

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Глазовский государственный педагогический институт имени В.Г. Короленко»

ОТЧЕТ

по учебной практике

ПМ 03 Участие в интеграции программных модулей

Специальность 09.02.03 Программирование в компьютерных системах

Выполнил(а):

студент(ка) ОП СПО группа 31

Руководитель практики: старший преподаватель
кафедры математики и информатики Касаткин К.А.

г. Глазов 2020

Оглавление

Введение.....	2
Актуальность	2
Цель и задачи.....	2
Аналоги японских головоломок	3
Судоку	3
Какуро	4
Японский кроссворд	5
Постановка задачи и требования, предъявляемые к программе.....	7
Описание алгоритма работы программы.....	8
Блок схема.....	12
Сред разработки и исходный код	13
Код программы.....	13
Тестирование программы.....	17
Руководство по установке	18
Руководство пользователя	19

Введение

Актуальность

Головоломка — непростая задача, для решения которой, как правило, требуется сообразительность, а не специальные знания высокого уровня.

Решение головоломок развивает память, воображение, находчивость, сообразительность, наблюдательность, логическое мышление и помогает усваивать предметы математического цикла.

Цель и задачи

Целью данной практики является создание проекта (игры) “Хитори”.

Для достижения данной цели, были поставлены следующие задачи:

- изучение средств разработки;
- проектирования проекта;
- реализация проекта в выбранной среде разработки.

Аналоги японских головоломок

Среди похожих игр головоломок есть 3 которые больше всех пользуются популярностью:

- судоку;
- какуро;
- японский кроссворд.

Перейдем к их более подробному рассмотрению.

Судоку

Самая популярная японская головоломка - это, конечно, судоку. В XVII веке Леонард Эйлер изобрёл игру «Carré latin» («Латинский квадрат»). На основе этой игры в 70-х годах прошлого века в Северной Америке были придуманы специальные числовые головоломки. Так, в США судоку появилась впервые в 1979 году в журнале «Dell Puzzle Magazine». Тогда она называлась «Number Place». Настоящую популярность судоку завоевала в 1980-1990-х годах, когда японский журнал «Nikoli» начал регулярно публиковать на своих страницах эту головоломку (с 1986 года). Стандартное Судоку представляет собой квадрат 9х9 клеток, в котором жирными линиями выделено 9 квадратов 3х3 клетки. Вам необходимо заполнить все клетки цифрами от 1 до 9 соблюдая простые правила:

- цифры не должны повторяться по горизонтали и вертикали;
- цифры не должны повторяться в квадратах 3х3.

		3		9	2			
4				3			1	
2	7							
	1		3					8
	5		1	6	7		3	
3					8		6	
							5	3
	3			8				9
			6	2		1		

Рис 1. Игровое поле судоку

Какуро

Головоломки КАКУРО впервые появились в Америке в 60-е годы. Называемые числовыми кроссвордами, они печатались в одних сборниках с обычными, чтобы читатели могли отвлечься от букв и переключиться на числа. Ситуацию изменил японский бизнесмен МакКи Кадзи, которому случайно попались на глаза числовые задачи. Именно он сумел рассмотреть их огромный потенциал. Кадзи видоизменил и переименовал головоломки. В его интерпретации они превратились в *kasan kurosu* - перекрестное сложение. На Западе головоломка также известна под названием *Cross Sums* - пересекающиеся суммы.

Поле головоломки состоит из клеток чёрного и белого цвета. Несколько белых клеток, идущих подряд по горизонтали или по вертикали, называются блоком. Цель игры - заполнить все белые клетки цифрами от 1 до

9 так, чтобы сумма цифр в блоке была равна числу, указанному в черной клетке слева от горизонтального блока и сверху от вертикального. При этом цифры в пределах одного блока не должны повторяться.

