Отчет по лабораторной работе № 8

“Исследование средств управления потоками ввода-вывода. Исследование механизма обработки исключений”

по дисциплине

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Выполнил студент группы ИВТ/б-22о

Горбенко К.Н.

Проверил:

Тимофеев И.С.

1. **Цель работы:** изучить способы реализации и особенности управления потоками ввода/вывода, исследовать способы генерации и обработки исключений.
2. **Задание на работу:**
   1. Написать программу преобразования температуры в градусах Цельсия (например, 15С) в температуру в градусах по Фаренгейту (например, 59F), и наоборот. 0 градусов по Цельсию соответствует 32 градусам по Фаренгейту. Изменение температуры на 1 градус по Цельсию соответствует изменению на 1.8 градуса по Фаренгейту. Результаты вычислений поместить в файл. Установить ширину поля 13 символов, точность - 4 цифры, заменить пробелы символом “\” с помощью функций манипуляторов.
   2. Найти значение математического выражения, описав перехват ошибок вычислений: **y = 10 arcsin(10x+2.2)**.
3. **Программа на языке С#.** 
   1. **Программа к первому заданию:  
       Класс TemperatureConversions:**

|  |
| --- |
| public static class TemperatureConversions  {  public static double ToFahrenheit(double celsius)  {  return (celsius - 9.0 / 5) + 32;  }   public static double ToCelsius(double fahrenheit)  {  return (fahrenheit - 32) \* 5.0 / 9;  }  } |

**Класс Program:**

|  |
| --- |
| class Program  {  static void Main(string[] args)  {   int userOption = -1;   do  {  try  {  userOption = GetUserOption();   switch (userOption)  {  case 1:  var celsius = ToCelsius(GetDegrees());  WriteConversionToFile(celsius);  break;   case 2:  var fahrenheit = ToFahrenheit(GetDegrees());  WriteConversionToFile(fahrenheit);  break;   case 3:  return;   default:  WriteLine("Wrong input, enter any key to try again");  ReadKey();  break;  }  }  catch (Exception e)  {  WriteLine(e.Message);  continue;  }  }  while (userOption != 3);  }   static int GetUserOption()  {  int? userOption = null;   while (userOption == null)  {  Clear();  WriteLine  ("What do you want to do?\n" +  "1 - Translate fahrenheit to celsium\n" +  "2 - Translate celsium to fahrenheit\n" +  "3 - Exit");   userOption = int.Parse(ReadLine());  }   return (int)userOption;  }   static double GetDegrees()  {  double? degrees = null;   while (degrees == null)  {  Clear();  WriteLine("Enter degrees\n");  degrees = double.Parse(ReadLine());  }   return (double)degrees;  }   static void WriteConversionToFile(double degrees)  {  var line = new string('\\', 13).Insert(0, $"{degrees:0.0000}");   using (var writer = new StreamWriter(path:"File.txt", append:true))  {  writer.Write(line);  }   WriteLine("The result has been written to a file. Enter any key to continue");  ReadKey();  }  } |

* 1. **Программа ко второму заданию:**

|  |
| --- |
| class Program:  {  static void Main(string[] args)  {  while (true)  {  try  {  Clear();  WriteLine("Enter x or type \"\*\" to exit");   var line = ReadLine();  if (line.Contains("\*")) return;   var x = double.Parse(line);  var y = 10 \* Asin(10 \* x + 2.2);   if (double.IsNaN(y) || double.IsInfinity(y)) throw new Exception("Asin() parameter should be from range [-1; 1]");   WriteLine($"y={y}");  ReadKey();  }  catch (Exception e)  {  WriteLine("An exception has occured: " + e.Message);  WriteLine("Enter any key to continue...");  ReadKey();  }  }  }  } |

1. **Описание программы.**

Для реализации конвертаций температуры был создан статический класс TemperatureConvertions, который содержит методы для конвертаций. Для манипулирования потоками создавался экземпляр строки необходимой длины (13 символов), который затем изменялся с помощью стандартных методов класса string. Для реализации обработки исключений метод Main содержит блоки try и catch.

Для решения задачи № 2 был использован статический класс Math, содержащий методы, описывающие математические операции. При вводе неправильного значения переменной x пользователем, программа должна бросить и поймать исключение с соответствующим сообщением.

1. **Вывод.**

В ходе лабораторной работы были изучены потоки ввода-вывода и механизм обработки исключений. В программах, написанных в ходе работы, не могут случиться сбои в работе, так как все ошибки перехватываются обработчиком исключений. Для вывода в файл использовался класс StreamWriter из пространства имен System.IO.