

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 04.08.2021 14:39:31

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d70e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии



ИЗУЧЕНИЕ РАЗВЕТВЛЕННЫХ АЛГОРИТМОВ

Методические указания по выполнению практических работ по дисциплине «Алгоритмические языки» для студентов всех форм обучения направления подготовки бакалавров 45.03.03 «Фундаментальная и прикладная лингвистика»

Курск 2019

УДК 004.43
Составитель Е.А. Петрик

Рецензент
Кандидат технических наук, доцент Т.И.Лапина

Изучение разветвленных алгоритмов: методические указания по выполнению практических работ / Юго-Зап. гос. ун-т; сост.: Е. А. Петрик. Курск, 2019. 14 с.: ил.11, табл. 1, Библиогр.: с.14.

Содержат краткие теоретические сведения о разветвленных алгоритмах, об условных операторах, а также приведены примеры и задания для практических работ.

Методические указания соответствуют требованиям программы по направлению подготовки бакалавров: 45.03.03 «Фундаментальная и прикладная лингвистика».

Предназначены для студентов всех форм обучения направления подготовки бакалавров 45.03.03 «Фундаментальная и прикладная лингвистика».

Текст печатается в авторской редакции

Подписано в печать 31.10.2019. Формат 60x84 1/16.
Усл. печ. л. 0,7. Уч. – изд. л. 0,6 .Тираж 100 экз. Заказ 640 . Бесплатно.
Юго - Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Цель работы

Изучение основных разветвленных алгоритмов, а также условных операторов языка с#. Создание программы с использованием разветвленных алгоритмов.

Основные понятия

Разветвленный алгоритм – это такой тип организации алгоритма, в котором в зависимости от вычисления какого-либо выражения осуществляется та или иная последовательность действий. Обычно такое выражение называется условием. Разветвленные алгоритмы в языках программирования записываются в виде условных операторов.

Условные операторы бывают полными и неполными, а также существуют операторы выбора.

Блок-схемы условных операторов изображены на рисунках 1, 2 и 3.