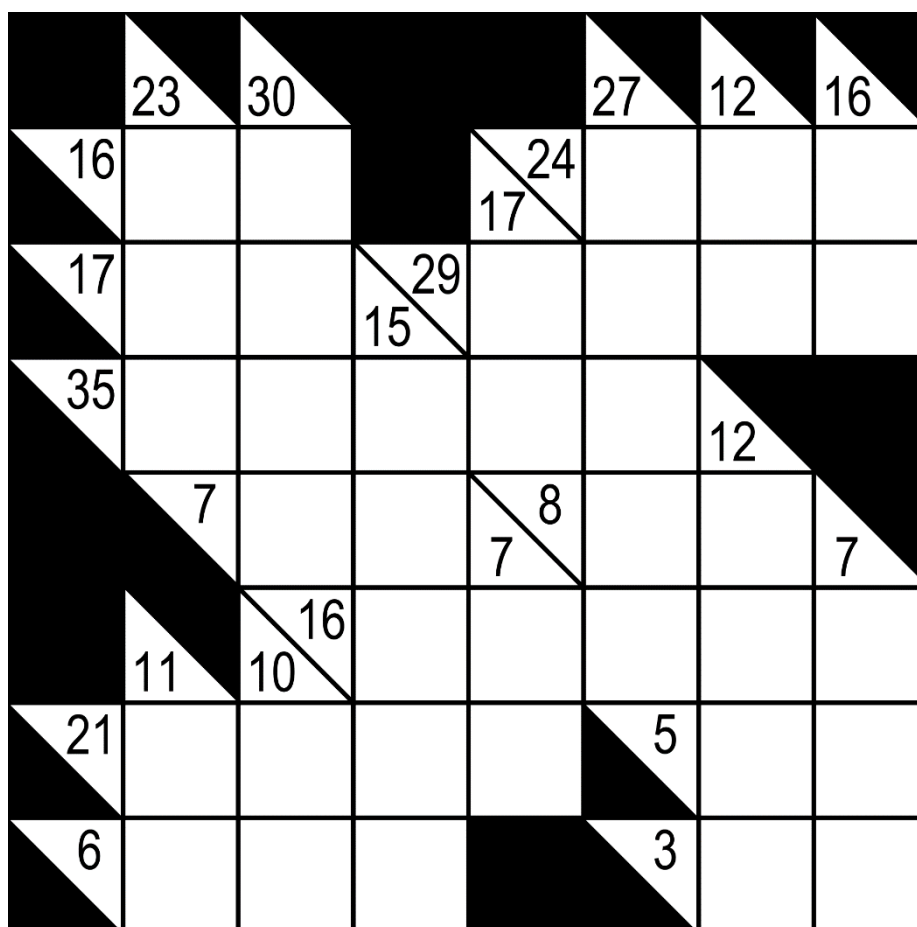


Рис 2. Игровое поле какуро

Японский кроссворд

Это головоломка, в которой зашифрованы не слова, а изображения. Японские кроссворды появились в Японии в конце XX века, на их авторство претендуют два человека. Одна из них - иллюстратор Нон Исида, утверждавшая, что ещё в 1970 году она создавала нонограммы как средство общения между людьми и животными. Нон Исида верила, что животные очень разумные существа, но из-за отсутствия средств коммуникации между людьми и животными человек недооценивает земных существ. В результате её научной работы, по утверждению Исиды, родились нонограммы (Nonogram) - поля с чёрными и белыми квадратами. В то же время японский автор

головоломок Тэцуя Нисио придумывает головоломки «Рисование по числам» (Paint by Numbers). Первоначально монограммы не вызвали особого интереса у любителей головоломок, так как никто не понимал, что из себя представляет эта головоломка, как её разгадывать. Широкую известность японские головоломки получили в 1989-1990 годах после публикации в британской газете The Telegraph.

Поле японского кроссворда расчерчено горизонтальными и вертикальными линиями разной толщины. Самые толстые линии отделяют центральную часть (поле для картинки) от цифр. Более тонкими линиями, поле делится на группы по 5 клеток - это сделано исключительно для удобства. Цифры указывают сколько клеток нужно закрасить черным цветом, если кроссворд ч/б (бывают еще цветные) по горизонтали и вертикали в указанном порядке: по горизонтали сверху вниз, по вертикали - слева направо. Если в столбце/строке несколько цифр, то между ними должна быть, как минимум, одна пустая клетка. Наша задача - определить, какие клетки нужно раскрасить.

