

# 12. OOP. JAVA

---

Java Exception Handling

# Exceptions

- The Java Tutorials: Exceptions  
<https://docs.oracle.com/javase/tutorial/essential/exceptions>
- Исключения в Java  
<http://www.quizful.net/post/java-exceptions>
- Исключения в Java, Часть I (try-catch-finally)  
<http://habrahabr.ru/company/golovachcourses/blog/223821>
- Исключения в Java, Часть II (checked/unchecked)  
<http://habrahabr.ru/company/golovachcourses/blog/225585>

# Java Exceptions

- Исключение – это проблема (ошибка) возникающая во время выполнения программы.
- Исключение в Java – это объект, который описывает аварийную ситуацию, произошедшую в некоторой части кода программы.

# Java Exception Handling

Обработка исключительных ситуаций – механизм языков программирования, предназначенный для описания реакции программы на ошибки времени выполнения и другие возможные проблемы (исключения), которые могут возникнуть при выполнении программы и приводят к невозможности (бессмысленности) дальнейшего исполнения базового алгоритма.

# Иерархия исключений

Все типы исключений являются прямыми наследниками класса **Throwable**. От него унаследовано 2 класса: **Error** и **Exception**.

**Throwable** – суперкласс всех исключений; **Error** – предок исключений JVM (их можно, но не следует обрабатывать); **Exception** – предок всех программных исключений; **RuntimeException** – предок исключений, которые не обязательно обрабатывать.

# Иерархия исключений

В Java определены проверяемые (**CHECKED**) и непроверяемые (**UNCHECKED**) исключения.

- **CHECKED** исключения, это те, которые ОБЯЗАТЕЛЬНО должны обрабатываться блоком **catch** или описываться в сигнатуре метода с помощью **throws**.
- **UNCHECKED** могут не обрабатываться и не быть описанными.

# Checked и Unchecked

Все **Throwable**, кроме **Error** и **RuntimeException** – проверяемые (**Checked**) исключения.

**Unchecked** исключения – наследники **RuntimeException** и **Error**.

**Checked** исключения отличаются от **Unchecked** исключений в Java тем, что:

- 1) Наличие/обработка **Checked** исключения проверяются на этапе компиляции (compile-time checking).  
Наличие/обработка **Unchecked** исключения происходит на этапе выполнения (runtime checking).
- 2) **Checked** исключения должны описываться в сигнатуре метода через **throws** или обрабатываться блоком **try/catch**.

# try-with-resources

## Java SE7

- **try** блок, декларирующий один или более ресурсов, которые будут автоматически закрыты при выходе из блока.
- Ресурсы – объекты классов, которые реализуют `java.lang.AutoCloseable` (JSE7).