

Проверка ошибок

Проверка значений переменных

Проверка возвращаемого значения метода

Использование дополнительных переменных
(флаги)

Как проверить создание объекта?

Исключения

Исключительной называется ситуация, из-за которой нормальное продолжение работы метода или даже всей программы невозможно. При этом обработка ее как обычной ошибки невозможна. В таком случае генерируются специальные объекты-исключения.

Исключения позволяют:

- разобраться с проблемой и восстановить работоспособность программы,

- остановить выполнение и сообщить о возникших трудностях.

Схема генерации и обработки исключений

Генерация `throw`

Перехват `try`

Обработка `catch`

Генерация исключений

Создается объект специального класса (new).

Работа программы прерывается, объект «выбрасывается».

Когда происходит:

- автоматически
- вручную (throws)

Автоматическая генерация исключений

```
int d = 0;  
int a = 42 / d;
```

Результат:

Exception in thread "main" java.lang.ArithmeticException:
/ by zero

at exceptionexample. ExceptionExample.main
(ExceptionExample.java:19)

Явная генерация исключения

```
throw new Exception();
```

```
throw new Exception(«Something happened»);
```

```
throw new NullPointerException();
```

Блок-перехватчик

```
try {
```

```
// код, где могли произойти исключения
```

```
// и весь код, зависящий от этих действий
```

```
}
```

Блоки-ловушки

```
try {...}
```

```
catch( Type1 id1)
```

```
{тело обработчика исключения}
```

```
catch( Type2 id2)
```

```
{тело обработчика исключения}
```

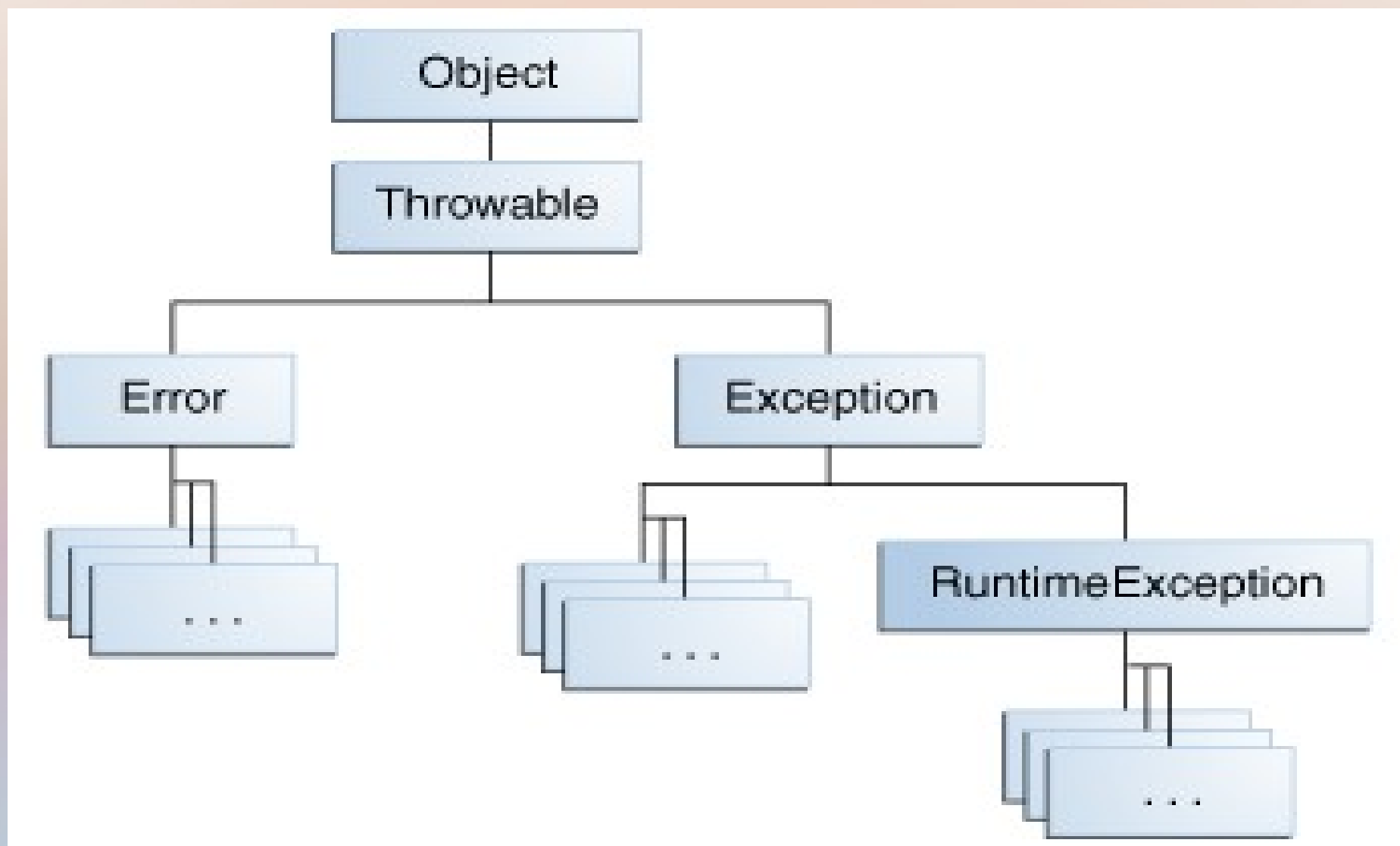

Блок finally

```
try{ }  
catch( Type1 id1) {}  
catch( Type2 id2) {}  
finally {операторы }
```

Классы исключений

- Обязательно наследник класса Exception
- Стандартные классы (библиотечные)
- Пользовательские исключения

Иерархия исключений



Типы исключений

- Неконтролируемые – не обрабатываются (Errors)
- Контролируемые – обязательно обрабатываются (все наследники Exception кроме RuntimeException)
- Наследники RuntimeException – их появление свидетельствует об ошибке в программе, которую надо устранить

Контролируемые исключения

Обязательная обработка

- в методе с помощью блоков `try – catch`
- за пределами метода – в заголовке метода указываются генерируемые исключения

```
void method() throws IOException {  
    throw new IOException();  
}
```

Стандартные исключения

IOException (ввод-вывод)

FileNotFoundException

FileSystemException

SQLException (ошибки доступа к базе данных)

TimeoutException (превышен лимит времени)

Исключения RuntimeException

ArithmeticalException – Возникла исключительная ситуация, связанная с ошибкой при выполнении арифметического вычисления

IndexOutOfBoundsException – Задано значение индекса массива или содержимого строки типа **String**, не принадлежащее допустимому диапазону.

IllegalArgumentException – Методу передано неверное значение аргумента

NullPointerException – Для доступа к полю или методу была применена ссылка, равная **null**.

NumberFormatException extends **IllegalArgumentException** – Строка, которая, как предполагалось должна содержать представление числа, не отвечает этому требованию.

Неконтролируемые исключения (Error)

ThreadDeath — вызывается при неожиданной остановке потока посредством метода `Thread.stop()`.

StackOverflowError — ошибка переполнение стека. Часто возникает в рекурсивных функциях из-за неправильного условия выхода.

OutOfMemoryError — ошибка переполнения памяти.

Методы исключений

Конструктор с параметром String

Exception(String message)

Получение строки с сообщением

public String getMessage()