

Северо-Осетинский Государственный Университет
Факультет математики и информационных технологий

Курсовая Работа
Обучающая игра “Черный Ящик”

Курса
Дудиев

Студент Первого
В.К.

Техническое задание

Постановка Задачи

Выбор и обоснование критериев разрабатываемой программы
обоснования разработки программы

Ограничения системы

Выбор языков программирования

Разработка эскизного и технического проектов программы

Введение

Назначение и область применения.

Технические характеристики

Постановка задачи

Описание алгоритма

Организация входных данных

Разработка рабочего проекта

Описание программного продукта

Описание Пользовательского интерфейса

Руководство пользователя

Спецификация программы

Текст программы

Программный текст главной формы:

Программный текст Окна для ответов:

Общие Сведения

Функциональные ограничения

Описание логической структуры

Используемые технические средства

Входные данные

Выходные данные

Внедрение

Руководство системного программиста

Дополнительные возможности Приложения

Действия по настройке программы

Сведения о структуре

Сообщения выдаваемые Пользователю во время выполнения задачи

Руководство программиста

Поддерживаемые операционные системы

Минимальные системные требования

Руководство оператора

Последовательность действий обеспечивающих загрузку, запуск, выполнение и завершение программы

Описание языка

Приложения и перечни

Заключение

Список литературы

Техническое задание

Постановка Задачи

Требуется разработать Приложение “Черный Ящик ” с возможностью ввода одного или более чисел. Потом приложение обработает эти числа в соответствии с выбранным уровнем, которых должно быть несколько и они будут увеличивать сложность с повышением уровня, и выведет результат напротив соответствующих чисел. Потом, если игрок догадался, по какому алгоритму обрабатываются числа, он нажимает на кнопку ответа. Там ему даётся несколько случайных чисел и он должен для каждого соответственно ввести число по предлагаемому алгоритму и система сравнит введенный и желаемый результат и выводит правильный ли ответ или нет и откроет следующий уровень, если ответ верный.

Выбор и обоснование критериев разрабатываемой программы

Было решено минимизировать различные надписи указатели и т.д. во внешнем виде программы , так как основными пользователями ожидаемо будут школьники начальных и средних классов. У них очень рассеянное внимание и нельзя допустить , чтобы их отвлекали лишние детали и они бы смогли полностью сконцентрироваться на вычислении алгоритма. в окне для ввода ответа было решено сделать обязательным вычисление и ответ для пяти чисел, чтобы исключить случайные правильные ответы(легче будет разгадать алгоритм чем пытаться подобрать правильный ответ).

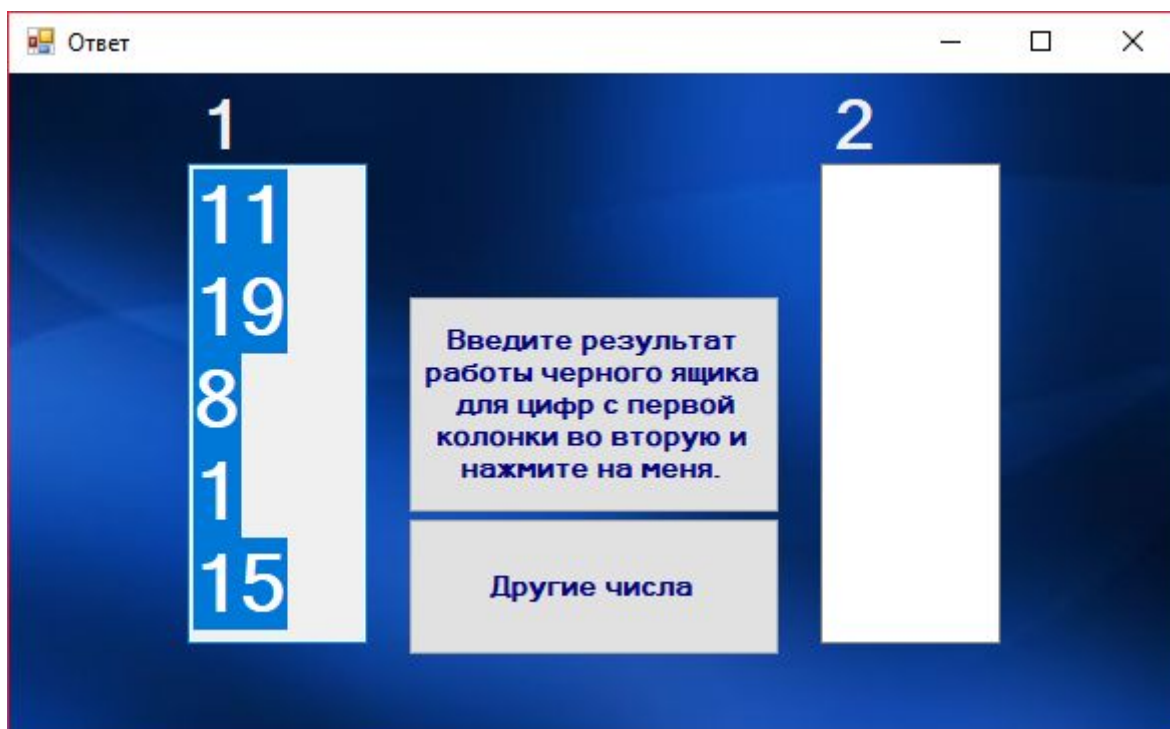


Рис 1. Обязательный ответ для 5 чисел

обоснования разработки программы

Обоснованием для разработки этой программы стало банальное упрощение работы учителей и приучение пользователей (в основном детей) к взаимодействию с компьютером и понимание алгоритмов происходящих внутри ЭВМ.

Ограничения системы

на данном этапе нельзя являясь обычным пользователем менять количество уровней с помощью интерфейса, количество данных чисел на проверку ответа, и самый большой минус программы в том, что нельзя менять вид алгоритма в уровнях на усмотрение учителя.

