

Министерство образования и науки Российской Федерации Федеральное
государственное бюджетное образовательное учреждение высшего
образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра комплексной информационной безопасности электронно-
вычислительных систем (КИБЭВС)

ГЕНЕРИРОВАНИЕ И ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ

Отчет по практической работе №6 по
дисциплине «Языки программирования»

Студенты гр.728-2

_____ Д. Р. Геворгян

Принял

Преподаватель кафедры

КИБЭВС

_____ М. А. Кальнеус

Томск 2019

1 Введение

Целью данной работы является изучение различных видов исключительных ситуаций и методов их обработки.

2 Ход работы

2.1 Программа с исключительной ситуацией

Была написана программа, которая обрабатывает исключительную ситуацию.

```
using System;
using System.Windows.Forms;

namespace lab6
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                if (textBox1.Text.Length != 0)
                    MessageBox.Show(Convert.ToInt32(textBox1.Text).ToString(), "Балдѣж");
                else
                    MessageBox.Show("Вы ничего не ввели!", "Отстой");
            }
            catch
            {
                MessageBox.Show("О нет! Нужно ввести цифры, а не буквы!", "Ну не тупи :с");
            }
        }
    }
}
```

Результат выполнения программы (Рисунок 2.1 – 2.2).

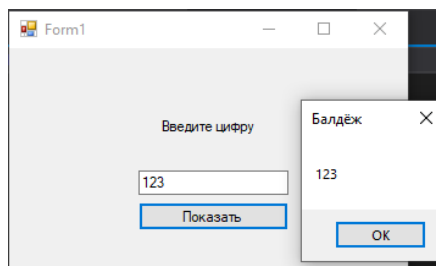


Рисунок 2.1 – Успешный результат работы программы

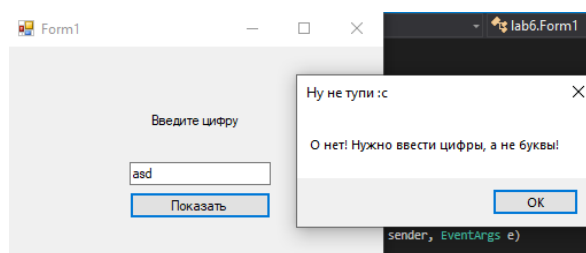


Рисунок 2.2 – Неудачный результат работы программы

2.2 Программа со схемой Бертана Мейера

Была доработана программа. Добавлены 3 попытки сделать верное действие.

```
using System;
using System.Windows.Forms;

namespace lab6
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        int count = 0;

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                if (textBox1.Text.Length != 0)
                {
                    MessageBox.Show(Convert.ToInt32(textBox1.Text).ToString(), "Балдёж");
                    count = 0;
                }
                else
                {
                    MessageBox.Show("Вы ничего не ввели!", "Отстой");
                }
            }
            catch
            {
                if (count > 2)
                {
                    MessageBox.Show("ЧТО НЕПОНЯТНОГО В СЛОВЕ ЦИФРА???\n\nПопыток больше не осталось", "Ну и дурааак...");
                    Application.Exit();
                }
                else
                {
                    MessageBox.Show($"Нужно ввести цифры, а не буквы!\n\nКоличество оставшихся попыток равно {3 - count}", "Хватит тупить :с");
                    count++;
                }
            }
        }
    }
}
```

Результат выполнения программы (Рисунок 2.3 – 2.4).

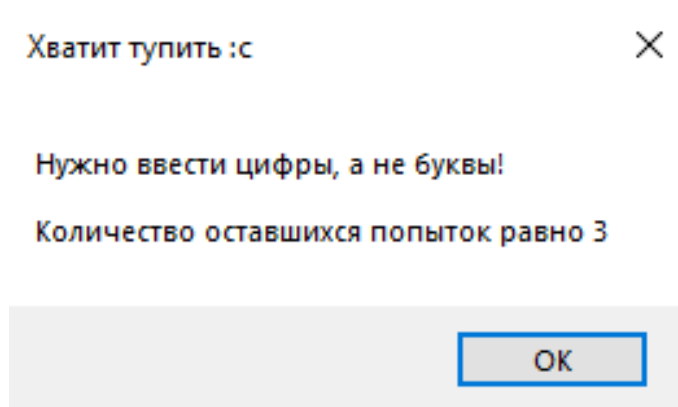


Рисунок 2.3 – Результат работы программы

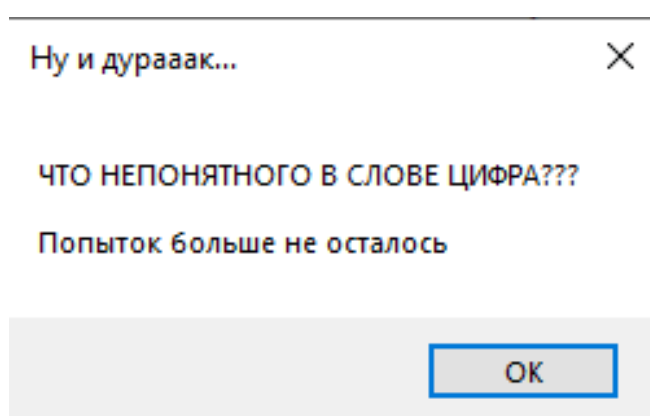


Рисунок 2.4 – Результат работы программы

2.3 Теоретическая часть

1. Понятие «исключительная ситуация».

Это необычное, аварийное событие (исключительная ситуация), которое обнаруживается аппаратно или программно и требует специальной обработки (обработки исключения). Обработка производится обработчиком исключения (ловушкой исключения).

2. Синхронные и асинхронные исключения.

Синхронные исключения могут возникнуть только в определённых, заранее известных точках программы. Примерами таких исключений могут быть: ошибки чтения файла, нехватка памяти, ошибка деления на ноль и т.д.

Асинхронные исключения могут возникать в любой момент времени и не зависят от того, какую конкретно инструкцию программы выполняет система.

Примерами таких исключений могут быть: аварийный отказ питания, поступление новых данных и т.д.

3. Структурная обработка исключений.

Структурная обработка исключений — механизм обработки программных и аппаратных исключений, позволяющий программистам контролировать обработку исключений.

4. Неструктурная обработка исключений.

Это механизм регистрации функций или команд-обработчиков для каждого возможного типа исключения. Язык или системные библиотеки предоставляю программисту как минимум две стандартные процедуры: регистрации обработчика и разрегистрации обработчика. Вызов первой из них «привязывает» обработчик к определённому исключению, вызов второй —отменяет эту «привязку».

5. Механизмы обработки исключений с возвратом и без возврата.

Механизм языков программирования предназначенный для описания реакции программы на ошибки времени выполнения и другие возможные проблемы, которые могут возникнуть при выполнении программы.

9. Схема Бертрана Мейера обработки исключительных ситуаций. Схема, в которой система дает юзеру исправить ситуацию. Иначе система передает управление вызывающему модулю, который и должен теперь попытаться исправить ситуацию.

3 Заключение

Были изучены различные виды исключительных ситуаций и методы их обработки. Создана программа.