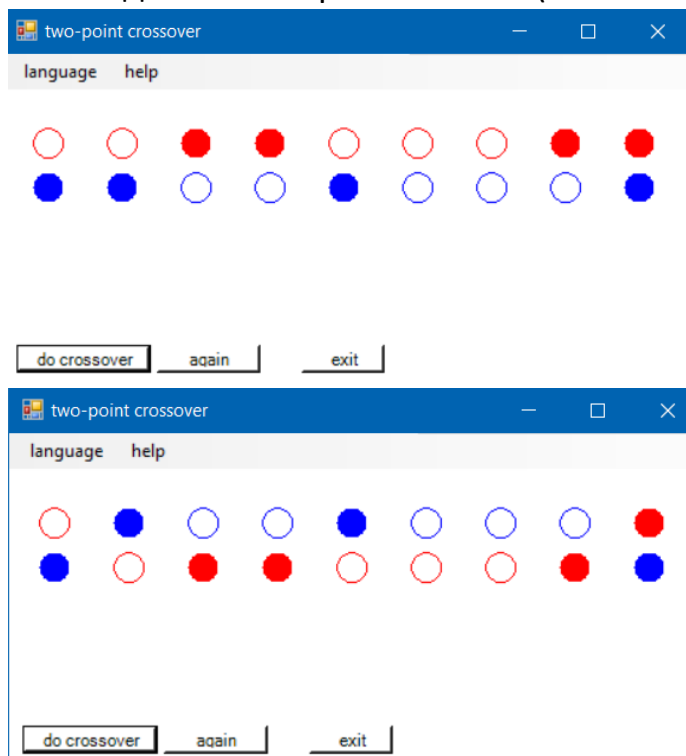
Окно после ввода длины хромосом:



Кроссовер:



Окно с длинными хромосомами (5+ элементов) и кроссовер:



Вывод

В ходе данной работы я реализовала и визуализировала алгоритм двухточечного кроссовера на C#. Выполняя курсовую работу я научилась применять на практике то, что изучила на протяжении курсов объектно-ориентированного программирования и инструментальных средств разработки прикладных программных систем, например, паттерны MVC и Abstract Factory, рисование и анимацию и другие возможности языка C# и Windows Forms.