Выбор языков программирования

так как данная программа не является сложной и не требует сложных программных манипуляций, на усмотрение программиста был выбран язык программирования C#.

Разработка эскизного и технического проектов программы

Введение

Игра “Черный Ящик” с первого взгляда может показаться детской, простой и понятной, если это и так , то только на первых уровнях, дальше идут алгоритмы с более высоким уровнем сложности , с которыми справится не каждый обычный человек. Эта, на первый взгляд, легкая игра помогает детям с ранних возрастов стараются приучить себя думать, анализировать данные, размышлять алгоритмически. Если вы Думаете , что это только для детей , то вы ошибаетесь. взрослым людям не связанным с техническими Специальностями и особенно для деятелей этих специальностей будет полезно такая небольшая “разминка для ума”, хотя могут быть “черные ящики” и для взрослых с алгоритмами, гораздо сложного уровня.

“BlackBox” это приложение которое реализовывает игру “Черный Ящик” в котором упор идет на математические алгоритмы то есть работы ведется только с цифрами. Оно включает в себя 10 уровней различной сложности (начиная с элементарных и заканчивая сложными

алгоритмами). Сейчас приложение выполняет только основные функции. но в будущем планируется улучшать и совершенствовать приложение опираясь также на отзывы пользователей

Назначение и область применения.

Программа разрабатывается для учебных заведений разного типа и может применяться на разных уроках или внеклассных занятиях связанных с изучением информационных технологий и математикой, но также может применяться людьми желающих начать изучать ИТ в любой возрастной категории. Для людей же , которые так или иначе знакомы с основными принципами программирования, данная программа с данными, конкретными алгоритмами покажется и является слишком элементарным и не даст много нового.

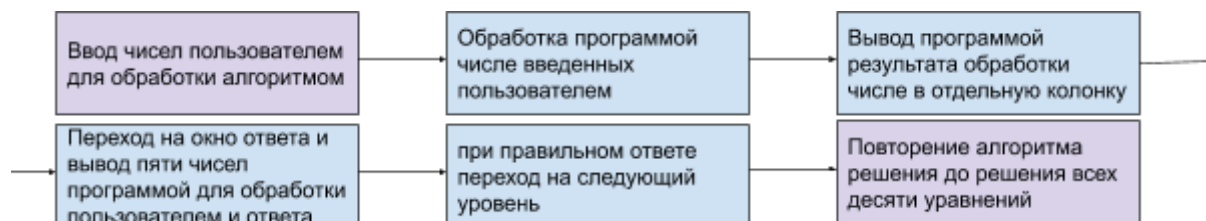
Технические характеристики

Постановка задачи

Задача состоит в реализации не сильно нагружающей, легкой в освоении и понимании игры “Черный Ящик” в котором по сути могут быть абсолютно разные объекты для обработки, начиная со слов и заканчивая предметами, но для данной задачи необходимо реализовать эту игру с математическим уклоном, где обрабатываемые объекты это числа

Описание алгоритма

Алгоритм этой игры достаточно прост и направлен на людей с не отличным владением компьютера



Организация входных данных

пользователь для прохождения, к примеру, одного уровня игры должен вводить данные как минимум два раза: в первом случае для того, чтобы определить алгоритм данного уровня. В этом случае на вход дается просто строка, которая далее разделяется на числа с помощью метода “Split” которая делит строку на числа ориентируясь на переход на следующую строку и все данные конвертируются в тип “int” и добавляются в список “List”. То же самое для случая проверки ответа.

Разработка рабочего проекта

Современные программные приложения должны удовлетворять требованию дружественного интерфейса, поэтому разработка программного приложения делится на два этапа:

- Создание программных модулей, реализующих поставленную задачу;
- Создание дружественного интерфейса.