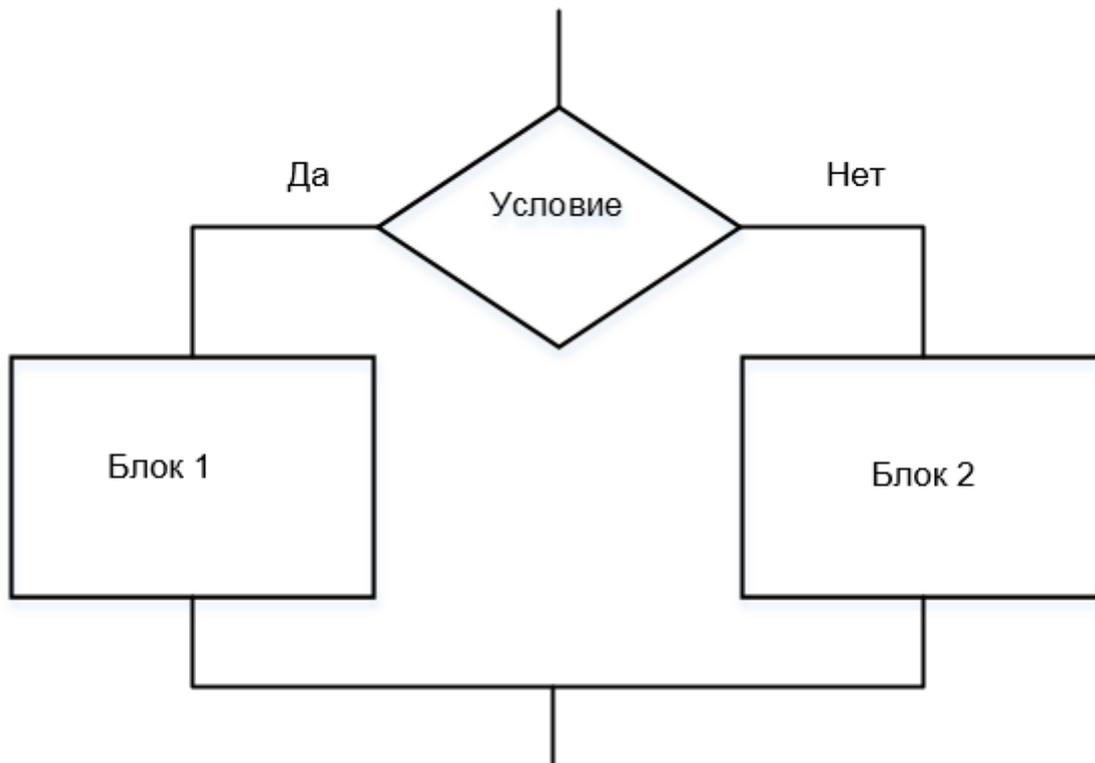


Рисунок 1 – Полный условный оператор



Рисунок 2 – Неполный условный оператор

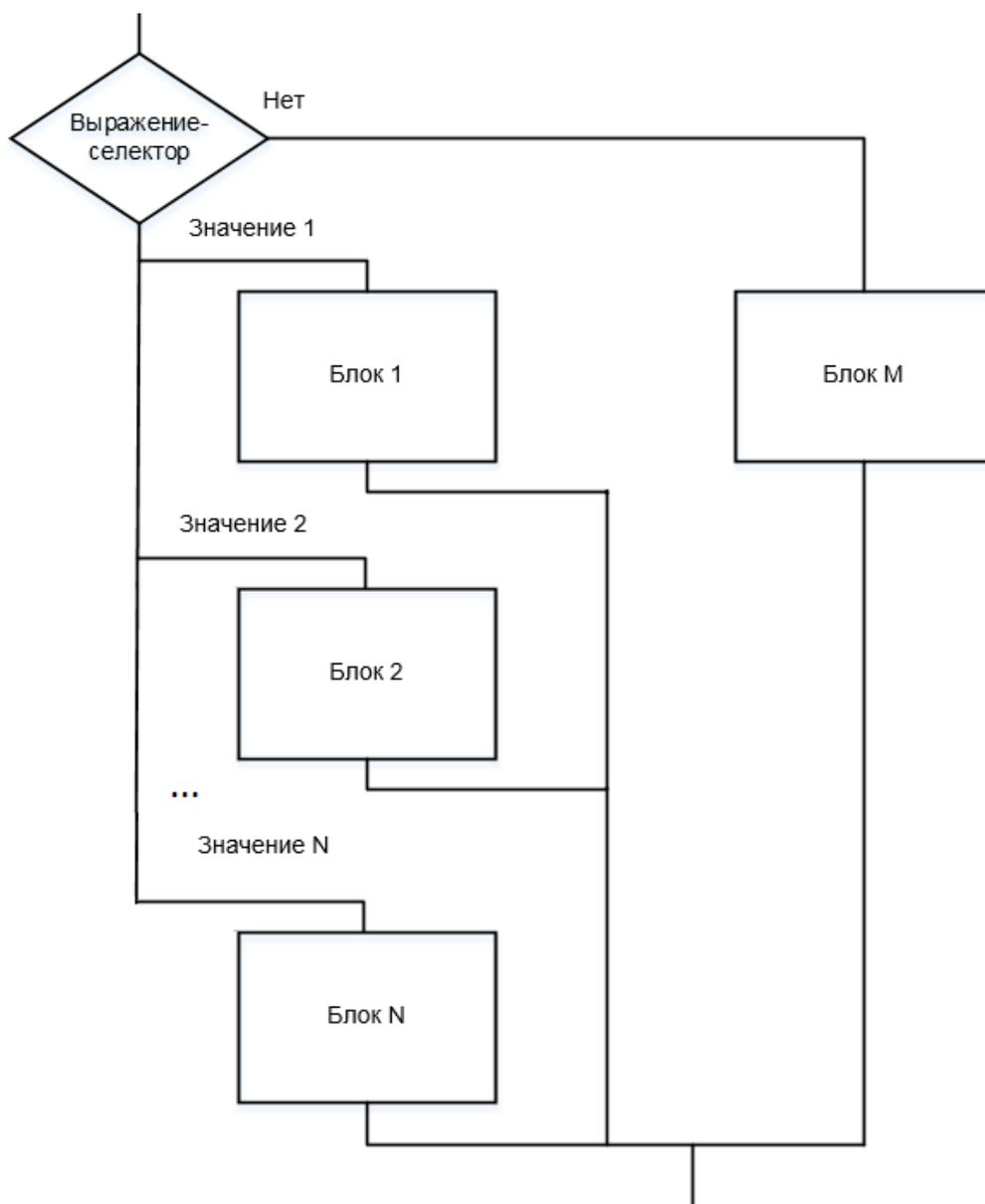


Рисунок 3 – Оператор выбора

В языке *c#* условные операторы записываются с помощью ключевых слов *if*, *else*, *switch*, *case*.

Общий вид полного условного оператора:

```
if (условное выражение)
{
    блок операторов;
}
```

```
else
{
    блок операторов;
}
```

Общий вид неполного условного оператора:

```
if (условное выражение)
{
    блок операторов;
}
```

Общий вид оператора выбора:

```
switch (выражение-селектор)
{
    case 1:
        блок операторов;
        break;
    case 2:
        блок операторов;
        break;
    ...
    case N:
        блок операторов;
        break;
    default:
        блок операторов;
        break;
}
```

В разветвленных алгоритмах обычно необходимо вычисление условия. В условии можно записывать логические операции (&& (and), || (or), ! (not) и др.), а также операции сравнения (>, <, >=, <= и др.) и скобки. Таблица истинности логических операций представлена в таблице 1.

Таблица 1 – Таблица истинности логических операций

a	b	a&&b	a b	!a
0	0	0	0	1
0	1	0	1	0
1	0	0	1	
0	0	1	1	

Например, выражение «Если на улице холоднее +5° и идет дождь, то гулять сегодня на улицу не идем, иначе идем гулять на улицу» можно записать на языке *c#* следующим образом:

```
int temp=10;
bool rain=true;
if (temp<5&&rain)
{
    label1.Text="Идем гулять на улицу";
}
else
{
    label1.Text="Не идем гулять на улицу";
}
```

Задание для практической работы

Задание 1

По количеству баллов определить оценку.

Для решения этой задачи необходимо использовать полный условный оператор *if-else*.

Создайте новое приложение *Windows Forms*.

На форму добавьте компоненты *Label* (текстовая надпись), *TextBox* (поле ввода текста), *Button* (кнопка).

Внешний вид формы с компонентами представлен на рисунке 4.

