

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Филиал
«Минский радиотехнический колледж»

Учебный предмет
«Конструирование программ и языки программирования»

Инструкция
по выполнению лабораторной работы №15
«Разработка, отладка и испытание программ обработки исключительных
ситуаций»

Минск 2024 г.

Лабораторная работа № 15

Тема работы: «Разработка, отладка и испытание программ обработки исключительных ситуаций»

1 Цель работы

Сформировать умения обрабатывать исключительные ситуации, возникновение которых возможно в процессе работы программы.

2 Задание

Номер варианта соответствует номеру по списку в журнале.

Для программ реализовать максимально возможное количество обработчиков исключительных ситуаций (не менее 2-ух исключений).

1. Создать программу для вычисления корней квадратного уравнения через дискриминант. Формула дискриминанта: $b^2 - 4ac$
2. Создать программу для вычисления значения выражения $y = \frac{ax^2+ab+z^3}{x+z}$
3. Создать программу для вычисления значения выражения $\sqrt{(x-y)}$
4. Создать программу для вычисления значения выражения $\log_a b$
5. Создать программу для вычисления суммы членов геометрической прогрессии $S_n = \frac{b_1(1-q^n)}{1-q}$
6. Создать программу для вычисления значения выражения $\sqrt[n]{a^k}$
7. Создать программу для вычисления значения выражения $\sqrt[n]{ab}$
8. Создать программу для вычисления значения выражения $\log_a \left(\frac{x}{y}\right)$
9. Создать программу для вычисления значения выражения $a^2 - b^2$
10. Создать программу для вычисления тангенсов углов.
11. Создать программу для вычисления котангенса углов.
12. Создать программу для вычисления значения выражения $(\arcsin x)'$
13. Создать программу для вычисления значения выражения $m = \log_a b$
14. Создать программу для вычисления значения выражения $* |x^n| = |x|^n$
15. Создать программу для вычисления значения выражения $\frac{1-\cos 2\alpha}{1+\cos 2\alpha}$

3 Оснащение работы

ПК, среда Visual Studio 2019, MSword.

4 Основные теоретические сведения

Язык C# предоставляет разработчикам возможности для обработки исключительных ситуаций. Для этого в C# предназначена конструкция try...catch...finally.

```
try
{

}
catch
```

```

    {
    }
    finally
    {
    }
}

```

При использовании блока `try...catch..finally` вначале выполняются все инструкции в блоке `try`. Если в этом блоке не возникло исключений, то после его выполнения начинает выполняться блок `finally`. И затем, конструкция `try..catch..finally` завершает свою работу.

Если же в блоке `try` вдруг возникает исключение, то обычный порядок выполнения останавливается, и среда CLR начинает искать блок `catch`, который может обработать данное исключение. Если нужный блок `catch` найден, то он выполняется, и после его завершения выполняется блок `finally`.

Если нужный блок `catch` не найден, то при возникновении исключения программа аварийно завершает свое выполнение.

Пример обработки исключения:

```

class Program
{
    static void Main(string[] args)
    {
        int x = 5;
        int y = x / 0;
        Console.WriteLine($"Результат: {y}");
        Console.WriteLine("Конец программы");
        Console.Read();
    }
}

```

В данном случае происходит деление числа на 0, что приведет к генерации исключения.

В этом случае единственное, что нам остается, это завершить выполнение программы.

Чтобы избежать подобного аварийного завершения программы, следует использовать для обработки исключений конструкцию `try...catch...finally`.

```

class Program
{
    static void Main(string[] args)
    {
        try
        {
            int x = 5;
            int y = x / 0;
            Console.WriteLine($"Результат: {y}");
        }
    }
}

```

```

        catch
        {
            Console.WriteLine("Возникло
исключение!");
        }
        finally
        {
            Console.WriteLine("Блок finally");
        }
        Console.WriteLine("Конец программы");
        Console.Read();
    }
}

```

Таким образом, программа по-прежнему не будет выполнять деление на ноль и соответственно не будет выводить результат этого деления, но теперь она не будет аварийно завершаться, а исключение будет обрабатываться в блоке catch.

Следует отметить, что в этой конструкции обязателен блок try.

Ряд исключительных ситуаций может быть предвиден разработчиком. Например, при расчете периметра фигуры необходимо вводить только цифровые значения, поэтому при вводе любых других символов программа выдаст ошибку. С одной стороны, здесь как раз та ситуация, когда можно применить блок try..catch, чтобы обработать возможную ошибку. Однако, гораздо оптимальнее было бы проверить допустимость преобразования:

```

static void Main(string[] args)
{
    Console.WriteLine("Введите число");
    int x;
    string input = Console.ReadLine();
    if (Int32.TryParse(input, out x))
    {
        x *= x;
        Console.WriteLine("Квадрат числа: " + x);
    }
    else
    {
        Console.WriteLine("Некорректный ввод");
    }
    Console.Read();
}

```

При допустимости преобразования переменная x будет содержать введенное число. Так, не используя try...catch возможно обработать исключительную ситуацию.

С точки зрения производительности, использование блоков try..catch более ресурсозатратно, чем применение условных конструкций. Поэтому, по

возможности, вместо try..catch лучше использовать условные конструкции на проверку исключительных ситуаций.

5. Порядок выполнения работы

1. Выделить ключевые моменты задачи.
2. Построить алгоритм и теоретическую объектную модель решения задачи.
3. Запрограммировать полученные алгоритмы и объектную модель.

6. Форма отчета о работе

Лабораторная работа № _____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы и задания

1. Исключительная ситуация – это...
2. Какие блоки должны присутствовать для обработки исключительных ситуаций?
3. Обязательно ли использовать блок finally, если ранее был использован блок catch?

8. Рекомендуемая литература

1. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. СПб. : Изд-во Питер, 2021. 896 с.

2. Прайс, М. Дж. C# 10 и .NET 6. Современная кросс-платформенная разработка / М. Дж. Прайс. СПб : Изд-во Питер, 2023. 848 с.

3. Васильев, А.Н. Программирование на C# для начинающих. Особенности языка / А.Н. Васильев. М. : Эксмо, 2022. 528 с.

4. Фримен, А. ASP.NET Core 3 с примерами на C# для профессионалов / А. Фримен. СПб. : Изд-во Вильямс, 2021. 1184 с.