Описание программного продукта

Описание Пользовательского интерфейса

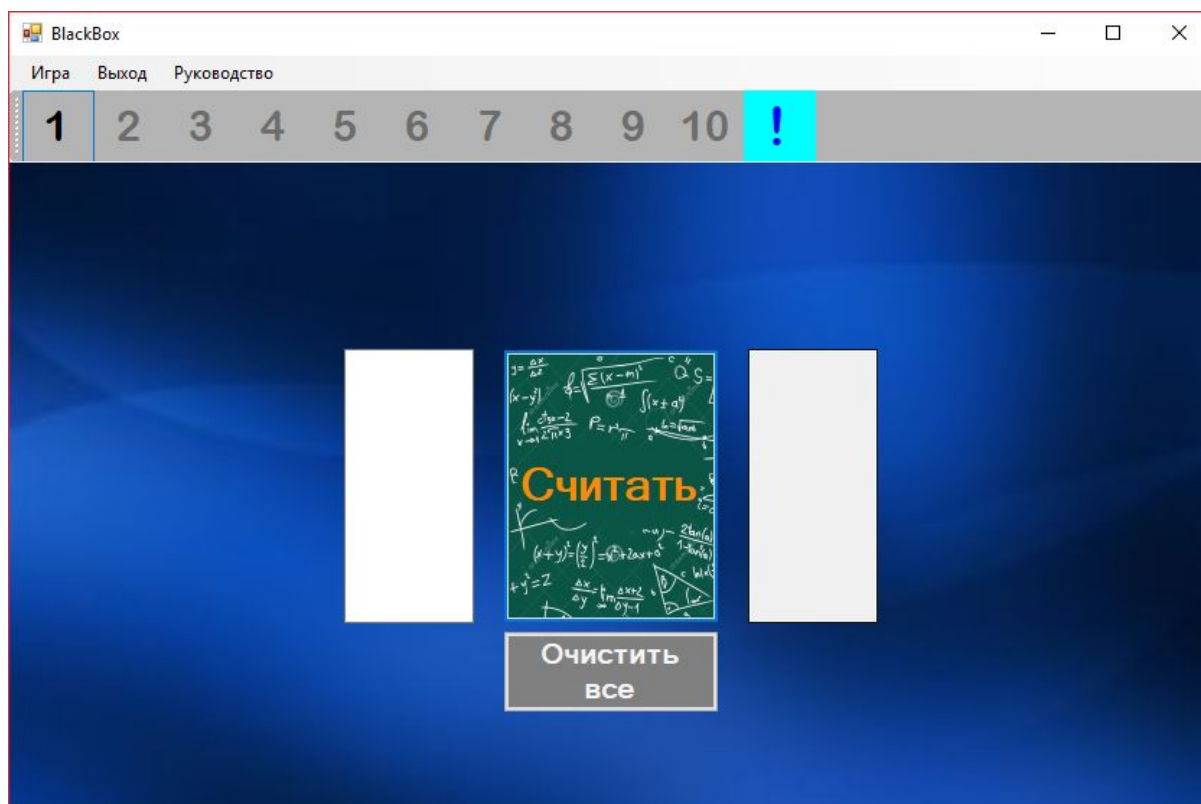


Рис.2. Главная Окно

Окно Состоит Из Кнопок Выбора Уровня, Поле для записи Чисел Пользователем, Кнопкой “Считать” для обработки чисел ранее введенных пользователем, Поля в котором выводятся числа обработанные алгоритмом, кнопка “Очистить всё” который удаляет все записи с обеих колонок

Кнопка “!”, которая переходит окно ввода ответа:

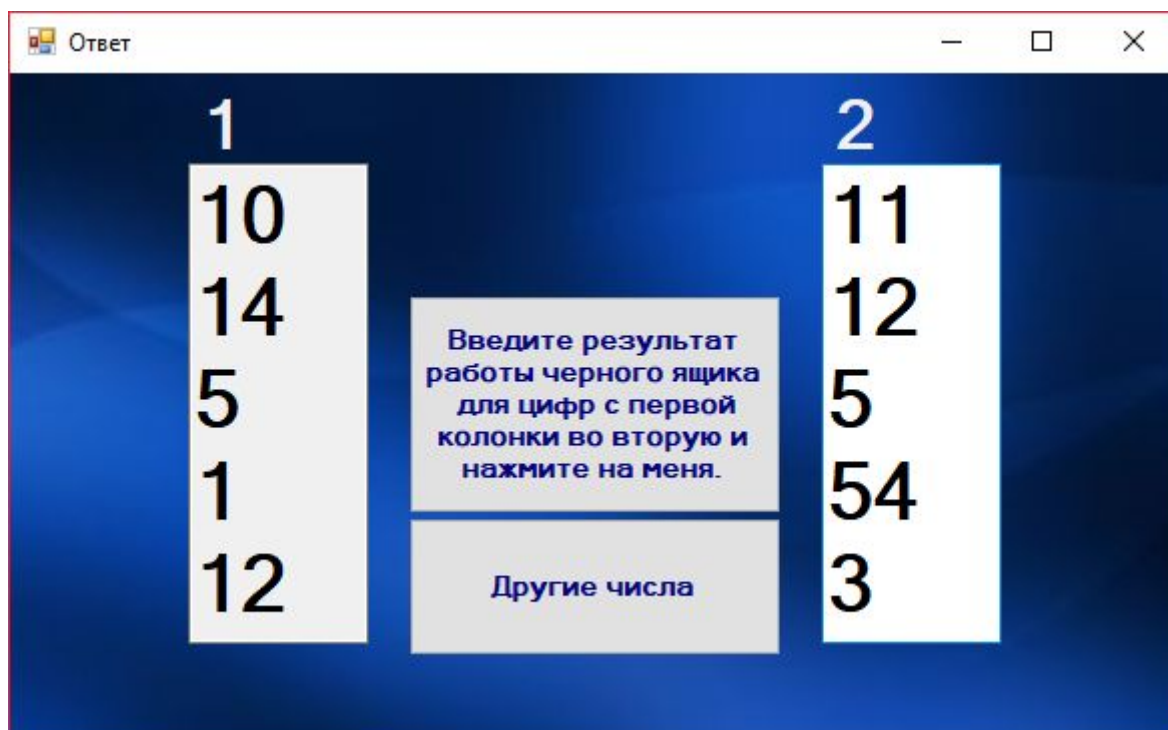


Рис.3. Окно для Ответа

В этом окне есть Поля в котором 5 разных,случайных чисел,второе поле для записи ответа пользователя, верхняя кнопка между колонками для расчета правильности ответа, а нижняя для генерации других чисел

Руководство пользователя

для начало запустите игру нажатием на “BlackBox.exe”. Перед вами откроется главное окно Приложения.Введите в первом поле пару чисел на отдельных строках и нажмите на кнопку “Считать” для того чтобы понять,какой алгоритм используется на данном уровне :

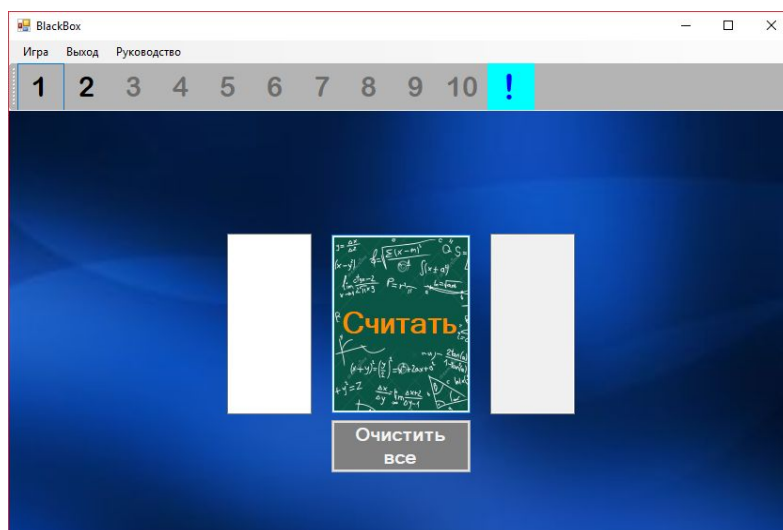


Рис.4.Пример Ввода чисел на главной форме

после того , как вы поняли обрабатывающий алгоритм нажимает на Кнопку “!” в левом-верхнем углу. Откроется Окно для ввода ответов. Вы должны обработать засветившиеся числа и для каждого соответственно написать ответ в противоположном поле, после чего нажать на кнопку между ними. вас оповестят о корректности ответа сообщением и кнопка окрасится в зеленый или красный цвет. После правильного ответа на главной форме разблокируется следующий уровень(Если сгенерированные числа вас не устраивают нажмите на кнопку “Другие числа” для их смены):

