

## #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:
  - а. Используя классический вид **try-catch** и свои пользовательские исключения;
  - б. Используя классический вид **try-catch-finally**;
  - в. Продемонстрируйте возможность многократной обработки одного исключения и проброс его выше по стеку вызовов;
  - г. 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 класс **Logger**, который будет заниматься логгированием различных событий и исключений. Логгер должен уметь логгировать ошибки/исключения, предупреждения и просто какую-то информацию. **Note**: используйте перечисление.
2. Логгер должен записывать лог в виде:  
*время, тип\_записи\_лога: дополнительное сообщение.*  
*27.10.2019 02:36, INFO: Test log message.*
3. Создайте 2 реализации логгера: **FileLogger** и **ConsoleLogger**. **FileLogger** будет записывать сообщения лога в файл, добавляя записи к уже существующим. **ConsoleLogger** – выводить сообщения на консоль.

4. Добавьте в классы из л.р. #6 логгер так, чтобы его возможно было быстро заменить во время выполнения другим и вместо простого вывода на консоль сообщения об ошибке, используйте свой логгер.

### 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#?