

**XML**

# DOM

Модель представления документа

XML - набор узлов (тегов)

**тег** `<bean/>` или `<bean></bean>`

Узел может содержать вложенные узлы

Node - представление узла в DOM

# DOM - дерево элементов

один корневой элемент root

<!-- Это комментарий. -->

Атрибуты тега (пары ключ=значение)

<bean id="10" class="Main" />

# DOM парсер

- считать весь документ
- пройти все узлы
- построить его дерево

# SAX

Не строит дерево, а проходит по документам последовательно

События:

- `startDocument()`
- `startElement()`
- `endElement()`
- `endDocument()`

# Reflection API

`java.lang.reflect`

- класс объекта
- конструкторы, методы, поля
- модификаторы доступа
- создать экземпляр
- вызвать метод
- установить поле (даже `private`)

# Class

// есть объект

```
MyClass my = new MyClass();  
Class clazz = my.getClass();
```

// есть тип

```
Class clazz = MyClass.getClass();
```

// есть имя

```
Class clazz = Class.forName("java.util.HashMap");
```

getFields(); protected, public

getDeclaredFields(); + private

getDeclared<Fields|Methods|Annotation>



# Аннотации

Аннотации - метаданные программы

@Override

@Deprecated

@SuppressWarnings

@Test

# Синтаксис

```
@Target(value=ElementType.FIELD)
@Retention(value= RetentionPolicy.RUNTIME)
public @interface Name {
    String name();
    String type() default "String";
}
```

=====

```
@Name (name="Ivan")
Person person;
```

# @Target

FIELD

CLASS

METHOD

# @Retention

Когда видна метайнформация

- `RUNTIME` - во время исполнения через `Reflection API`
- `SOURCE` - только в исходном коде
- `CLASS` - присутствует после компиляции, но не доступна через `Reflection`

