**Module\_5\_Exception\_Handling**

1) What is the purpose of Exception Handling in C#

Исключения и обработка исключений **помогают справляться с непредвиденными или исключительными ситуациями, возникающими во время работы программы** .

Базовым для всех типов исключений является тип **Exception**. Этот тип определяет ряд свойств, с помощью которых можно получить информацию об исключении.

**InnerException**: Программа может перехватить исключение и повторно вызвать другое исключение, передав исходное пойманное исключение как InnerException.

* **Message**: хранит сообщение об исключении, текст ошибки
* **Source**: хранит имя объекта или сборки, которое вызвало исключение
* **StackTrace**: возвращает строковое представление стека вызывов, которые привели к возникновению исключения
* **TargetSite**: возвращает метод, в котором и было вызвано исключение

Однако так как тип Exception является базовым типом для всех исключений, то выражение catch (Exception ex) будет обрабатывать все исключения, которые могут возникнуть.

Но также есть более специализированные типы исключений, которые предназначены для обработки каких-то определенных видов исключений. Их довольно много, я приведу лишь некоторые: (**predefined exception types)**

* **DivideByZeroException**: представляет исключение, которое генерируется при делении на ноль
* **ArgumentOutOfRangeException**: генерируется, если значение аргумента находится вне диапазона допустимых значений
* **ArgumentException**: генерируется, если в метод для параметра передается некорректное значение
* **IndexOutOfRangeException**: генерируется, если индекс элемента массива или коллекции находится вне диапазона допустимых значений
* **InvalidCastException**: генерируется при попытке произвести недопустимые преобразования типов
* **NullReferenceException**: генерируется при попытке обращения к объекту, который равен null (то есть по сути неопределен)

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.

2) Can a try block have multiple catch blocks?

Yes

3) Describe a flow how exceptions are handled?

4) What is the base class from which all exceptions are derived?

5) What is the difference between Exception and Inner Exception?

* **InnerException**: хранит информацию об исключении, которое послужило причиной текущего исключения

Программа может перехватить исключение и повторно вызвать другое исключение, передав исходное пойманное исключение как InnerException.

6) What is the difference between throw ex; and throw; statements?

Throw ex => потеряли информацию о том, где произошло это исключение.

Когда возникает исключение, часть информации, которую оно несет, представляет собой трассировку стека. Трассировка стека — это список иерархии вызовов методов, который начинается с метода, создающего исключение, и заканчивается методом, перехватывающим исключение. Если исключение создается повторно путем указания исключения в throw.ex операторе, трассировка стека перезапускается в текущем методе, и список вызовов метода между исходным методом, вызвавшим исключение, и текущим методом теряется. Чтобы сохранить исходную информацию о трассировке стека с исключением, используйте throwоператор без указания исключения.

Diagram

Description automatically generated

7) What is the purpose of finally statement?

С помощью **блока finally** можно очистить все ресурсы, выделенные в **блоке try, и запустить код, даже если в блоке** try возникнет исключение

8) What predefined .NET Exceptions do you know?

9) Is there a way to create a custom exception?

Однако в некоторых случаях, когда ни одно из них не соответствует вашим потребностям, вы можете создать свои собственные исключения.