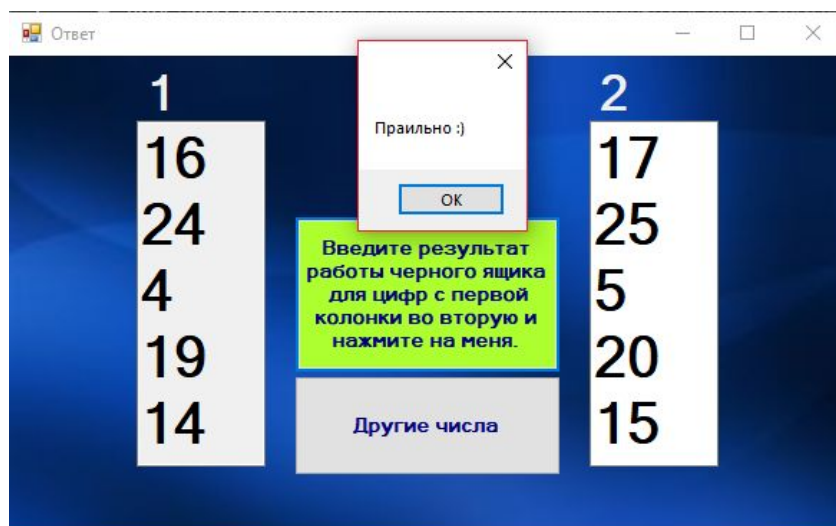


Рис.5.Пример правильного Ответа

и так до тех пор пока не будут отгаданы все 10 алгоритмов.

Спецификация программы

Программа разрабатывается для учебных заведений разного типа и может применяться на разных уроках или внеклассных занятиях связанных с изучением информационных технологий и математикой

Текст программы

Программный текст главной формы:

```
using System;
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 BlackBox
{
    //Внимание!!!! Проест очень очень сырой большинсто исключений не
    //обработано. Не старайтесь сломать программу, я вам говрю:у вас это получится!!!
    public partial class MainForm : Form
    {
        //лист входных данных
        List<string> listOfInput = new List<string>();
        static int SelectedLevel = 1;
        public bool answerTruthCheck = false;

        void locationChange()
        {
            buttonToAnswer.Left = panelMain.Width / 2 - buttonToAnswer.Width / 2;
            buttonToAnswer.Top = panelMain.Height / 2 - buttonToAnswer.Height / 2;
            buttonRemoveAll.Left = panelMain.Width / 2 - buttonRemoveAll.Width / 2;
            buttonRemoveAll.Top = panelMain.Height / 2 - buttonToAnswer.Height / 2 +
            5 + buttonToAnswer.Height;
        }
    }
}
```

```

        textBoxInput.Left = buttonToAnswer.Left - textBoxInput.Width - 20;
        textBoxOutput.Left = buttonToAnswer.Left + buttonToAnswer.Width + 20;
        textBoxInput.Top = buttonToAnswer.Top;
        textBoxOutput.Top = buttonToAnswer.Top;
    }

    public MainForm()
    {
        InitializeComponent();
        locationChange();
    }

    private double OutputLvl(string input)
    {
        try
        {
            //для каждого уровня своё возвращаемое число
            switch (SelectedLevel)
            {
                case 1:
                {
                    return Convert.ToDouble(input) + 1;
                }

                case 2:
                {
                    return Convert.ToDouble(input) + 5;
                }

                case 3:
                {
                    return Convert.ToDouble(input) + 15;
                }

                case 4:
                {
                    return Convert.ToDouble(input) * 2;
                }

                case 5:
                {

```

```

        return Convert.ToDouble(input) * 3;
    }

    case 6:
    {
        return Convert.ToDouble(input) * 2 + 1;
    }

    case 7:
    {
        return Convert.ToDouble(input) * 2 - 2;
    }

    case 8:
    {
        return Convert.ToDouble(input) / 2;
    }

    case 9:
    {
        return Convert.ToDouble(input) / 2 + 1;
    }
    case 10:
    {
        return Math.Pow(Convert.ToDouble(input), 2);
    }

    }
}
catch (FormatException)
{
    MessageBox.Show("Вводите только цифры и без пустых строк между
ними!");
}
return 0;
}

```

```

private void buttonToAnswer_Click(object sender, EventArgs e)
{
    listOfInput.Clear();
}

```

```

textBoxOutput.Clear();
listOfInput.AddRange(textBoxInput.Text.Split('\n'));

for (int i = 0; i < listOfInput.Count(); i++)
{
    if (listOfInput[i] != "")
    {
        textBoxOutput.Text += (OutputLvl(listOfInput[i]) + "\r\n");
    }
    else
    {
        break;
    }
}

}

internal void outputLvl1()
{
    throw new NotImplementedException();
}

//выбор уровня (по-другому не смог)
private void toolStripButtonLvl1_Click(object sender, EventArgs e)
{
    SelectedLevel = 1;
    toolStripButtonLvl1.Checked = true;
    toolStripButtonLvl2.Checked = false;
    toolStripButtonLvl3.Checked = false;
    toolStripButtonLvl4.Checked = false;
    toolStripButtonLvl5.Checked = false;
    toolStripButtonLvl6.Checked = false;
    toolStripButtonLvl7.Checked = false;
    toolStripButtonLvl8.Checked = false;
    toolStripButtonLvl9.Checked = false;
    toolStripButtonLvl10.Checked = false;
}

private void toolStripButtonLvl2_Click(object sender, EventArgs e)
{
    SelectedLevel = 2;

