**Ответы на лабораторную работу №6**

1. Генерируется оператором throw
2. В блоке try мы записываем выражение, которое проверяется на исключение, если его нет, то CLR никогда не переходит к выполнению операций в блоке catch, если же в блоке try порождается исключение, то оно обрабатывается в блоке catch и, если есть блок finally, то он выполняется после обработки исключения.
3. Throw
4. Try, catch, throw, finally.

Механизм обработки:

1. Если исключение не произошло:

* try выполняем до конца
* catch пропускаем
* finally выполняем

1. Исключение произошло:

* выполнение try прекращается (все что идет за возникшим исключение игнорируется)
* ищем блок catch на соответствие по типу исключения

1. Если блок catch не найден:

* разматывает стек, локальные объекты, выходят из области видимости
* снова генерируется исключение в точке вызова метода
* если блока не найдено, то сообщение - необработанное исключении⇒ дальнейшее выполнение программы останавливается

1. Если блок catch найден:

* Передается управление ближайшему catch-обработчику, совместимому с типом выброшенного исключения
* объект-исключения передается, если это предусмотрено, обработчику в качестве параметра
* переходим/ищем finally, если finally нет, то выполнение программы продолжается начиная с позиции, след. за последним обработчиком данного блока try

1. Если есть finally, то исполнится блок в finally, если нет и его, то программа не запустится из-за ошибки.
2. Фильтр исключения позволяет указать дополнительные условия, при которых используется обработчик исключения. Эти условия принимают форму булева выражения, перед которым ставится ключевое слово when.
3. Блоки try могут быть вложенные
4. Блоки try и catch
5. Наиболее конкретные обработчики должны находиться в начале списка, т.к. CLR поиск подходящего идёт сверху вниз
6. It’s OK.
7. Ключевое слово throw;
8. GetBaseException(): При переопределении в производном классе возвращает исключение [Exception](https://learn.microsoft.com/ru-ru/dotnet/api/system.exception?view=net-5.0), которое является первопричиной одного или нескольких последующих исключений.

GetObjectData: заполняет объект [SerializationInfo](https://learn.microsoft.com/ru-ru/dotnet/api/system.runtime.serialization.serializationinfo?view=netcore-3.0) данными, необходимыми для сериализации целевого объекта.