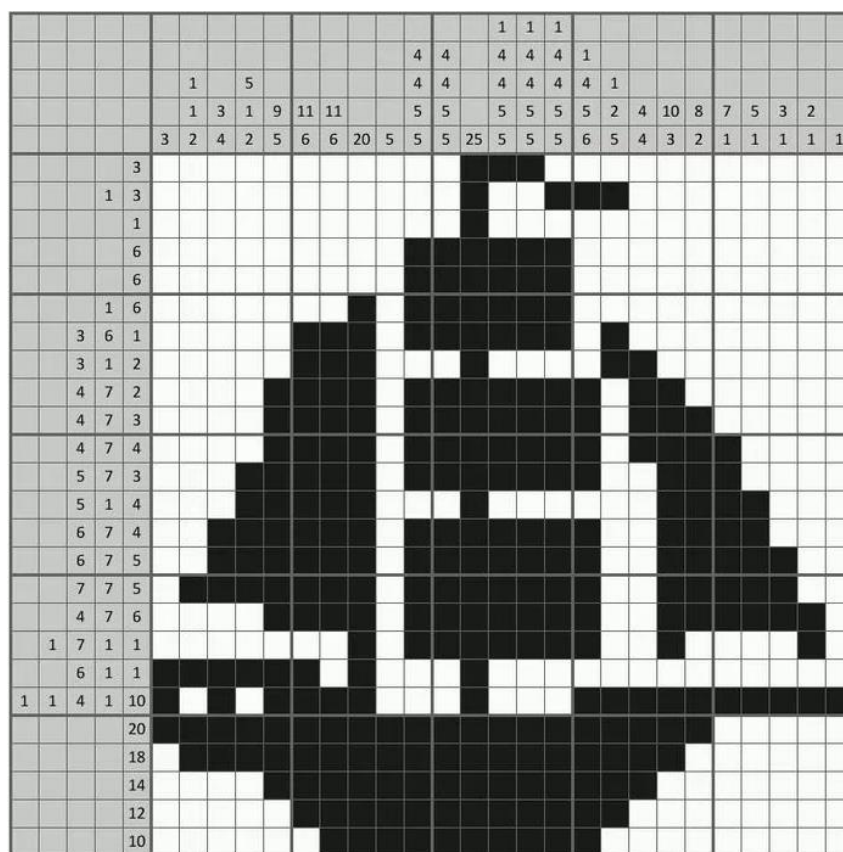


Рис 3. Игровое поле японского кроссворда

Постановка задачи и требования, предъявляемые к программе

Требования:

- кнопка загрузки задачи;
- диалоговое окно для выбора задачи;
- область вывода задачи;
- кнопка проверки;
- количество попыток и количество ходов.

Описание алгоритма работы программы

После запуска программы в открывшемся окне мы видим следующее:

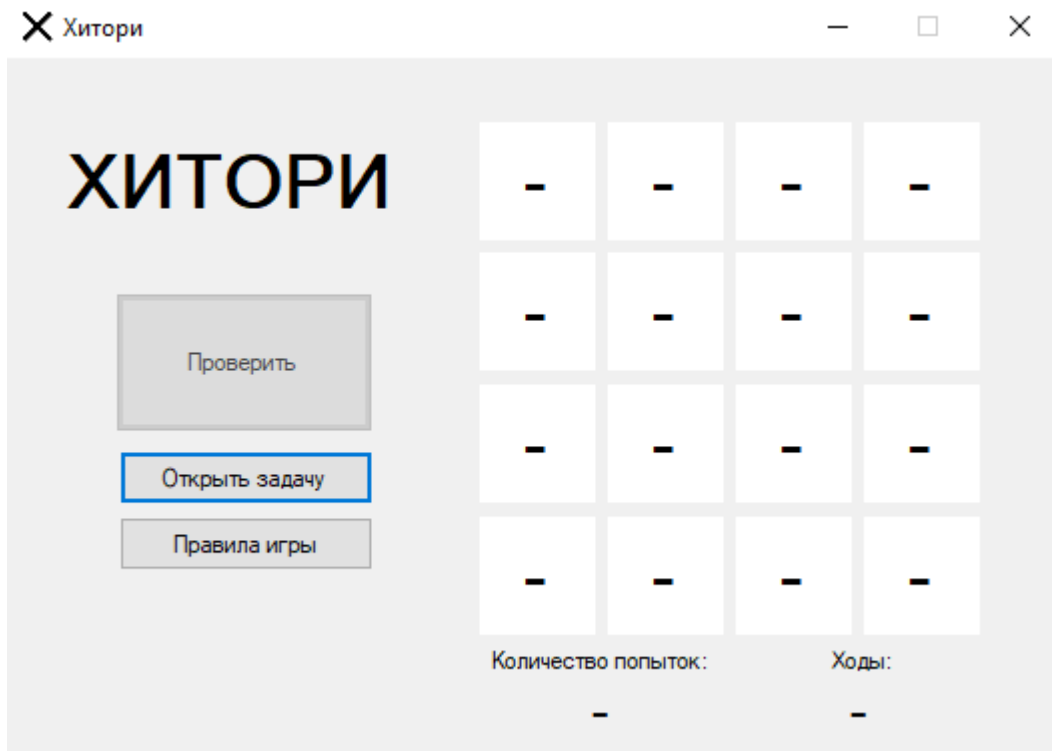


Рис 4. окно программы

В открывшемся окне имеется две активные кнопки “Открыть задачу” и “Правила игры”, при нажатии на кнопку “Правила игры” открывается окно в котором написаны правила.