```

```

        toolStripButtonLv11.Checked = false;
        toolStripButtonLv12.Checked = true;
        toolStripButtonLv13.Checked = false;
        toolStripButtonLv14.Checked = false;
        toolStripButtonLv15.Checked = false;
        toolStripButtonLv16.Checked = false;
        toolStripButtonLv17.Checked = false;
        toolStripButtonLv18.Checked = false;
        toolStripButtonLv19.Checked = false;
        toolStripButtonLv110.Checked = false;
    }

    private void toolStripButtonLv13_Click(object sender, EventArgs e)
    {
        SelectedLevel = 3;
        toolStripButtonLv11.Checked = false;
        toolStripButtonLv12.Checked = false;
        toolStripButtonLv13.Checked = true;
        toolStripButtonLv14.Checked = false;
        toolStripButtonLv15.Checked = false;
        toolStripButtonLv16.Checked = false;
        toolStripButtonLv17.Checked = false;
        toolStripButtonLv18.Checked = false;
        toolStripButtonLv19.Checked = false;
        toolStripButtonLv110.Checked = false;
    }

    private void toolStripButtonLv14_Click(object sender, EventArgs e)
    {
        SelectedLevel = 4;
        toolStripButtonLv11.Checked = false;
        toolStripButtonLv12.Checked = false;
        toolStripButtonLv13.Checked = false;
        toolStripButtonLv14.Checked = true;
        toolStripButtonLv15.Checked = false;
        toolStripButtonLv16.Checked = false;
        toolStripButtonLv17.Checked = false;
        toolStripButtonLv18.Checked = false;
        toolStripButtonLv19.Checked = false;
        toolStripButtonLv110.Checked = false;
    }

```



```

private void toolStripButtonLvl5_Click(object sender, EventArgs e)
{
    SelectedLevel = 5;
    toolStripButtonLvl1.Checked = false;
    toolStripButtonLvl2.Checked = false;
    toolStripButtonLvl3.Checked = false;
    toolStripButtonLvl4.Checked = false;
    toolStripButtonLvl5.Checked = true;
    toolStripButtonLvl6.Checked = false;
    toolStripButtonLvl7.Checked = false;
    toolStripButtonLvl8.Checked = false;
    toolStripButtonLvl9.Checked = false;
    toolStripButtonLvl10.Checked = false;
}

```

```

private void toolStripButtonLvl6_Click(object sender, EventArgs e)
{
    SelectedLevel = 6;
    toolStripButtonLvl1.Checked = false;
    toolStripButtonLvl2.Checked = false;
    toolStripButtonLvl3.Checked = false;
    toolStripButtonLvl4.Checked = false;
    toolStripButtonLvl5.Checked = false;
    toolStripButtonLvl6.Checked = true;
    toolStripButtonLvl7.Checked = false;
    toolStripButtonLvl8.Checked = false;
    toolStripButtonLvl9.Checked = false;
    toolStripButtonLvl10.Checked = false;
}

```

```

private void toolStripButtonLvl7_Click(object sender, EventArgs e)
{
    SelectedLevel = 7;
    toolStripButtonLvl1.Checked = false;
    toolStripButtonLvl2.Checked = false;
    toolStripButtonLvl3.Checked = false;
    toolStripButtonLvl4.Checked = false;
    toolStripButtonLvl5.Checked = false;
    toolStripButtonLvl6.Checked = false;
    toolStripButtonLvl7.Checked = true;
    toolStripButtonLvl8.Checked = false;
}

```

```

        toolStripButtonLvl9.Checked = false;
        toolStripButtonLvl10.Checked = false;
    }

private void toolStripButtonLvl8_Click(object sender, EventArgs e)
{
    SelectedLevel = 8;
    toolStripButtonLvl1.Checked = false;
    toolStripButtonLvl2.Checked = false;
    toolStripButtonLvl3.Checked = false;
    toolStripButtonLvl4.Checked = false;
    toolStripButtonLvl5.Checked = false;
    toolStripButtonLvl6.Checked = false;
    toolStripButtonLvl7.Checked = false;
    toolStripButtonLvl8.Checked = true;
    toolStripButtonLvl9.Checked = false;
    toolStripButtonLvl10.Checked = false;
}

private void toolStripButtonLvl9_Click(object sender, EventArgs e)
{
    SelectedLevel = 9;
    toolStripButtonLvl1.Checked = false;
    toolStripButtonLvl2.Checked = false;
    toolStripButtonLvl3.Checked = false;
    toolStripButtonLvl4.Checked = false;
    toolStripButtonLvl5.Checked = false;
    toolStripButtonLvl6.Checked = false;
    toolStripButtonLvl7.Checked = false;
    toolStripButtonLvl8.Checked = false;
    toolStripButtonLvl9.Checked = true;
    toolStripButtonLvl10.Checked = false;
}

private void toolStripButtonLvl10_Click(object sender, EventArgs e)
{
    SelectedLevel = 10;
    toolStripButtonLvl1.Checked = false;
    toolStripButtonLvl2.Checked = false;
    toolStripButtonLvl3.Checked = false;
    toolStripButtonLvl4.Checked = false;
    toolStripButtonLvl5.Checked = false;