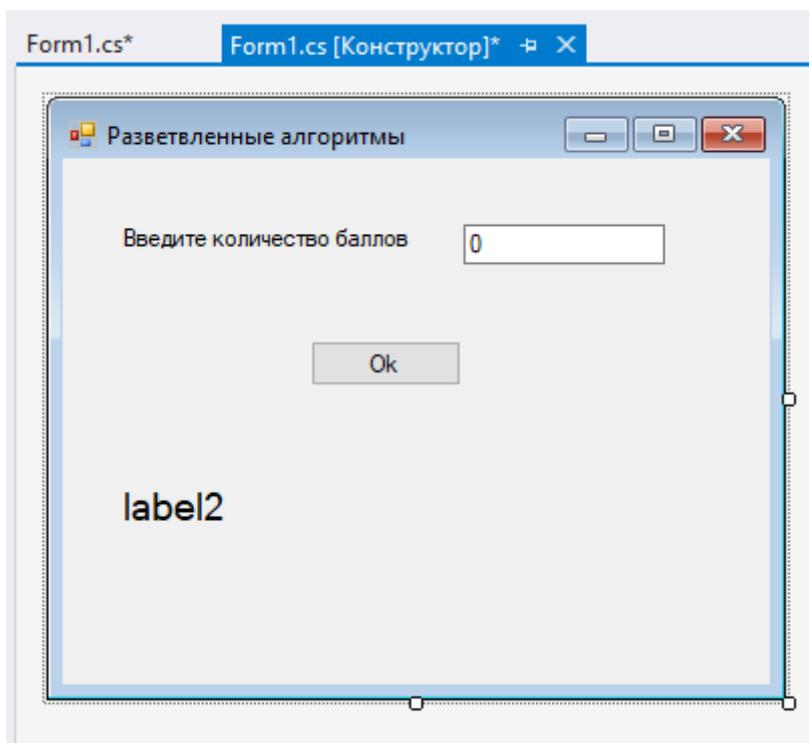
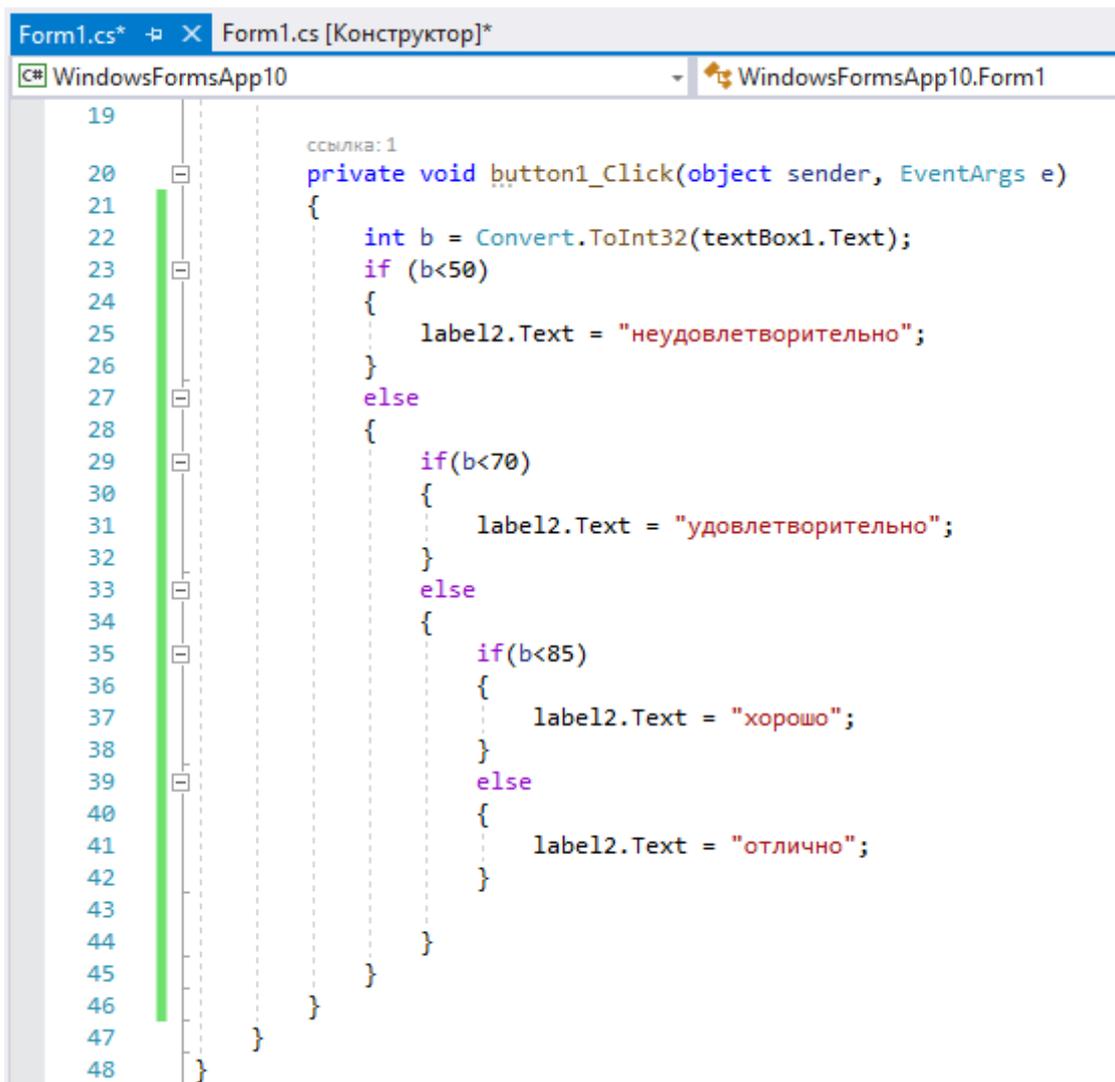