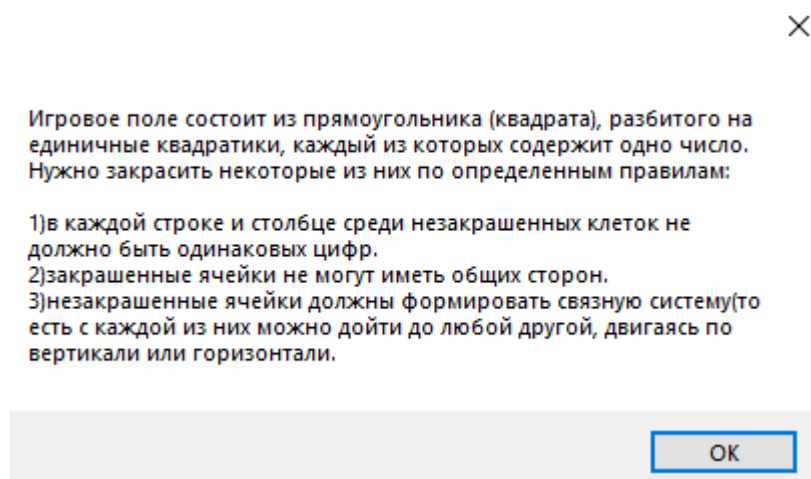


Рис 5. окно с правилами игры

Если нажать на кнопку “Открыть задачу”, откроется диалоговое окно где можно будет выбрать нужную нам задачу (формата .txt).

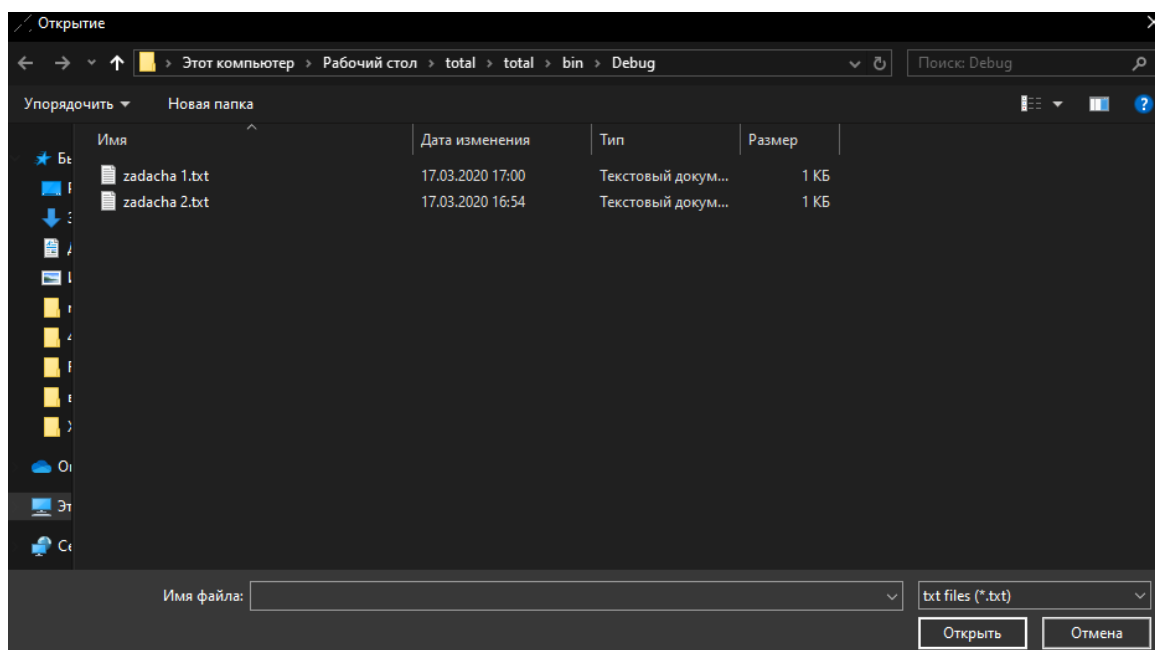


Рис 6. диалоговое окно выбора задачи

Выбираем задачу и нажимаем открыть, после видим, как игровое поле заполнилось данным, кнопка “Проверить” стала активна и “Количество попыток” стало равно трем.

