

Аннотации в языке Java.

Теоретическая часть.

Аннотации (*Annotation*) в языке Java позволяют использовать метаданные непосредственно в Java коде. Аннотации представляют информацию о программе, которая не относится непосредственно к логике программы. Аннотации не имеют непосредственного влияния на код к которому они относятся.

Несколько областей применения аннотаций:

1. **Информация для компилятора** — аннотации могут использоваться компилятором для определения ошибок и подавления сообщений с подтверждениями
2. **Обработка на этапе разработки и компиляции** — инструменты разработки могут анализировать аннотации и на их основе генерировать информацию (например XML)
3. **Обработка на этапе выполнения** — некоторые аннотации могут быть доступны на этапе компиляции

Аннотации могут быть применены к определению классов, полей, методов и других лексем программы.

Примеры применения аннотаций

1. Аннотация с элементами (elements)

a) Несколько элементов со значениями

```
@Author(  
    name = "Benjamin Franklin",  
    date = "3/27/2003"  
)  
class MyClass() { }
```

b) Единственный элемент

```
@SuppressWarnings(value = "unchecked")  
void myMethod() { }
```

2. Если Аннотация содержит только один элемент, имя элемента может быть опущено

```
@SuppressWarnings("unchecked")  
void myMethod() { }
```

3. Если элементы отсутствуют, могут быть опущены и скобки

```
@Override
void mySuperMethod() { }
```

Применение аннотаций при документировании кода

Заголовок, обычно применяющийся, в компании Ч.

```
public class Generation3List extends Generation2List {

    // Author: John Doe
    // Date: 3/17/2002
    // Current revision: 6
    // Last modified: 4/12/2004
    // By: Jane Doe
    // Reviewers: Alice, Bill, Cindy

    // class code goes here

}
```

Следующая аннотация может позволить генерировать ту же информацию

```
@interface ClassPreamble {

    String author();
    String date();
    int currentRevision() default 1;
    String lastModified() default "N/A";
    String lastModifiedBy() default "N/A";
    String[] reviewers(); // Note use of array
}
```

Определение аннотации напоминает определение интерфейса, но, в отличии от определения интерфейса ключевому слову `interface` предшествует символ `@` (`@` = "AT" сокращение от Annotation Type).

Тело определения аннотации содержит определения элементов (*annotation type element*).

Определения последних напоминают определения методов. Отметим, что элементы могут иметь значения по умолчанию.

```
@ClassPreamble (
    author = "John Doe",
    date = "3/17/2002",
    currentRevision = 6,
    lastModified = "4/12/2004",
    lastModifiedBy = "Jane Doe"
    reviewers = {"Alice", "Bob", "Cindy"} // Note array notation
)
public class Generation3List extends Generation2List {
    // реализация класса
}
```

Аннотации, используемые компилятором

В спецификации языка Java упоминаются следующие аннотации : `@Deprecated`, `@Override`, и `@SuppressWarnings`.

@Deprecated— помеченный элемент устарел и не должен более использоваться. Использование элемента программы, помеченного аннотацией приводит к выводу предупреждающего сообщения на этапе компиляции.

```
// Javadoc comment follows
/**
 * @deprecated
 * explanation of why it was deprecated
 */
@Deprecated
static void deprecatedMethod() { }
```

@Override—элемент переопределяет элемент суперкласса

```
// mark method as a superclass method
// that has been overridden
@Override
int overriddenMethod() { }
```

@SuppressWarnings—отключает вывод предупреждений компилятором.

```
// use a deprecated method and tell
// compiler not to generate a warning
@SuppressWarnings("deprecation")
void useDeprecatedMethod() {
    objectOne.deprecatedMethod(); //deprecation warning - suppressed
}
```

Обработка аннотаций

Для обработки аннотаций может быть написан обработчик аннотаций (*annotation processor*). Такой обработчик читает Java-программу и на основе аннотаций выполняет определённые действия (генерация XML, генерация кода, документирование). В Java 5.0 для обработки аннотаций используется утилита `apt`. Начиная с 6-ой версии функциональность `apt` вошла в Java компилятор.

```
import java.lang.annotation.*;

@Retention(RetentionPolicy.RUNTIME)
@interface AnnotationForRuntime {

    // Elements that give information
    // for runtime processing

}
```

Дополнительная литература:

<https://java.sun.com/docs/books/tutorial/java/javaOO/annotations.html>

<http://java.sun.com/j2se/1.5.0/docs/guide/language/annotations.html>