

Министерство образования и науки Российской Федерации

ФГАОУ ВО «КФУ им. В.И. Вернадского»

физико-технический институт (структурное подразделение)

Кафедра компьютерной инженерии и моделирования

Шор Константин Александрович

отчет по лабораторной работе №4
по дисциплине «**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ**»

Направление подготовки:
09.03.01 "Информатика и вычислительная техника"

Оценка - "хорошо"



Симферополь, 2021

Лабораторная работа №4.

Тема: Обработка исключительных ситуаций

Цель работы: научиться на практике обрабатывать исключения, генерировать собственные исключения, отлавливать исключения различных типов, проверять числовые данные на выход за границы значений.

Описание ключевых понятий:

Exception – представляет ошибки, которые происходят во время выполнения приложений. Наследуется от Object/

Try/catch/ Finally– при возникновении исключений CLR ищет оператор catch, который обрабатывает это исключение. Если текущий выполняемый метод не содержит блок catch, среда CLR выполняет поиск в методе, который вызвал текущий метод, и так далее по стеку вызовов. Если блок catch не находится, то среда CLR отображает пользователю сообщение о необработанном исключении и останавливает выполнение программы. Порядок выполнения: если блок try не нашел исключений, то начинается выполнение блока Finally. Если же исключение было обнаружено в блоке try, то начинается выполнение блока catch и уже после блока Finally.

Checked – используется для явного включения проверки на переполнение при выполнении арифметических операций и преобразований с данными целого типа.

Throw – сообщает о возникновении исключения во время выполнения программы.

Перед выполнением лабораторной работы изучена следующая литература:

1. Изучил презентацию лектора: «Обработка исключений в C#»
2. Прочитана 23 лекция Биллиг В.А. Основы объектного программирования на языке C#
3. Просмотрен видеоурок на ютуб канале SimpleCode
4. Изучил нужные сведения на сайте MSDN Microsoft и Metanit.com.

Выполнены 3 задания, описанных в методических указаниях к выполнению лабораторных работ.

Задание 1. Написать с использованием конструкций Switch, Try,Catch метод анализа опасных состояний оборудования компьютера.

```
this.textBlock1.Text = "Initiating test sequence: ";
SwitchDevices.Switch sd = new SwitchDevices.Switch();
try
{
    // Step 1 - disconnect from the Power Generator
    if (sd.DisconnectPowerGenerator() == SwitchDevices.SuccessFailureResult.Fail)
    {
        this.textBlock1.Text += "\nStep 1: Failed to disconnect power generation system";
    }
    else
    {
        this.textBlock1.Text += "\nStep 1: Successfully disconnected power generation system";
    }
}
catch(SwitchDevices.PowerGeneratorCommsException) { textBlock1.Text += "\n*** Exception in step 1: "; }
```

Рис.1 Вылавливание ошибки

```
if (rand.Next(1, 20) > 19) throw new CoolantTemperatureReadException("Failed to read backup coolant system temperature");
if (rand.Next(1, 20) > 19) throw new CoolantPressureReadException("Failed to read backup coolant system pressure");
return c;
}
```

Рис.2 Выкидывание ошибки

```
Ссылка: 4
public class PowerGeneratorCommsException : Exception
{
    Ссылка: 1
    public PowerGeneratorCommsException(string message) : base(message) { }
}

Ссылка: 5
public class CoolantSystemException : Exception
{
    Ссылка: 2
    public CoolantSystemException(string message) : base(message) { }
}
```

Рис.3 Наследование

Задание 2. Обработка исключительных состояний при вычислении произведения матриц.

```
try
{
    for (int i = 0; i < matrix3.GetLength(1); i++)
    {
        for (int j = 0; j < matrix3.GetLength(0); j++)
        {
            matrix3[i, j] = 0;
            for (int k = 0; k < matrix1.GetLength(0); k++)
            {
                matrix3[i, j] += matrix1[k, j] * matrix2[i, k];
            }
        }
    }
    Matrix.initializeGrid(ref Result, matrix3);
}
catch
{
    MessageBox.Show("Ошибка");
}
```

Рис. 4 Обработка ошибки в матрицах

Задание 3. Использование Checked и Anchecked для обработки дополнения целых чисел.

```
try
{
    one = int.Parse(NumberOne.Text);
    Result.IsEnabled = true;
}
catch
{
    MessageBox.Show("Слишком большое первое число");
    Result.IsEnabled = false;
}

try
{
    two = int.Parse(NumberTwo.Text);
    Result.IsEnabled = true;
}
catch
{
    MessageBox.Show("Слишком большое второе число");
    Result.IsEnabled = false;
}

try
{
    checked { result = one * two; }
    Check.IsChecked = false;
}
catch
{
    result = one * two;
    Check.IsChecked = true;
}
Result.Text = result.ToString();
```

Рис.5 Обработка исключений

Представлены проект, реализованных в Visual Studio Common Eddition 2019.

Проекты представлены преподавателю в электронной форме, продемонстрирована их работоспособность, разъяснены детали программного кода.

Заданные вопросы преподавателем:

- 1) Где в коде вылавливаются исключения?**
- 2) С помощью какого оператора выкидываться исключения?**
- 3) Как сделать так, чтобы при возникновение необработанной ошибки система не крашилась, а переходила в аварийный режим ?**
- 4) Есть ли методы у Exсeption методы, которые могут найти прошлые исключения?**