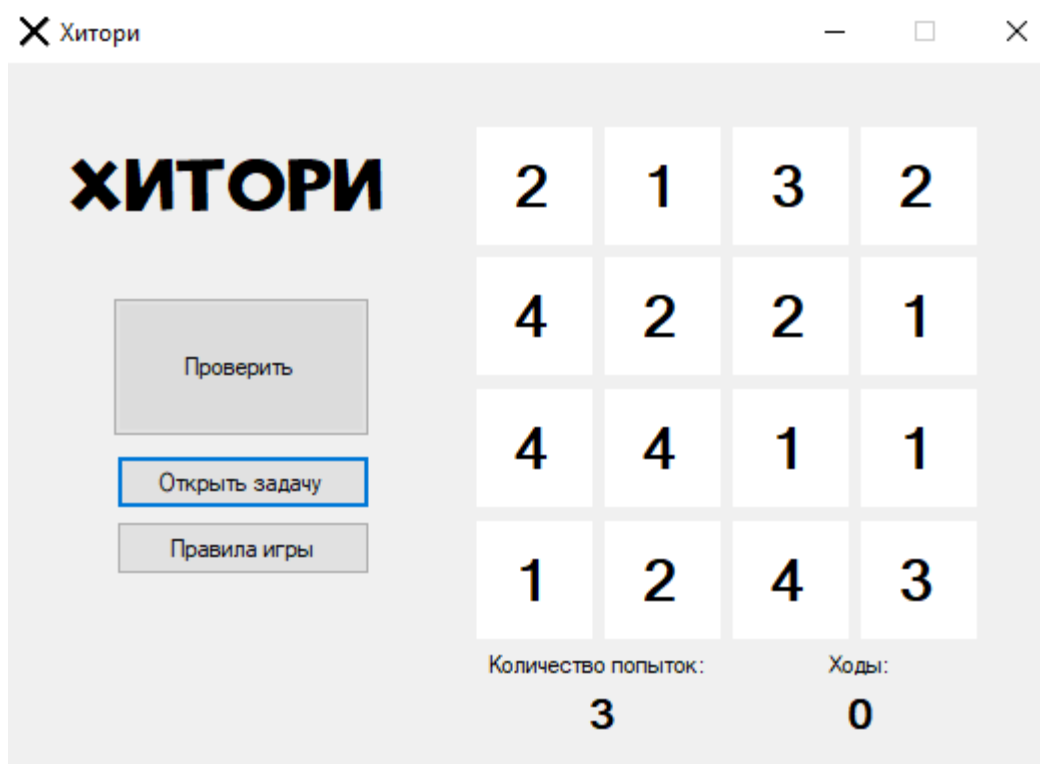


Рис 7. окно программы с загруженной задачей

Далее простым щелчком по ячейки окрашиваем ее в черный цвет, повторный щелчок возвращает ячейке белый цвет.

2	1	3	2
4	2	2	1
4	4	1	1
1	2	4	3

Рис 8. игровое поле с закрашенными ячейками

Если уверены в закрашенных ячейках, то жмем кнопку “Проверить”, в случае неверных закрашенных ячейка выйдет сообщение о потраченной попытке.

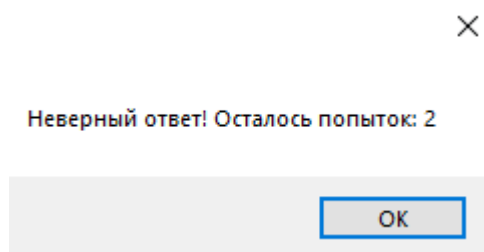


Рис 9. неверный ответ

В случае если ответ окажется верным выйдет сообщение о победе.

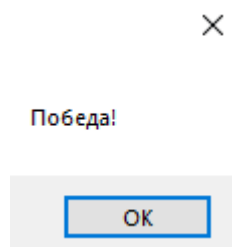


Рис 10. верный ответ

При трех потраченных ошибках кнопка “Проверить” заблокируется и вылезет сообщение о том, что вы проиграли так же кнопка, заблокируется, когда ответ будет верным.