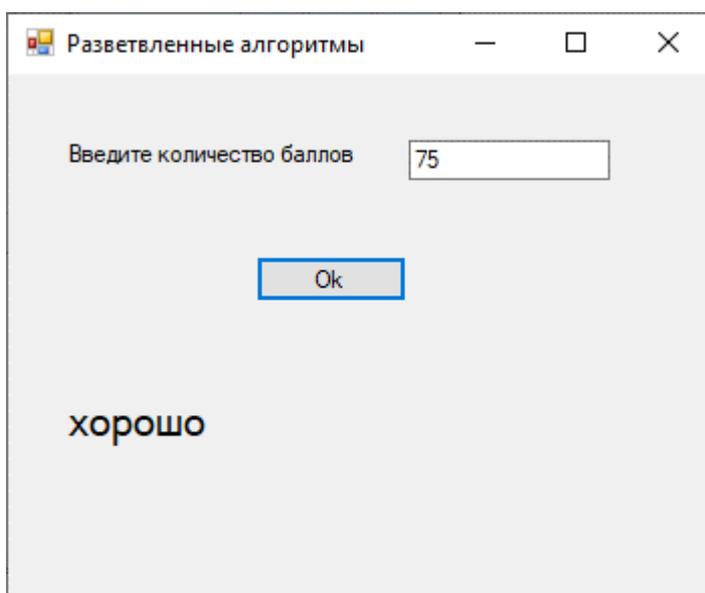
Рисунок 4 – Внешний вид формы

Для решения задачи необходимы следующие *входные данные*: количество баллов (целое число от 0 до 100). Входные данные вводятся в поле для ввода данных: `TextBox`. Вычисления будут производиться по нажатию кнопки `Button`. Результат будет выводиться в `label2.Text`. Текст программы приведен на рисунке 5. Результат работы программы приведен на рисунке 6.



```
19
20 ссылка: 1
21 private void button1_Click(object sender, EventArgs e)
22 {
23     int b = Convert.ToInt32(textBox1.Text);
24     if (b<50)
25     {
26         label2.Text = "неудовлетворительно";
27     }
28     else
29     {
30         if(b<70)
31         {
32             label2.Text = "удовлетворительно";
33         }
34         else
35         {
36             if(b<85)
37             {
38                 label2.Text = "хорошо";
39             }
40             else
41             {
42                 label2.Text = "отлично";
43             }
44         }
45     }
46 }
47
48 }
```

Рисунок 5 – Внешний вид формы



Разветвленные алгоритмы

Введите количество баллов

Ok

хорошо

Рисунок 6 – Внешний вид формы

Этапы практической работы

Запустите приложение Microsoft Visual Studio

Задание 2

По введенной начальной русской букве напечатать названия соответствующих ей цветов радуги.

Для решения этой задачи необходимо использовать оператор выбора switch.

Этапы практической работы

Запустите приложение Microsoft Visual Studio

Создайте новое приложение Windows Forms.

На форму добавьте компоненты Label (текстовая надпись), TextBox (поле ввода текста), Button (кнопка), DataGridView (таблица)

Внешний вид формы с компонентами представлен на рисунке 7.

