#7 Исключения.

Задание:

1. Изучите работу с исключениями и класс **Exception**, ознакомьтесь с базовыми свойствами класса:
   https://docs.microsoft.com/en-us/dotnet/api/system.exception?view=netframework-4.8
   https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/exceptions/
2. Создайте иерархию своих исключений, унаследовав их от каких-либо базовых: **Exception**, **DivideByZeroException** и т.д. (3 типа и более)
3. Используя предыдущую лабораторную работу (#6) смоделируйте выброс исключения и продемонстрируйте его отлов и обработку:
   a. Используя классический вид **try-catch** и свои пользовательские исключения;
   b. Используя классический вид **try-catch-finally**;
   c. Продемонстрируйте возможность многоразовой обработки одного исключения и проброс его выше по стеку вызовов;
   d. Использование фильтров исключений (доступно начиная с *C#6*)
4. Ознакомьтесь с классами **Debug** и **Debugger**:
   https://docs.microsoft.com/en-us/dotnet/api/system.diagnostics.debug?view=netframework-4.8
   https://docs.microsoft.com/en-us/dotnet/api/system.diagnostics.debugger?view=netframework-4.8
5. Продемонстрируйте использование метода **Assert**.
6. Изучите возможности отладки приложения с помощью точек останова Visual Studio. Ознакомьтесь с *условными точками останова*.

Повышенное задание:

1. Создайте интерфейс **ILogger** с 2 перегрузками метода **Log**:
   a. Первая перегрузка:
      i. Принимает параметром тип исключения (**Exception**);
   b. Вторая перегрузка:
      i. Принимает параметрами:
         1. Сообщение для лога
         2. Тип сообщения (**Error**, **Warning**, **Information**).
2. Используйте перечисление для указания типа сообщения.

3. Создайте 2 реализации интерфейса: **FileLogger** и **ConsoleLogger**. **FileLogger** записывает сообщения лога в файл, **ConsoleLogger** – выводит их на консоль.
4. Перегрузка, которая принимает один параметр **Exception**, автоматически извлекает сообщение об ошибке и осуществляет логгирование с типом сообщения **Error**.
5. Реализуйте **FileLogger** и **ConsoleLogger** как «*классы-одиночки*».
6. **FileLogger** должен предоставлять возможность указания имени файла для лога. Данный логгер создает файл, если его нет и дозаписывает сообщения в файл, если таковой существует.
7. **ConsoleLogger** должен выводить сообщения лога на консоль соответствующим цветом:
    a. **Error** – красный
    b. **Warning** – желтый
    c. **Info** – зеленый/ярко-синий/голубой
8. Возможность указания параметров можно предусмотреть как с помощью свойств, так и с помощью параметров конструкторов классов.
9. Логгеры должны записывать сообщения в виде:
   время, тип_записи_лога: дополнительное сообщение.
   27.10.2019 02:36, INFO: Test log message.

Вопросы:

1. Что такое исключение? В чем заключается методика обработки исключения?
2. Какие ключевые слова используются при обработке исключений? Расскажите механизм обработки.
3. Обязательно ли нам указывать тип исключения в блоке **catch**?
4. Как работает блок **finally**?
5. Расскажите механизм многократной обработки одного исключения.
6. Что такое фильтры исключений?
7. Какой синтаксис позволяет отловить любое исключение в C#?