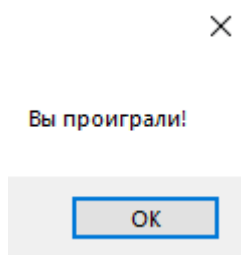
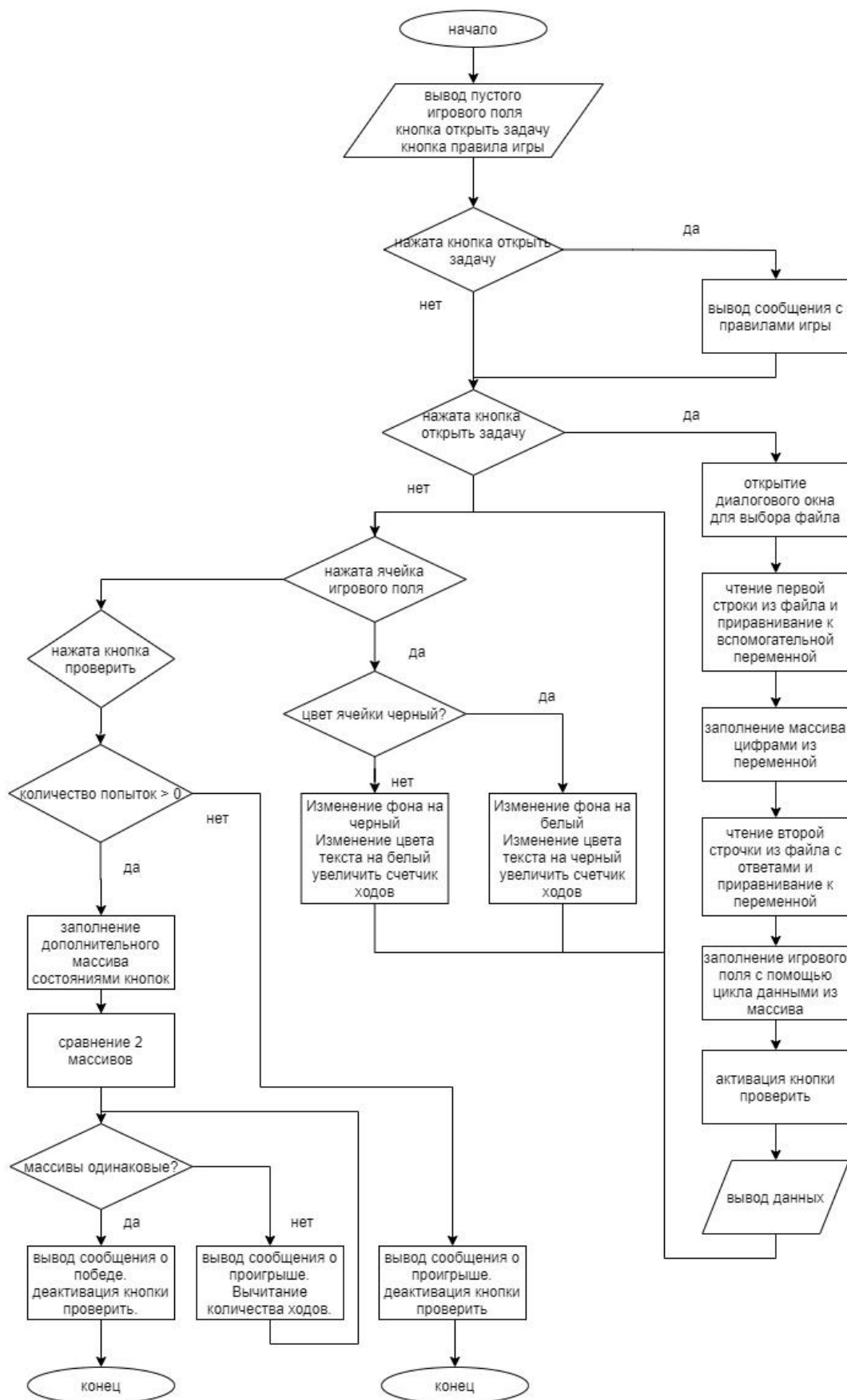


Рис 11. сообщение о проигрыше

Блок схема



Сред разработки и исходный код

В качестве среды разработки я выбрал C# т.к. это наиболее распространённый язык программирования для приложений.

Код программы

```
using System;
using System.IO;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace total
{
    public partial class Form1 : Form
    {
        char[] array = {    }; //цифры на поле
        int[] arrayS = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }; //состояния цифр
        int clk = 0;
        //количество ходов
        int trying = 3;
        //количество попыток
        string check = "";
        string answers = "";

        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            button1.Enabled = false;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            for (int i = 0; i < 16; i++)
            {
                check = check + arrayS[i];
            }
            if (check == answers)
            {
                button1.Enabled = false;
                MessageBox.Show("Победа!");
            }
        }
    }
}
```

```

    }
    else {
        check = "";
        trying--;
        label20.Text = Convert.ToString(trying);
        MessageBox.Show("Неверный ответ! Осталось
попыток: " + Convert.ToString(trying));
    }
    if (trying == 0) {
        button1.Enabled = false;
        MessageBox.Show("Вы проиграли!");
    }
}

private void label1_Click(object sender, EventArgs e)
{
    clk++;
    labelClk.Text = Convert.ToString(clk);
    if(label1.BackColor == Color.White) {
        label1.BackColor = Color.Black;
        label1.ForeColor = Color.White;
        arrayS[0] = 1;
    }
    else {
        label1.BackColor = Color.White;
        label1.ForeColor = Color.Black;
        arrayS[0] = 0;
    }
}

private void label2_Click(object sender, EventArgs e)
{
    clk++;
    labelClk.Text = Convert.ToString(clk);
    if (label2.BackColor == Color.White)
    {
        label2.BackColor = Color.Black;
        label2.ForeColor = Color.White;
        arrayS[1] = 1;
    }
    else
    {
        label2.BackColor = Color.White;
        label2.ForeColor = Color.Black;
        arrayS[1] = 0;
    }
}

...
private void label16_Click(object sender, EventArgs e)
{

