Курс Java-разработчик

ООП

Нештатные ситуации

- Нештатная ситуация – это ситуация, в которой выполнение некоторого фрагмента кода программы оказывается по тем или иным причинам невозможно.

Исключительные и ошибочные ситуации

- Исключительная ситуация – это нештатная ситуация, возникшая в силу внешних по отношению к программе причин.

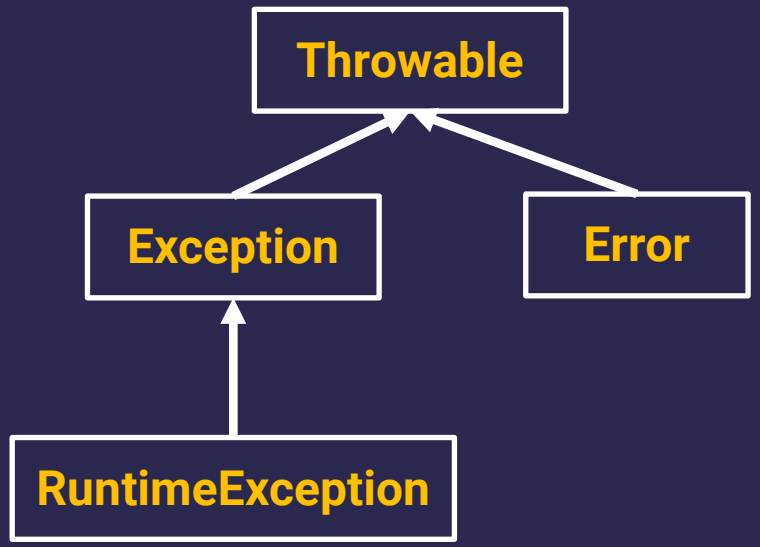
Примеры: не открылся файл, произошло незапланированное закрытие сетевого соединения.

- Ошибочная ситуация – это нештатная ситуация, возникшая из-за ошибки в коде программы.

Примеры: выход за границы массива; деление на ноль; переполнение стека.

Исключения

Исключение – это объект, описывающий нештатную ситуацию.



Иерархия исключений в Java

Throwable – базовый класс для всех классов исключений.
Error – базовый класс для классов исключений, описывающих «смертельные» для программы системные ошибочные ситуации
Exception – базовый класс для всех классов несистемных исключений.
RuntimeException – базовый класс для классов несистемных исключений, описывающих ошибочные ситуации.

Конструкторы класса **Throwable**

Основные конструкторы класса **Throwable**:

- Throwable(String message)
- Throwable(String message, Throwable cause)

Оба конструктора принимают в качестве параметра сообщение message, описывающее нештатную ситуацию.
Второй конструктор, кроме того, позволяет организовать так называемую «цепочку исключений», когда исключение (cause) – вложено в другое исключение, то есть подразумевается, что вложенное исключение является причиной объемлющего исключения.

Методы класса **Throwable**

- Throwable getCause ();
- String getMessage ();
- String toString ();
- void printStackTrace ();

Создание собственного исключения

Отметим, что в качестве базового класса для создаваемого исключения можно выбрать **Throwable**, **Exception**, **RuntimeException**, **Error** или любой другой класс – подкласс класса **Throwable**.
Тем не менее, наиболее типично наследовать собственные классы исключений от **Exception** или **RuntimeException** в зависимости от того, какой тип нештатных ситуаций описывается исключением.

```
1  package com.example;
2
3  public class NegativeValueException extends Exception {
4
5      public NegativeValueException(String message) {
6          super(message);
7      }
8
9  }
```

Перехват нештатных ситуаций

- Перехват нештатных ситуаций – это механизм, обеспечивающий продолжение работы программы при возникновении нештатной ситуации.

В случае возникновения нештатной ситуации генерируется исключение, описывающее нештатную ситуацию. Генерация исключения приводит к передаче управления на фрагмент кода программы, называемый обработчиком исключения.

Операторы перехвата исключений

```
try {
    /* код, в котором может возникнуть
    нештатная ситуация */
} catch (SomeException ex) {
    /* обработчик исключений, который
    можно привести к типу SomeException */
} catch (SomeException2 ex) {
    /* ... */
} finally {
    /* код , который должен вызываться
    при любом выходе из try – блока */
}
```

Порождение исключения

Порождение исключения выполняется оператором

- throw исключение;

Например,

- throw new SomeException ();

Если класс исключения не является подклассом классов **Error** и **RuntimeException**, то для любого порождённого исключения в коде программы должен быть предусмотрен обработчик.
Например, объявление метода должно выглядеть так:

```
void function() throws SomeException1, SomeException2, ... {
    /* code here */
}
```

Домашнее задание

- Добавить пакет исключений для зоопарка (com.zoo.exception), создать два типа (класса) проверяемых исключений.
- Для некоторых методов "бросить" исключения при определенных условиях, а в главном методе перехватывать данные исключения.