```

```

        toolStripButtonLv16.Checked = false;
        toolStripButtonLv17.Checked = false;
        toolStripButtonLv18.Checked = false;
        toolStripButtonLv19.Checked = false;
        toolStripButtonLv110.Checked = true;
    }

    //кнопка ответа
    private void toolStripButtonForAnswer_Click(object sender, EventArgs e)
    {

        FormForAnswer answerForm = new FormForAnswer();
        answerForm.SelectedLevel2 = SelectedLevel;
        answerForm.ShowDialog();
        ToolStripButton[] lvlButtons = { toolStripButtonLv11, toolStripButtonLv12,
        toolStripButtonLv13,          toolStripButtonLv14,toolStripButtonLv15,toolStripButtonLv16,
        toolStripButtonLv17,toolStripButtonLv18,toolStripButtonLv19, toolStripButtonLv110};
        if (answerForm.answerTruthCheck && SelectedLevel != 10)
        {
            lvlButtons[SelectedLevel].Enabled = true;
        }
        else if (answerForm.answerTruthCheck && SelectedLevel == 10)
        {
            MessageBox.Show("Мои поздравление, вы выиграли!");
        }
    }

    //изменение позиции элементов при изменении размеров формы
    private void mainForm_SizeChanged(object sender, EventArgs e)
    {
        locationChange();
    }

    private void начатьЗановоToolStripMenuItem_Click(object sender, EventArgs
e)
    {

```

```

        ToolStripButton[] lvlButtons = { toolStripButtonLvl1, toolStripButtonLvl2,
toolStripButtonLvl3, toolStripButtonLvl4, toolStripButtonLvl5, toolStripButtonLvl6,
toolStripButtonLvl7, toolStripButtonLvl8, toolStripButtonLvl9, toolStripButtonLvl10 };
        for (int i = 1; i < lvlButtons.Length; i++)
        {
            lvlButtons[i].Enabled = false;
        }
    }

    private void выходToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void buttonRemoveAll_Click(object sender, EventArgs e)
    {
        textBoxInput.Text = null;
        textBoxOutput.Text = null;
    }

    //Подсказка для учителя
    private void labelForClues_MouseDoubleClick(object sender, MouseEventArgs
e)
    {
        MessageBox.Show("Подсказка для учителя :) \r\n 1 уровень: a+1 \r\n 2
уровень: a+5 \r\n 3 уровень: a+15 " +
            "\r\n 4 уровень: a*2 \r\n 5 уровень: a*3 \r\n 6 уровень: a*2+1 \r\n 7 уровень:
a*2-2\r\n 8 уровень: a/2" +
            "\r\n 9 уровень: a/2+1 \r\n 10 уровень: a в степени 2.");
    }

    //Руководство к игре
    private void руководствоToolStripMenuItem_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Как играть в BlackBox:\r\n" +
            "1. Введите несколько чисел в левую белую колонку\r\n" +
            "(Не больше 4)начиная каждое число с новой строки\r\n" +
            "2. Нажмите на кнопку \"Считать\". Каждому числу с\r\n" +
            "левой колонки в правую будет записано число,\r\n" +
            "результат работы алгоритма\r\n" +
            "3. Если вы догадались, какой алгоритм используется\r\n" +
            "для обработки числа, то нажмите на кнопку с\r\n" +
            "восклицательным Знаком\r\n" +
            "4. Перед вами появится окно для Ответа. В столбце \r\n" +

```

```

        "номер 1 записано несколько чисел\r\n" +
        "5. Для того ,чтобы дать ответ , вам необходимо\r\n" +
        " обработать числа с столбцы 1 по алгоритму который\r\n" +
        " вы считаете верным и записать ответ в столбец 2,\r\n" +
        " каждый ответ должен находиться напротив числа,для\r\n" +
        " ответа нажмите на кнопку между столбцами" +
        "6. Если ваш ответ верный, вам станет доступен\r\n" +
        " следующий уровень с другим, более сложным алгоритмом,\r\n" +
        " и так до тех пор, пока вы не пройдете все уровни\r\n" +
        "*Для того, чтобы начать игру сначала, в верхнем меню\r\n" +
        " выберите пункт игра->Начать заново");
    }
}
}

```

Программный текст Окна для ответов:

```

using System;
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 BlackBox
{
    public partial class FormForAnswer : Form
    {
        public bool answerTruthCheck = true;
        Random Ran = new Random();
        double[] examples = new double[5];
        double[] answers = new double[5];
        public double SelectedLevel2;

        void locationChange()
        {
            buttonForAnswer.Left = panelForAnswer.Width / 2 - buttonForAnswer.Width / 2;
            buttonForAnswer.Top = panelForAnswer.Height / 2 - buttonForAnswer.Height / 2;
        }
    }
}

```

```

        buttonAnotherNumbers.Left = panelForAnswer.Width / 2 -
buttonAnotherNumbers.Width / 2;
        buttonAnotherNumbers.Top = buttonForAnswer.Top + buttonForAnswer.Height + 2;
        textBoxForExamples.Left = buttonForAnswer.Left - textBoxForExamples.Width -
20;
        textBoxForAnswers.Left = buttonForAnswer.Left + buttonForAnswer.Width + 20;
        textBoxForExamples.Top = buttonForAnswer.Top - textBoxForExamples.Height / 2
+ buttonForAnswer.Height / 2;
        textBoxForAnswers.Top = buttonForAnswer.Top - textBoxForAnswers.Height / 2 +
buttonForAnswer.Height / 2;
        label1.Left = textBoxForExamples.Left;
        label1.Top = textBoxForExamples.Top - label1.Height;
        label2.Left = textBoxForAnswers.Left;
        label2.Top = textBoxForAnswers.Top - label2.Height;
    }
    public FormForAnswer()
    {
        InitializeComponent();
        locationChange();
    }
    private void FormForAnswer_Load(object sender, EventArgs e)
    {
        completionIfTextBoxExample();
    }

    private void FormForAnswer_SizeChanged(object sender, EventArgs e)
    {
        locationChange();
    }
    void completionIfTextBoxExample()
    {
        textBoxForAnswers.Text = null;
        textBoxForExamples.Text = null;
        double example;
        bool samecheck = true;
        for (int i = 0; i < examples.Length; i++)
        {
            example = Ran.Next(1, 25);
            //если рандомное число совпадет с уже имеющимися, то пропускаем его
            {

```

```

        for (int j = 0; j <= i; j++)
            if (example == examples[j])
            {
                i--;
                samecheck = false;
                break;
            }
    }
    if (samecheck)
    {
        examples[i] = example;
        textBoxForExamples.Text += (examples[i] + "\r\n");
    }
    samecheck = true;
}
}

