МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Пензенский государственный технологический университет»

(ПензГТУ)

''Кафедра прикладной информатики''

ОТЧЕТ

о выполнении лабораторной работы № 3.

по дисциплине «Информационные технологии»

Выполнил студент

группы 19ИЭ1бзи Иванов Е. А.

Проверила: Голобокова Елена Михайловна

Пенза, 2019

Лабораторная работа №3.

## Разработка программ с разветвленной структурой

## 1.1. Цель работы

Приобретение навыков в составлении и отладке программ на языке C# с использованием операторов: условного перехода *if*, разветвления *switch..case*, обработки исключительных ситуаций  *try..catch.*

## 1.2. Задание на лабораторную работу

1. Освоить функции системы MicrosoftVisualStudio по отладке программы (п. 5).

2. Составить программу в соответствии с требованиями и вариантом.

Примечание: для увеличения сложности программы были использованы формы. В последнем варианте была применена обфускация кода. Код после обфускации представлен не будет, так как разобрать код не представляется возможным. Вместо switch-case будет применена форма главного меню. (switch я применял в предыдущих работах)

## Текст программы (файл mainForm.CS)

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 Laboratory3

{

publicpartialclassForm1 : Form

{

publicForm1()

{

InitializeComponent();

}

privatevoid вариант1ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form4 form4 = new Form4();

form4.Show();

}

privatevoid вариант2ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form5 form5 = new Form5();

form5.Show();

}

privatevoid вариант3ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form6 form6 = new Form6();

form6.Show();

}

privatevoid вариант4ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form7 form7 = new Form7();

form7.Show();

}

privatevoid вариант5ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form2 form2 = new Form2();

form2.Show();

}

privatevoid вариант6ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form8 form8 = new Form8();

form8.Show();

}

privatevoid вариант7ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form9 form9 = new Form9();

form9.Show();

}

privatevoid вариант8ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form10 form10 = new Form10();

form10.Show();

}

privatevoid вариант16ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form11 form16 = new Form11();

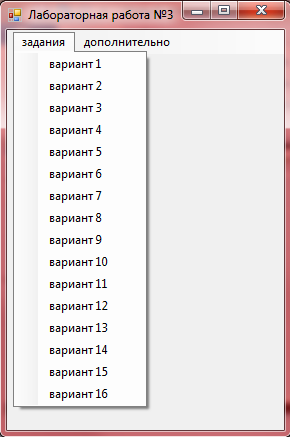
form16.Show();

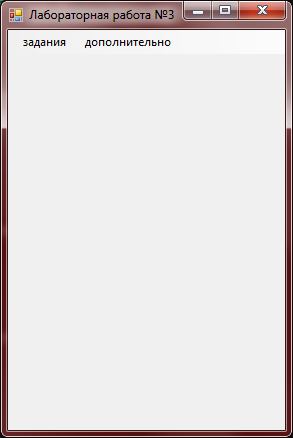
}

}

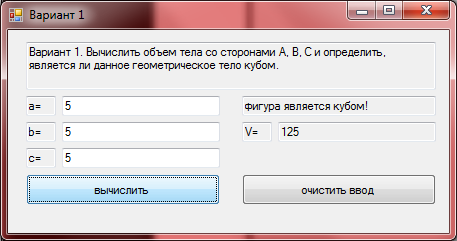
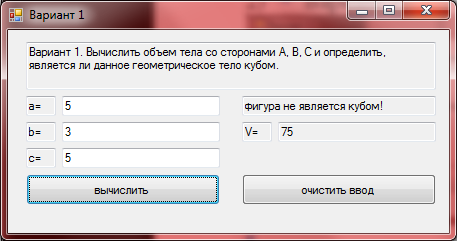
}

## Примеры работы программы

1. Главное меню.



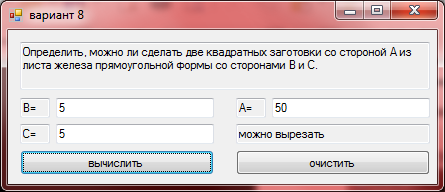
1. Вариант 1



Действительно,

не является кубом, поскольку не все стороны равны. является кубом, поскольку все стороны равны.

1. Вариант 8



действительно, так как площадь двух заготовок меньше, чем площадь металлического листа.

Во всех тестовых случаях программа работает корректно.

**Дополнительные вопросы:**

1. Назначение, формы записи и порядок выполнения оператора условия *if.*

*оператор IF служит для определения условия выполнения задачи и выбора оптимального решения на основе логической функции.*

*примечание (несмотря на то, что выражение может иметь любой тип, на выходе из условия всегда есть скрытое значение типа boolean, и в зависимости от условия оно принимает два значения, которые можно использовать в дальнейшей программе не объявляя их). форма записи алгоритма выглядит как*

*if (выражение) {*

*[program body]*

*}*

*else {*

*[program body 2]*

*}*

1. Особенности использования вложенных условных операторов.

вложенные условные операторы предназначены в первую очередь для более подробного выбора нескольких вариантов внутри нескольких вариантов, так же есть тип рекурсивного вызова, который основан на похожем принципе. так или иначе, данный вид условий крайне полезен при работе с большим количеством значений и вариантов выполнения, а так же использовался для построения простейших и примитивных искусственных интеллектов.

1. Каковы отличия оператора выбора *switch*..*case* от оператора условия *if*?

оператор выбора и оператор условий предназначены для разных вещей, тем не менее имеют схожую структуру реализации. В первую очередь они отличаются синтаксисом, после уже манерой и стилем применения. Проще говоря, if -else лучше использовать, когда всего пара вариантов программы, а switch-case проще использовать для более множественного выбора.

1. Какие правила должны выполняться при использовании оператора выбора *switch*..*case*?

правила? синтаксис и все.

1. Назначение оператора try..catch..finally.

обработка ошибок. Расписывать много, но если вкратце, то try ловит ошибку при выполнении кода, catch реагирует на пойманную ошибку, и выполняет тот или иной код, который указан при нем, finally выполняет код даже в случае ошибки.

1. Для чего нужна отладка программы?

ответ в вопросе, честно.

1. Как выполнять программу в пошаговом режиме?

есть несколько способов. пошаговый из клавиатуры (hot-key) и пошаговый из меню.

1. Как поставить точку останова?

выбрать строку и выбрать пункт в меню. (Никогда ими не пользуюсь.)

1. Как отменить точки останова?

обратная ситуация предыдущему вопросу.

примечание автора: Данная работа выполнена с учетом на профессионала, а потому задание было мною же усложнено. вместо оригинального задания использовались формы в режиме net.Framework. сам исходный код добавляю на свой гитхаб, а так же отсылаю вам рабочую программу и исходный код в архиве.