```

```

        clk++;
        labelClk.Text = Convert.ToString(clk);
        if (label16.BackColor == Color.White)
        {
            label16.BackColor = Color.Black;
            label16.ForeColor = Color.White;
            arrayS[15] = 1;
        }
        else
        {
            label16.BackColor = Color.White;
            label16.ForeColor = Color.Black;
            arrayS[15] = 0;
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        clk = 0;
        //количество ходов
        trying = 3;
        //количество попыток
        check = "";
        answers = "";
        labelClk.Text = Convert.ToString(clk);
        var fileContent = string.Empty;
        var filePath = string.Empty;
        using (OpenFileDialog openFileDialog = new
OpenFileDialog())
        {
            openFileDialog.Filter = "txt files
(*.txt)|*.txt";
            openFileDialog.FilterIndex = 2;
            openFileDialog.RestoreDirectory = true;

            if (openFileDialog.ShowDialog() ==
DialogResult.OK)
            {
                filePath = openFileDialog.FileName;

                var fileStream = openFileDialog.OpenFile();
            }
        }

        StreamReader fs = new StreamReader(filePath);
        string s = "";
        s = fs.ReadLine();
        array = s.ToCharArray(0, 16);
        s = fs.ReadLine();
        answers = s;
        fs.Close();
        List<Control> list_labels = new List<Control>();
    }

```



```

        list_labels.Add(label1);
        list_labels.Add(label2);
        list_labels.Add(label3);
        list_labels.Add(label4);
        list_labels.Add(label5);
        list_labels.Add(label6);
        list_labels.Add(label7);
        list_labels.Add(label8);
        list_labels.Add(label9);
        list_labels.Add(label10);
        list_labels.Add(label11);
        list_labels.Add(label12);
        list_labels.Add(label13);
        list_labels.Add(label14);
        list_labels.Add(label15);
        list_labels.Add(label16);
        label20.Text = Convert.ToString(trying);
        for (int i = 0; i < 16; i++)
        {
            list_labels[i].Text =
Convert.ToString(array[i]);
        }
        button1.Enabled = true;
    }

    private void button3_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Игровое поле состоит из
прямоугольника (квадрата), разбитого на единичные квадратики,
каждый из которых содержит одно число. \nНужно закрасить
некоторые из них по определенным правилам: \n \n1) в каждой
строке и столбце среди незакрашенных клеток не должно быть
одинаковых цифр. \n2) закрашенные ячейки не могут иметь общих
сторон. \n3) незакрашенные ячейки должны формировать связную
систему (то есть с каждой из них можно дойти до любой другой,
двигаясь по вертикали или горизонтали.");
    }
}

```

Тестирование программы

В ходе тестирования программы был выявлен ряд недочетов:

- При загрузке новой задачи во время действующей игры не обнулялось количество ходов и попыток.

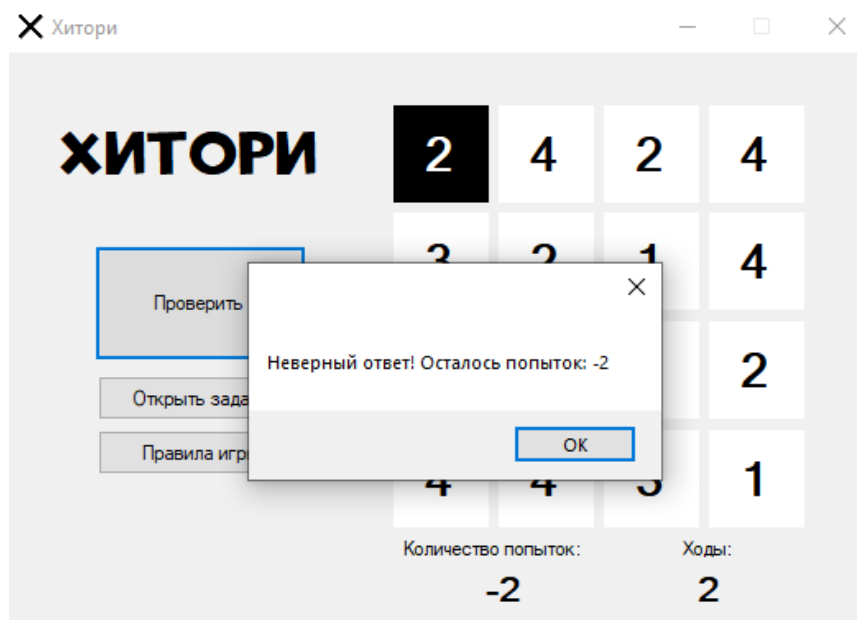


Рис 12. неправильный подсчет количества попыток

Если открыть диалог для выбора файла задачи, но отменить выбор, то программа завершается с ошибкой.