```

```

private void buttonForAnswer_Click(object sender, EventArgs e)
{
    MainForm mainF = new MainForm();
    string[] stringAnswers = textBoxForAnswers.Text.Split('\n');
    try
    {
        for (int i = 0; i < answers.Length; i++)
        {
            answers[i] = Convert.ToDouble(stringAnswers[i]);
        }
    }
    catch (FormatException)
    {
        MessageBox.Show("Вводите только цифры, без пустых строк между ними и  
для выделения целой части используйте запятую.");
    }
    catch (IndexOutOfRangeException)
    {
        MessageBox.Show(" дайте ответ на все цифры.");
    }

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

```

```

        if (OutputLvl(examples[i]) != answers[i])
        {
            //MessageBox.Show(Convert.ToString(OutputLvl(examples[i])));
            answerTruthCheck = false;
            break;
        }
    }

    if (answerTruthCheck)
    {
        buttonForAnswer.BackColor = Color.GreenYellow;
        MessageBox.Show("Праильно :)");
        this.Close();
    }
    else
    {
        buttonForAnswer.BackColor = Color.Red;
        MessageBox.Show("Не правильно :(");
        textBoxForAnswers.Text = null;
        answerTruthCheck = true;
    }
}

private double OutputLvl(double input)
{
    //для каждого уровня своё возвращаемое число
    switch (SelectedLevel2)
    {
        case 1:
        {
            return (input + 1);
        }

        case 2:
        {
            return (input + 5);
        }

        case 3:
        {
            return (input + 15);
        }
    }
}

```



```

    }

    case 4:
    {
        return (input * 2);
    }

    case 5:
    {
        return (input * 3);
    }

    case 6:
    {
        return (input * 2 + 1);
    }

    case 7:
    {
        return (input * 2 - 2);
    }

    case 8:
    {
        return (input / 2);
    }

    case 9:
    {
        return (input / 2 + 1);
    }
    case 10:
    {
        return Math.Pow(input, 2);
    }

}
return 0;
}

```

```

private void buttonAnotherNumbers_Click(object sender, EventArgs e)
{

```

```

        textBoxForExamples.Text = null;
    }
}
}

```

Общие Сведения

Приложение написано на объектно-ориентированном языке программирования C# разработанный для платформы Microsoft .NET Framework с помощью конструктора Windows Forms. программа в основном состоит из простых функций которые взаимодействуют друг с другом во время нажатия на различные кнопки на интерфейсе.

Приложение не требует специального ПО для запуска.

Функциональные ограничения

Значительным минусом является возможность смены алгоритма в уровнях только с помощью вмешательства в код самой программы. Сделать это с помощью пользовательского интерфейса невозможно.

Описание логической структуры

Основными методами на главной форме являются

1. метод “locationChange” который меняет расположение объектов интерфейса формы и вызывается при изменении размеров окна и при инициализации компонентов при старте программы
2. Метод “OutputLvl” который принимает число введенный пользователем и выбранный уровень и возвращает обработанная определенным алгоритмом уровня число

Основные методы окна для ответов

1. Тот же метод что и на главной форме “OutputLvl”

Используемые технические средства

при создании приложения “BlackBox” использовался исключительно Персональный компьютер средней мощности.

Входные данные

Основная Задача при обработке входных данных: исключить попадание в алгоритмы иных символов и знаков кроме как чисел как отрицательных, так и вещественных. при вводе иных знаков, с помощью Метода Try - Catch пользователю сообщается что он делает не так с помощью MessageBox.

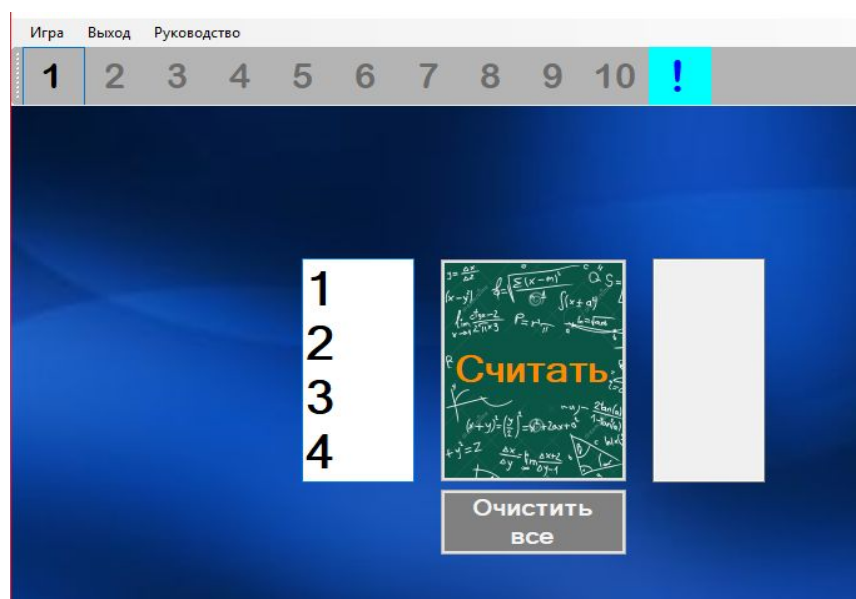


Рис.6. Входные Данные: 1 2 3 4 5



Рис.6. Входные данные: 19 4 20 25 8

Выходные данные

В обоих случаях, выходные представляют собой список элементов типа `double` который выводятся в свойства `Text` определенных `TextBox`-ов.

Внедрение

Руководство системного программиста

Дополнительные возможности Приложения

Была добавлена так называемая функция “Помощь учителю” представляющая собой бесцветный элемент типа `Label` в левом нижнем углу при нажатии двойным кликом на которую открывается окошко `MessageBox` с Описанием алгоритмов на всех уровнях.