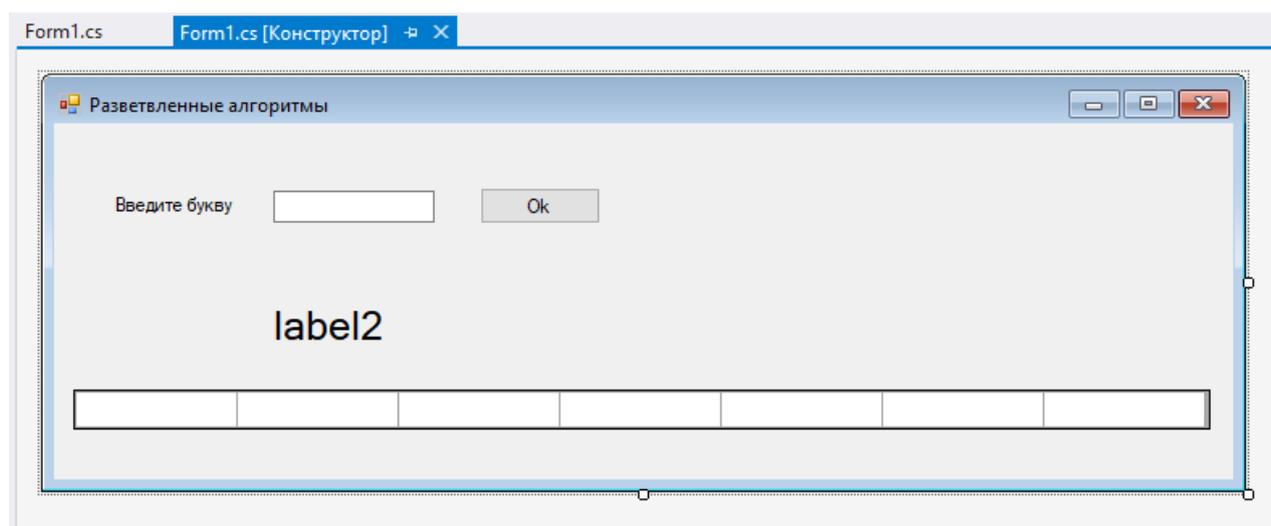


Рисунок 7 – Внешний вид формы

У компонента DataGridView необходимо настроить свойства строк и столбцов через редактор свойств: ColumnHeadersVisible=False; RowHeadersVisible=False; Columns (рисунок 8).

