



Exceptions



Определение

Исключения в Java - механизм для обработки ошибок и других исключительных ситуаций, возникающих во время выполнения программы.



Ключевые слова

```
package l42.slides.ex1;

public class Main {
    public static void main(String[] args) {
        try {
            // Этот код может вызвать
            // ArithmeticException из-за деления на ноль
            int result = 10 / 0;
        } catch (ArithmeticException e) {
            // Этот код выполнится когда
            // возникнет исключение ArithmeticException
            System.out.println("Произошла ариф. ошибка." );
        } finally {
            // Этот код выполнится всегда
            System.out.println("Блок finally выполнен." );
        }
    }

    // обязуем обработать исключение
    // ArithmeticException при вызове метода.
    // если throws не указан - обработка необязательна
    public void checkAge(int age) throws ArithmeticException {
        if (age < 18) {
            throw new ArithmeticException( "Доступ запрещен" );
        } else {
            System.out.println("Доступ разрешен." );
        }
    }
}
```

Новые ключевые слова:

- try: блок кода, в котором может произойти исключение.
- catch: блок кода, который обрабатывает исключение, возникшее в блоке try.
- finally: блок кода, который выполняется после try/catch независимо от того, было ли исключение.
- throw: оператор для явного выброса исключения.
- throws: используется в сигнатуре метода для указания, что метод может выбросить исключение.



Создание собственных исключений

```
package 142.slides.ex3;

class MyCheckedException extends Exception {
    public MyCheckedException(String message) {
        super(message);
    }
}

class MyUncheckedException extends RuntimeException {
    public MyUncheckedException(String message) {
        super(message);
    }
}

public class Main {
    public void myMethod() throws MyCheckedException {
        throw new MyCheckedException("Произошла ошибка");
    }

    public void myMethod2() {
        throw new MyUncheckedException("Произошла ошибка");
    }
}
```

Для создания собственного исключения необходимо наследовать класс от Exception или RuntimeException.



Рекомендации

- Не подавляйте исключения: всегда обрабатывайте исключения адекватно.
- Используйте конкретные типы исключений: ловите и обрабатывайте наиболее специфические исключения (вместо общих родителей).
- Не используйте исключения для управления потоком выполнения программы: исключения предназначены только для обработки ошибок.
- Включайте полезную информацию в исключения: когда вы выбрасываете исключение, добавляйте полезное сообщение и контекст.



try-with-resources

```
package 142.slides.ex2;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class Main {
    public static void main(String[] args) {
        try (
            BufferedReader br = new BufferedReader(
                new FileReader("path/to/file.txt")
            )
        ) {
            // Чтение файла
            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            // Обработка исключения
            System.out.print("Не получилось считать файл");
            e.printStackTrace();
        }
    }
}
```

try-with-resources - специальная форма блока try, которая гарантирует закрытие ресурсов после их использования.

Ресурс должен реализовать интерфейс `AutoCloseable` или `Closeable`.



Контрольная точка

- Понятно ли?

Если все ясно, ставим плюсы, иначе - задаем вопросы.



Домашнее задание

- Ознакомиться с документацией и статьей по ссылкам со второго слайда
 - <https://docs.oracle.com/javase/tutorial/essential/exceptions/index.html>
- Кратко
 - <https://metanit.com/java/tutorial/4.1.php>
 - <https://metanit.com/java/tutorial/4.2.php>
 - <https://metanit.com/java/tutorial/4.3.php>
- Задачи
 - Минимум: https://www.w3schools.com/java/java_try_catch.asp (и прочитать раздел “кратко” выше если не делаете план максимум)
 - Максимум:
 - <https://pastebin.com/PBVEgpCE>
 - вникнуть в код
 - уточнить типы отлавливаемых исключений



The end