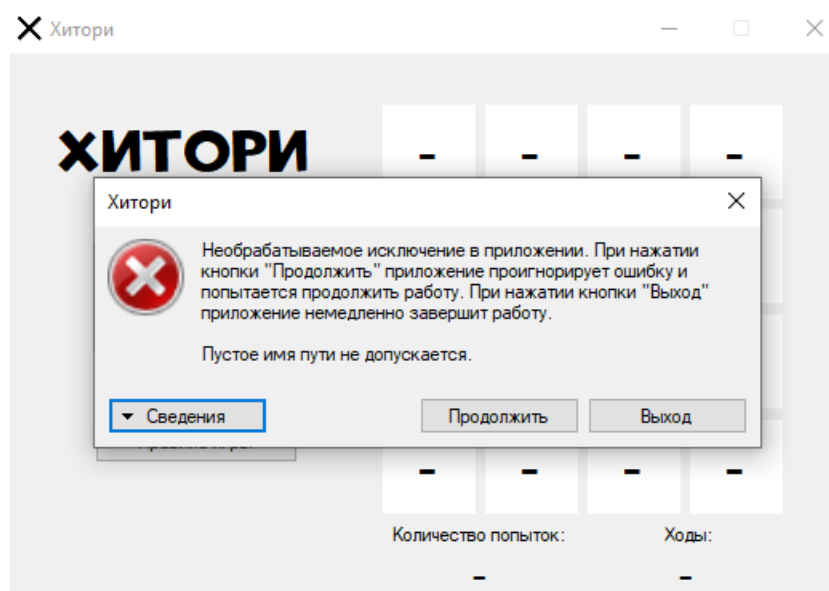


Рис 13. Аварийное завершение программы при отмененном выборе файла

Руководство по установке

Перед тем как начать пользоваться программой надо её скачать. Перейдя по ссылке [“https://github.com/BigSmokeRememberThatName/parctic2020”](https://github.com/BigSmokeRememberThatName/parctic2020) качаем rar-архив “Hitori”. После скачивания архива его следует разархивировать в любое удобное для вас место. После извлечения папки “total” из архива запустите файл с названием “total.exe”, задачи для игры находятся в папке “задачи” в количестве 10 штук.

Минимальные системные требования:

ОС: Windows Vista.

ОЗУ: 1ГБ.

Свободное место на диске: 1МБ.

Разрешение экрана: 800x600.

Дополнительно: мышка.

Руководство пользователя

После запуска программы для открытия изображения нужно нажать кнопку “Загрузить задачу” в открывшемся диалоговом окне выбрать нужную нам задачу и нажать открыть.

Для ознакомления с правилами игры необходимо нажать на кнопку «Правила игры». После этого открывается сообщение с правилами игры.

Для проверки правильности решения необходимо нажать на кнопку проверить. После этого игра сообщит о правильности\неправильности решения:

При правильном решении игра показывает сообщение с правильностью решения. Кнопка «Проверить» блокируется, и игра завершается

При неправильном решении игра вычитает количество попыток на единицу. Если количество попыток равняется нулю, то игра сообщает о проигрыше и кнопка «Проверить» блокируется.

Для создания своей задачи необходимо создать .txt файл и заполнить по примеру:

2132422144111243

0001010010010000

Первая строка – цифры на поле, вторая - правильные ячейки.

Ссылки на интернет-источники

1. Программирование на языке C# 5.0: Тотальный тренинг по Си-шарпу. Начальный уровень [Электронный ресурс]

URL: <http://rvgames.de/ru/cs13.htm>(дата обращения 10.03.19)

2. DataGridView Класс [Электронный ресурс]

URL: <https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.forms.datagridview?view=netframework-4.8>(дата обращения 12.03.19)

3. PictureBox Класс [Электронный ресурс]

URL: <https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.forms.picturebox?view=netframework-4.8> (дата обращения 14.03.19)

4. FileStream. Чтение и запись файла [Электронный ресурс]

URL: <https://metanit.com/sharp/tutorial/5.4.php>(дата обращения 17.03.19)

5. Array Класс [Электронный ресурс]

URL: <https://docs.microsoft.com/ru-ru/dotnet/api/system.array?view=netframework-4.8> (дата обращения 14.03.19)