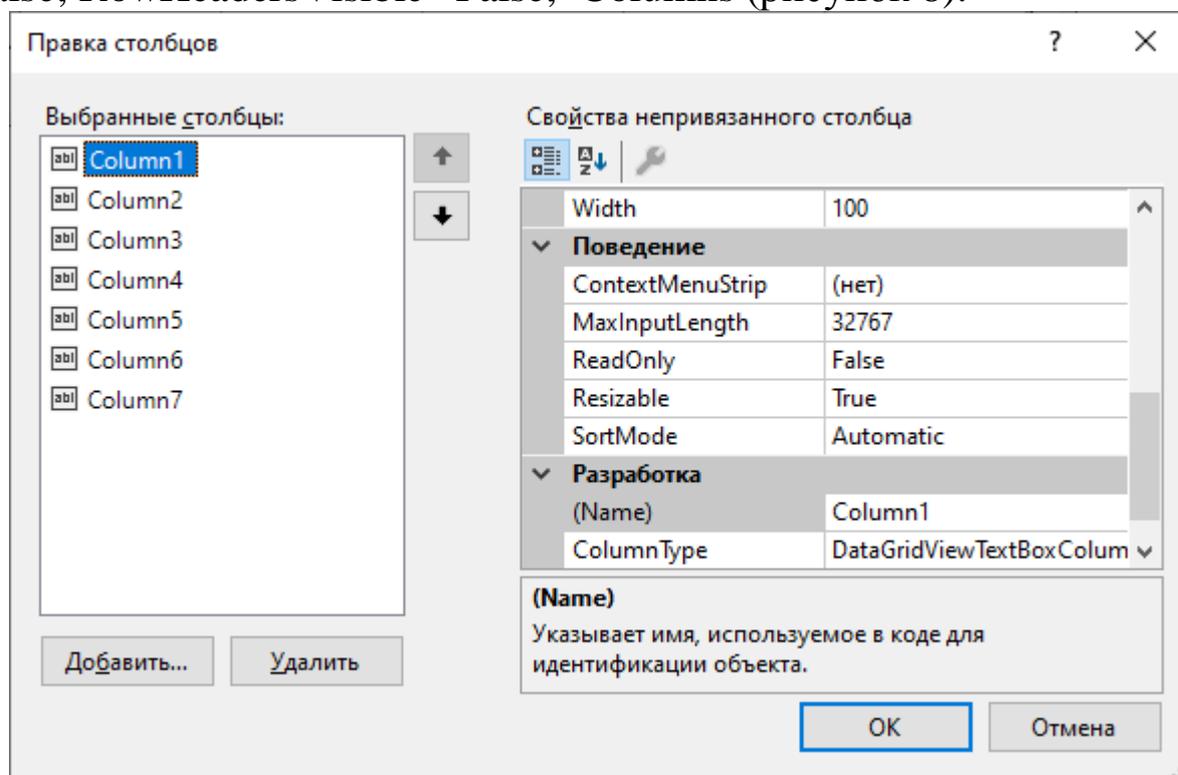


Рисунок 8 – Настройка свойств

Для решения задачи необходимы следующие *входные данные*: первая буква цвета. Входные данные вводятся в поле для ввода данных: TextBox. Вычисления будут производиться по нажатию кнопки Button. Результат будет выводиться в label2.Text в виде названия цвета, а также сам компонент label2 будет окрашиваться введенным цветом. Для наглядности цветом радуги будем окрашивать соответствующую ячейку в DataGridView. Текст программы приведен на рисунке 9. Результат работы программы приведен на рисунках 10 и 11.



```
Form1.cs [Конструктор]
WindowsFormsApp9
WindowsFormsApp9.Form1
textBox1

ссылка: 1
20 private void button1_Click(object sender, EventArgs e)
21 {
22
23     switch (textBox1.Text)
24     {
25     case "к":
26     case "К":
27         label2.Text = "красный";
28         label2.ForeColor = Color.Red;
29         dataGridView1.Columns[0].DefaultCellStyle.BackColor = Color.Red;
30         break;
31     case "о":
32     case "О":
33         label2.Text = "оранжевый";
34         label2.ForeColor = Color.Orange;
35         dataGridView1.Columns[1].DefaultCellStyle.BackColor = Color.Orange;
36         break;
37     case "ж":
38     case "Ж":
39         label2.Text = "желтый";
40         label2.ForeColor = Color.Yellow;
41         dataGridView1.Columns[2].DefaultCellStyle.BackColor = Color.Yellow;
42         break;
43     case "з":
44     case "З":
45         label2.Text = "зеленый";
46         label2.ForeColor = Color.Green;
47         dataGridView1.Columns[3].DefaultCellStyle.BackColor = Color.Green;
48         break;
49     case "г":
50     case "Г":
51         label2.Text = "голубой";
52         label2.ForeColor = Color.Blue;
53         dataGridView1.Columns[4].DefaultCellStyle.BackColor = Color.Blue;
54         break;
55     case "с":
56     case "С":
57         label2.Text = "синий";
58         label2.ForeColor = Color.Navy;
59         dataGridView1.Columns[5].DefaultCellStyle.BackColor = Color.Navy;
60         break;
61     case "ф":
62     case "Ф":
63         label2.Text = "фиолетовый";
64         label2.ForeColor = Color.Purple;
65         dataGridView1.Columns[6].DefaultCellStyle.BackColor = Color.Purple;
66         break;
67     default:
68         label2.Text = "не определено";
69         break;
70     }
71 }
72
73
```

