

Рефлексия

№ урока: 4 Курс: Java Advanced

Средства обучения: Компьютер с установленной IntelliJ IDEA.

Обзор, цель и назначение урока

Понимание и использование механизма рефлексии (Reflection API) позволяет программисту получить данные о классе во время выполнения программы.

Изучив материал данного занятия, учащийся сможет:

- Понимать механизм работы Reflection API.
- Определить класс объекта.
- Получить информацию о модификаторах класса, полях, методах, конструкторах и суперклассах.
- Выяснить, какие константы и методы принадлежат интерфейсу.
- Создать экземпляр класса, имя которого неизвестно до момента выполнения программы.
- Получить и установить значение свойства объекта.
- Вызвать метод объекта.
- Модифицировать поведение и состояние объектов, в том числе инкапсулированных.
- Применять рефлексия для модификации программы в процессе исполнения.

Содержание урока

1. Получение информации о классе.
2. Получение информации о классе, имя которого неизвестно до момента исполнения.
3. Получение модификаторов доступа.
4. Получение пакета, в котором содержится класс.
5. Получение родительского класса и реализуемых интерфейсов.
6. Получение конструкторов.
7. Получение полей и методов.
8. Получение геттеров и сеттеров.
9. Получение приватных полей и методов.
10. Изменение доступа инкапсулированных полей.

Резюме

- В информатике отражение или рефлексия (англ. reflection) означает процесс, во время которого программа может отслеживать и модифицировать собственную структуру и поведение во время выполнения. Парадигма программирования, положенная в основу отражения, называется рефлексивным программированием.
- В большинстве современных компьютерных архитектур программные инструкции (код) хранятся как данные. Разница между кодом и данными в том, что выполняя код, компьютеры обрабатывают данные. То есть инструкции выполняются, а данные обрабатываются так, как предписано этими инструкциями. Однако программы, написанные на Java, способны обрабатывать собственные инструкции как данные и выполнять, таким образом, рефлексивные модификации.
- Все классы для работы с рефлексией расположены в пакете [java.lang.reflect](#).
- Для использования рефлексии необходимо создать объект типа [java.lang.Class](#). Метод [getClass\(\)](#) полезен когда есть экземпляр объекта, но неизвестно какого именно класса этот экземпляр.
- Если есть класс, для которого в момент компиляции известен тип, то получить экземпляр класса можно с помощью метода [class](#).

- Если имя класса неизвестно в момент компиляции, но становится известным в процессе выполнения программы, то необходимо использовать метод `forName()` чтобы получить объект типа `Class`.
- Метод `getPackage()` позволяет получить пакет, в котором содержится класс. Возвращает объект класса `Package`.
- Метод `getInterfaces()` позволяет получить реализуемые интерфейсы. Возвращает массив класса `Class`.
- Метод `getDeclaredFields()` позволяет получить все поля класса, за исключением унаследованных. Возвращает массив класса `Fields`. Метод `getFields()` возвращает только `public` поля.
- Метод `getDeclaredMethods()` позволяет получить все методы класса, за исключением унаследованных. Возвращает массив класса `Method`. Метод `getMethods()` возвращает только `public` методы.
- Метод `getDeclaredConstructors()` позволяет получить все конструкторы класса. Возвращает массив класса `Constructor`. Метод `getConstructors()` возвращает только `public` конструкторы.
- Метод `getAnnotations()` позволяет получить аннотации. Возвращает массив класса `Annotation`. Чтобы получить сам класс аннотации необходимо использовать метод `annotationType()`.
- Метод `getModifiers()` позволяет получить модификаторы класса. Возвращает значение типа `int` с константным кодом модификатора. Чтобы декодировать этот код, необходимо использовать методы класса `Modifier`. Модификаторы, зарезервированные JVM: `public`, `protected`, `private`, `final`, `static`, `abstract`, `interface`.
- Чтобы получить доступ к изменению полей и методов с модификатором `private`, необходимо использовать метод `setAccessible()` класса `Field` или `Method` соответственно.

Закрепление материала

1. Что такое рефлексия?
2. Как получить отражение класса через экземпляр класса?
3. Как получить значение модификатора и преобразовать его в понятный человеку вид?
4. Как получить реализуемые интерфейсы?
5. Как получить метод с модификатором `protected`? Как его вызвать?
6. Как получить инкапсулированные методы с модификатором `private`?
7. Как изменить значение инкапсулированного поля с модификатором `private`?

Дополнительное задание

Написать программу-рефлектор, которая позволит получить всю информацию о классах, которые находятся в папке `Additional`.

Самостоятельная деятельность учащегося

Задание 1

Выучить основные конструкции и понятия, рассмотренные на уроке.**

Задание 2

```
*****
*****
*****
*****
*****
```

Задание 3

```
*****
*****
*****
```


Задание 4

Рекомендуемые ресурсы

Руководство по Reflection API

<https://docs.oracle.com/javase/tutorial/reflect/>

Класс Class

<http://docs.oracle.com/javase/7/docs/api/java/lang/Class.html>

Пакет java.lang.reflect

<http://docs.oracle.com/javase/7/docs/api/index.html?java/lang/reflect/package-summary.html>

Дополнительное руководство по Reflection API

<http://tutorials.jenkov.com/java-reflection/index.html>