Лабораторная работа №13

Вариант 11

Выполнил: Олишкевич Игорь;

Тема: обработка исключительных ситуаций;

Цель:

Образовательная:

- изучить принципы обработки исключительных ситуаций в С#,

- сформировать умения обработки исключительной ситуации в С#.

Развивающая:

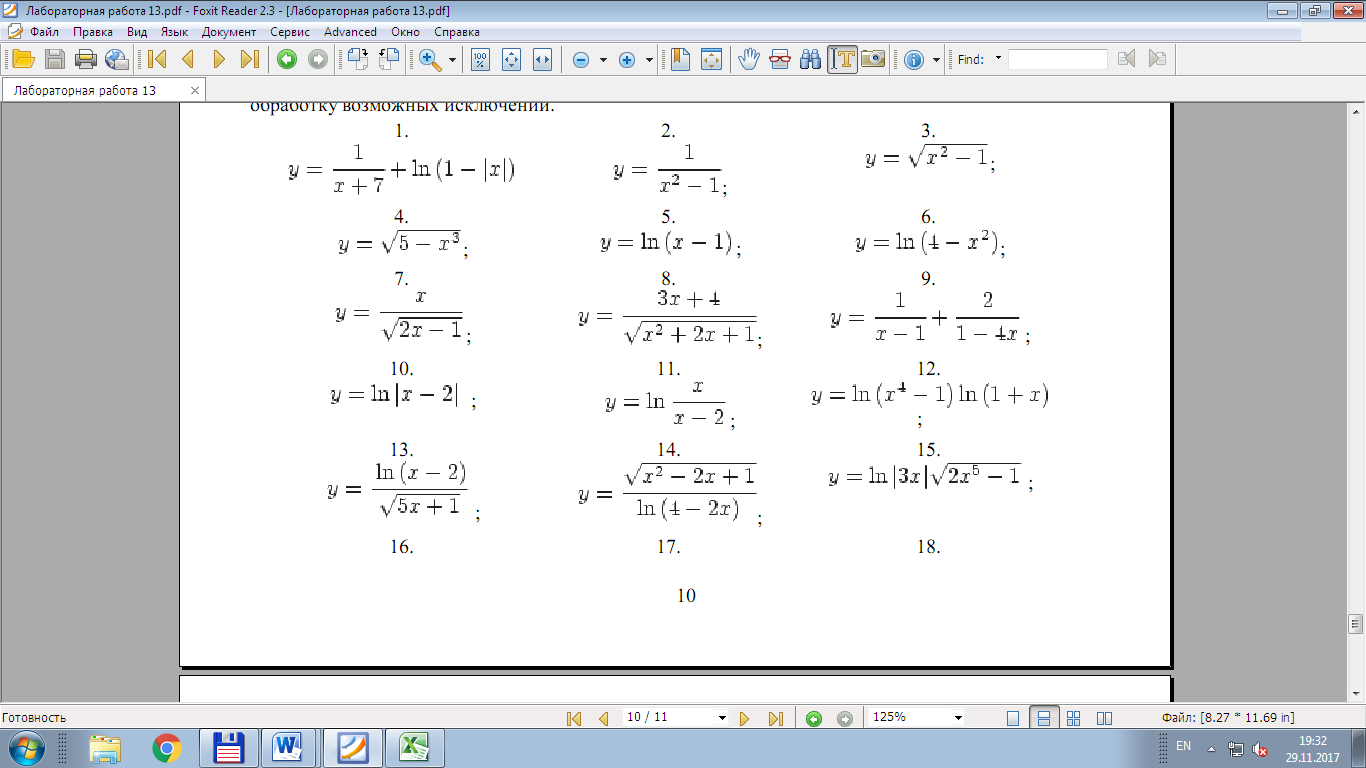
- создать условия для развития способности четко формулировать свои мысли.

Воспитательная:

- создать условия для воспитания сознательного и серьёзного отношения учащихся к учебной дисциплине.

Задание: построить иерархию классов в соответствии с вариантом задания: транспорт, экспресс, автомобиль, поезд. Расширить иерархию классов из с использованием виртуального класса в качестве основы иерархии. Показать пример использования полиморфизма методов.

**Задание 1:**



**Решение:**

using System;

namespace Hello

{

class Program

{

static double f(double x)

{

try

{

//если х не попадает в область определения, то генерируется исключение

if (x <= 2 && x >= 0) throw new Exception();

else

return Math.Log10(x / (x - 2));

}

catch

{

throw;

}

}

static void Main(string[] args)

{

try

{

Console.Write("Введите a: ");

double a = double.Parse(Console.ReadLine());

Console.Write("Введите b: ");

double b = double.Parse(Console.ReadLine());

Console.Write("Введите h: ");

double h = double.Parse(Console.ReadLine());

for (double i = a; i <= b; i += h)

try

{

Console.WriteLine("y({0})={1:f4}", i, f(i));

}

catch

{

Console.WriteLine("y({0})=error", i);

}

}

catch (FormatException)

{

Console.WriteLine("Неверный формат ввода данных");

}

catch

{

Console.WriteLine("Неизвестная ошибка");