Действия по настройке программы

Программа не требует и не предполагает каких либо настроек для запуска и работы с ним и готова к использованию сразу же после старта

Сведения о структуре

Программа состоит из двух основных окон:



Они связаны между собой. Передача данных между ними происходит только в одном случае:

1. Передается номер выбранного уровня с главной формы на второстепенную для проверки пользователя.

Сообщения выдаваемые Пользователю во время выполнения задачи

Если в Процессе Работы с приложением на главной форме в месте ввода данных ввести любой другой символ кроме числа , знаков “+” “-” “.”

Пользователю выводится следующее сообщение :

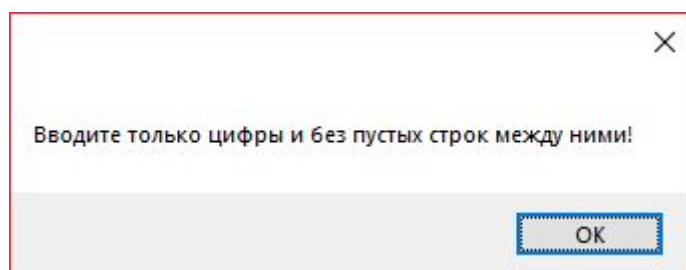


Рис.7 Сообщение о некорректности вводимых данных

В окне для ответа обязательным условием является запись ответов для всех пяти представленных чисел, не следование этому правилу вызывает следующее сообщение

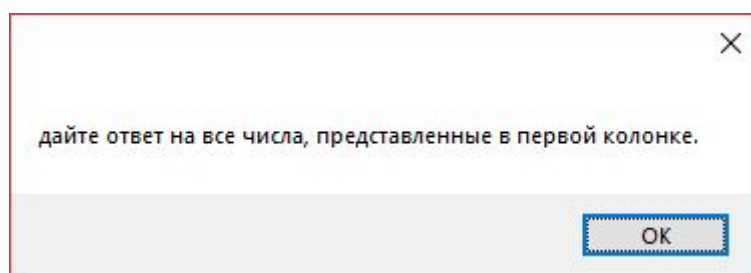


Рис.8 Сообщение о Неполном ответе во втором окне

Руководство программиста

Поддерживаемые операционные системы

- Windows 10 версии 1703 и выше: Домашняя, Pro, для образовательных учреждений и Корпоративная (выпуски LTSC и S не поддерживаются).
- Windows Server 2019: Standard и Datacenter
- Windows Server 2016: Standard и Datacenter
- Windows 8.1 : Core, Профессиональная и Корпоративная.
- Windows Server 2012 R2 : Essentials, Standard, Datacenter.
- Windows 7 с пакетом обновления 1 (SP1) (с последними обновлениями Windows) : Домашняя расширенная, Профессиональная, Корпоративная, Максимальная.
-

Минимальные системные требования

- Процессор с тактовой частотой не ниже 1,8 ГГц. Рекомендуется использовать как минимум двухъядерный процессор.
- 2 ГБ ОЗУ; рекомендуется 8 ГБ ОЗУ (минимум 2,5 ГБ при выполнении на виртуальной машине)
- Место на жестком диске: до 210 ГБ (минимум 800 МБ) свободного места в зависимости от установленных компонентов; обычно для установки требуется от 20 до 50 ГБ свободного места.
- Скорость жесткого диска: для повышения производительности установите Windows и Visual Studio на твердотельный накопитель (SSD)

- Видеоадаптер с минимальным разрешением 720p (1280 на 720 пикселей); для оптимальной работы Visual Studio рекомендуется разрешение WXGA (1366 на 768 пикселей) или более высокое.

Руководство оператора

Последовательность действий обеспечивающих загрузку, запуск, выполнение и завершение программы

1. Запуск файла “BlackBox.exe”
2. Ввод в левую колонку минимум одного числа
3. нажатие на кнопку “Считать” для выявления используемого алгоритма
4. Нажатие на кнопку “!”
5. В окне “Ответ” Ввести пять чисел (Каждый на новой строке) в столбце под номером 2 (результат обработки предлагаемым алгоритмом данных пяти чисел из столбца номер 1)
6. Для Выхода Нажатие Кнопки “Выход” на главной форме

Описание языка

C# является объектно-ориентированным языком, но поддерживает также и компонентно-ориентированное программирование. Разработка современных приложений все больше тяготеет к созданию программных компонентов в форме автономных и самоописательных пакетов, реализующих отдельные функциональные возможности. Важная особенность таких компонентов — это модель программирования на основе свойств, методов и событий. Каждый компонент имеет атрибуты, предоставляющие декларативные сведения о компоненте, а также встроенные элементы документации. C# предоставляет языковые конструкции, непосредственно поддерживающие такую концепцию

работы. Благодаря этому С# отлично подходит для создания и применения программных компонентов.

Вот лишь несколько функций языка С#, обеспечивающих надежность и устойчивость приложений: сборка мусора автоматически освобождает память, занятую уничтоженными и неиспользуемыми объектами; обработка исключений предоставляет структурированный и расширяемый способ выявлять и обрабатывать ошибки; строгая типизация языка не позволяет обращаться к неинициализированным переменным, выходить за пределы индексируемых массивов или выполнять неконтролируемое приведение типов.

Приложения и перечни

1. ИТ - Информационные технологии.
2. ЭВМ - Электронно-вычислительная машина.

Заключение

В ходе работы Было Разработано приложение “BlackBox” для замены ручной записи цифр и результатов, тем самым упрощая работу в частности учителям и приучая детей взаимодействию с ИВМ.

Был получен огромный опыт работы с более-менее реальным файлом

Список литературы

1. <https://docs.microsoft.com/ru-ru/?view=netframework-4.7.2-> хранилище документации Майкрософт для пользователей, разработчиков и ИТ-специалистов.
2. https://professorweb.ru/my/csharp/charp_theory/level1/index.php - Данное руководство было создано для быстрого и наглядного обучения всем тонкостям программирования на языке C#.
3. <http://www.cyberforum.ru/> - так же, иногда, когда я не помнил синтаксис некоторых операторов и т.п. я пользовался данным форумом