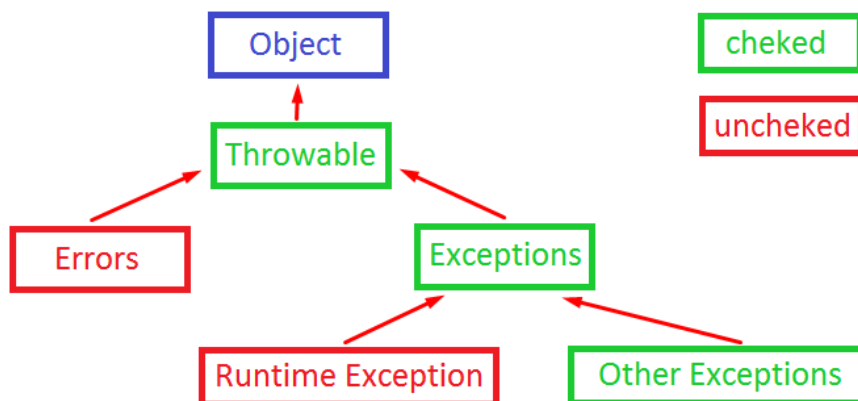


Вопросы по теме Описание и иерархия исключений:

1. Что такое исключение?
2. Опишите иерархию исключений.
3. Что вы знаете об обрабатываемых и не обрабатываемых (checked/unchecked) исключениях?
4. В чем особенность RuntimeException?

Исключения.

Исключение (ошибка) – непредвиденная ситуация, возникшая во время выполнения или компилирования программы.



Все исключения являются объектами. Каждый вид исключения имеет свой класс. Т.е. вся иерархия исключений – это цепочка наследования одних классов от других. На вершине наследования для любого класса в Java стоит класс Object.

Класс Throwable является суперклассом для всех ошибок и исключений в Java. Этот класс записывает отчет о текущем состоянии стека во время выполнения программы (stack trace), содержит текстовое сообщение об ошибке и его возможную причину (исключение более низкого уровня). Проверяемый тип.

Класс Error наследуется от класса Throwable. Ошибки класса Error (+ наследники) предназначены для сигнализации о серьезных низкоуровневых ошибках JVM или ошибках загрузки классов (например, переполнение памяти или стека, JVM не может найти нужный класс и т.д.), после которых дальнейшая работа программы обычно невозможна. Такие ошибки как правило не обрабатываются. Они предназначены для скорейшего аварийного завершения работы. Непроверяемый тип.

Класс Runtime Exception наследуется от класса Exception. Ошибки класса Runtime Exception (+ наследники) предназначены для сигнализации об ошибках в логике программы, необнаруженных на этапе компиляции. Обычно при таких ошибках дальнейшее выполнение программы не имеет смысла (например, ошибка доступа к базе данных). Непроверяемый тип.

Класс Exception (все исключения, наследуемые от этого класса, кроме Runtime Exception) наследуется от класса Throwable. Ошибки этого класса являются неотъемлемой частью работы программы, их возникновение предсказуемо. Такие ошибки программист обязан обработать соответствующим образом (например, деление на 0). Проверяемый тип.

Проверяемый тип исключений – это ошибки, возможность возникновения которых JVM проверяет во время компиляции и, если такая возможность имеется, то не дает скомпилировать код, пока данные исключения не будут обработаны соответствующим образом. Возможность возникновения таких ошибок программист не может исключить из программы, но может задать алгоритм действий при их возникновении. Возникновение проверяемого исключения не всегда означает некорректную работу программы. Иногда такие ошибки – это часть нормальной работы.

Обработка проверяемых исключений:

1. если в методе может возникнуть проверяемое исключение, то в своей сигнатуре такой метод должен содержать тип возможного исключения

```
void myMethod() throws Exception { }
```

2. если вызывается метод, в сигнатуре которого указано исключение, то это исключение обязано быть обработано:
 - добавить исключение в сигнатуру своего метода с помощью throws;
 - обработать исключение.

Непроверяемый тип исключений – это ошибки, возникновение которых невозможно предсказать (с точки зрения JVM), поэтому JVM не проверяет возможность их возникновения. Такие ошибки программист сам должен предсказывать и писать программу исходя из их возможного возникновения (например, если в параметр метода вместо объекта передается null, нужно предвидеть такую возможность и создать соответствующую проверку).

Создание собственных исключений.

Для создания собственного исключения необходимо свой класс-исключение унаследовать от уже существующего исключения.

```
class MyException extends Exception { }
```