}

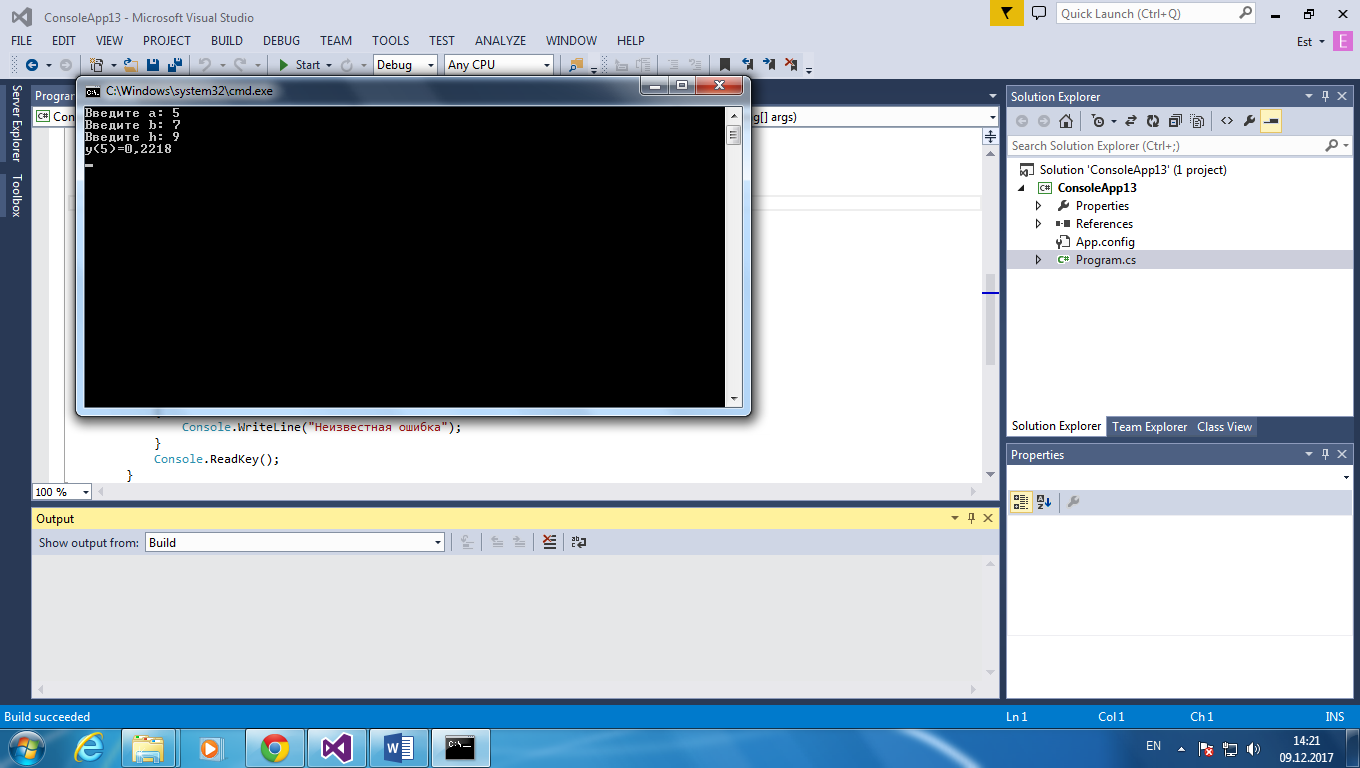
Console.ReadKey();

}

}

}

Тестирование:



Ответы на контрольные вопросы:

1. Для этого в C# предназначена конструкция try...catch...finally. При возникновении исключения среда CLR ищет блок catch, который может обработать данное исключение. Если такого блока не найдено, то пользователю отображается сообщение о необработанном исключении, а дальнейшее выполнение программы останавливается. И чтобы подобной остановки не произошло, и надо использовать блок try..catch.
2. Обрабо́тка исключи́тельных ситуа́ций (англ. exception handling) — механизм языков программирования, предназначенный для описания реакции программы на ошибки времени выполнения и другие возможные проблемы (исключения), которые могут возникнуть при выполнении программы и приводят к невозможности (бессмысленности) дальнейшей отработки программой её базового алгоритма. В русском языке также применяется более короткая форма термина: «обработка исключений».

3-4) Функции обработки исключений в языке C# помогают вам справиться с непредвиденными или исключительными проблемами, которые возникают при выполнении программы. Обработка исключений использует ключевые слова try, catch и finally для действий, которые могут оказаться неудачными. Это позволяет обрабатывать ошибки так, как кажется разумным, а также правильно высвобождать ресурсы. Исключения могут создаваться средой выполнения (CLR), платформой .NET Framework, библиотеками сторонних поставщиков или кодом самого приложения. Чтобы создать исключение, используйте ключевое слово throw.

Во многих случаях исключение может создаваться не тем методом, который вызывается в вашем коде, а одним из последующих методов в стеке вызовов. В этом случае среда CLR разворачивает весь стек в поисках метода с блоком catch для исключений соответствующего типа, и выполняет первый обнаруженный блок catch подходящего вида. Если подходящий блок catch не будет обнаружен во всем стеке вызовов, среда CLR завершает процесс и выводит сообщение для пользователя. В этом примере метод выполняет проверку деления на нуль и перехватывает ошибку. Если не использовать обработку исключений, такая программа завершит работу с ошибкой DivideByZeroException was unhandled (Исключение DivideByZero не обработано).

5) Программные инструкции, которые нужно проконтролировать на предмет исключений, помещаются в блок try. Если исключение возникает в этом блоке, оно дает знать о себе выбросом определенного рода информации. Выброшенная информация может быть перехвачена и обработана соответствующим образом с помощью блока catch. Любой код, который должен быть обязательно выполнен при выходе из блока try, помещается в блок finally. Рассмотрим пример, демонстрирующий, как отследить и перехватить исключение.