Рисунок 9 – Окно редактора кода

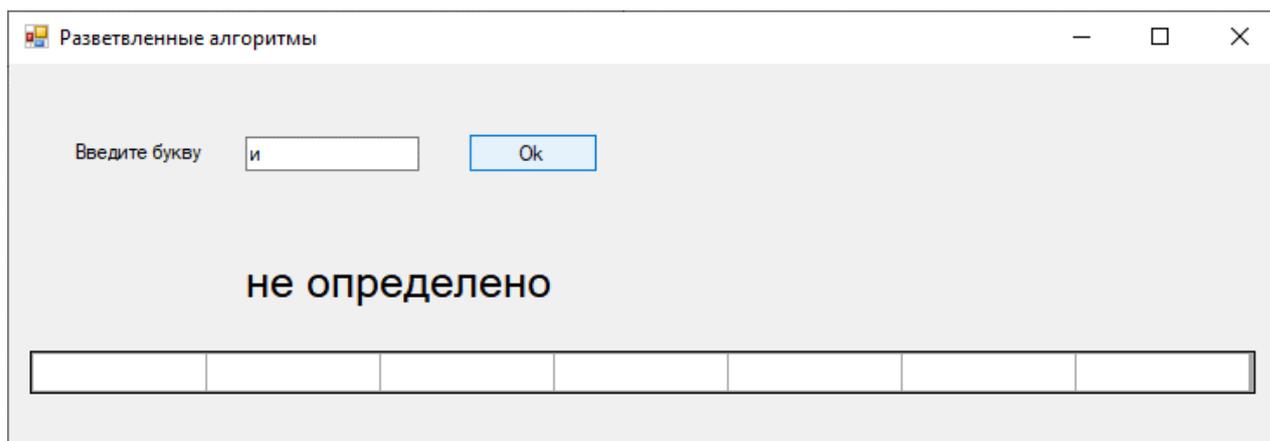


Рисунок 10 – Пример работы программы

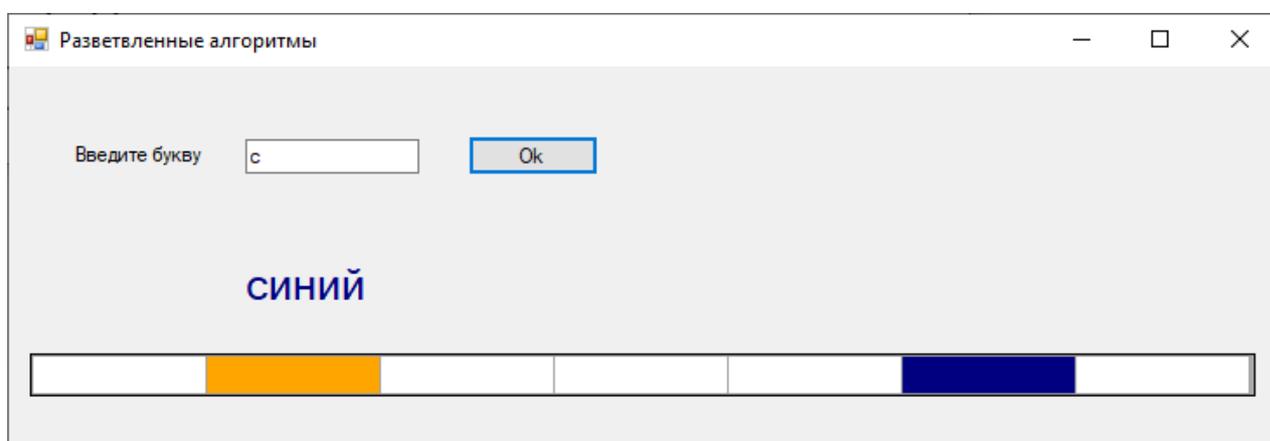


Рисунок 11 – Пример работы программы

Индивидуальные задания для практической работы

Составьте алгоритм и напишите программу для решения двух заданий.

1. Тема: «Условные операторы»

Определить рабочие и выходные дни в месяце в зависимости от того, какой это месяц и на какой день недели выпадает первое число месяца.

2. Тема: «Операторы выбора»

2.1. По введенной цифре от 1 до 9 напечатать их названия на русском языке.

2.2. По введенной цифре от 1 до 7 напечатать названия соответствующих дней недели на русском языке.

- 2.3. По введенной цифре от 1 до 12 напечатать названия соответствующих месяцев на русском языке.
- 2.4. По введенной цифре от 1 до 9 напечатать их названия на английском языке.
- 2.5. По введенной цифре от 1 до 12 напечатать названия соответствующих месяцев на английском языке.
- 2.6. По введенной цифре от 1 до 7 напечатать названия соответствующих дней недели на английском языке.

Контрольные вопросы

1. Что такое разветвленный алгоритм?
2. Какие типы разветвленных алгоритмов бывают?
3. Что такое условное выражение?
4. Какие операции могут присутствовать в условных выражениях?
5. Какие операторы существуют в языке C# для организации разветвленных алгоритмов?

Список литературы

1. Справочник по C# [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/> (дата обращения: 20.08.19)
2. Голицына О. Л. Программное обеспечение [Текст] : учебное пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - 3-е изд., перераб. и доп. - М. : Форум, 2010. - 201 с.
3. Иванова Г. С. Технология программирования [Текст] : учебник / Г. С. Иванова. – М. : Кнорус, 2011. – 336 с.