Семинар 4 – Рефлексия

<https://javarush.com/groups/posts/513-reflection-api-refleksija-temnaja-storona-java>

# Задача 8(10)

1. Что такое рефлексия в Java?

Рефлексия в Java осуществляется с помощью Java Reflection API.

Рефлексия Java — это процесс, при котором данные кода программы исследуются прямо во время ее работы. Также этот процесс позволяет изменять и исследовать классы, интерфейсы, поля, методы программы, не зная их имен. Не нужно путать рефлексию с интроспекцией! Это часто делают молодые Java-разработчики.

Интроспекция — это процесс проверки кода, при котором можно только увидеть типы компонентов программы прямо во время ее выполнения.

1. Каковы её минусы?

Уменьшается производительность. Рефлексия работает в динамическом режиме и поэтому постоянно «сканирует» классы для загрузки нужного вам. Это очень сильно снижает скорость работы всего приложения.

Уменьшается безопасность. Используя рефлексию, можно случайно воздействовать на ту часть кода, на которую не нужно воздействовать вообще. К примеру, можно получить разрешение воздействовать на закрытые поля классов и изменить их значение.

Усложняется поддержка. После рефлексии нам достается довольно трудный для понимания код, который сложно читать и отлаживать.

1. что такое класс Class?

**Class**. Он представляет классы и интерфейсы в исполняемом приложении Java.

Экземпляры класса Class представляют классы и интерфейсы в запущенном приложении Java. Примитивные типы Java (boolean, byte, char, short, int, long, float и double) и ключевое слово void также представлены как объекты класса. У него нет открытого конструктора. Объекты класса создаются автоматически виртуальной машиной Java ([JVM](https://translated.turbopages.org/proxy_u/en-ru.ru.9ce39866-63f3b30c-87022441-74722d776562/https/www.geeksforgeeks.org/jvm-works-jvm-architecture/)). Это конечный класс, поэтому мы не можем его расширять.

Методы класса class широко используются в API Reflection.

1. Что такое класс Field?

Класс Field предоставляет информацию и динамический доступ к одному полю класса или интерфейса. Field также разрешает расширение преобразований во время операции get или set access, но выдает исключение IllegalArgumentException, если происходит сужение преобразования.

1. Что такое класс Constructor?

Constructorпредоставляет информацию и доступ к одному конструктору для класса.

Constructorразрешает расширяющие преобразования при сопоставлении фактических параметров с newInstance() с формальными параметрами базового конструктора, но выдает исключение, IllegalArgumentExceptionесли произойдет сужающее преобразование.

1. Что такое класс Method?

Предоставляет Methodинформацию об одном методе класса или интерфейса и доступ к нему. Отраженный метод может быть методом класса или методом экземпляра (включая абстрактный метод).

Разрешает Methodрасширяющие преобразования при сопоставлении фактических параметров для вызова с формальными параметрами базового метода, но выдает исключение, IllegalArgumentExceptionесли произойдет сужающее преобразование.

1. Приведите примеры где можно использовать рефлексию?
2. Если вы внимательно просмотрели содержимое класса, то наверняка увидели отсутствие **getter**’a для поля **name**. Само поле **name** помечено модификатором доступа **private**, обратиться к нему вне самого класса у нас не выйдет => мы не можем получить его значение. “Так в чем проблема? — скажете вы. — Допиши **getter** или измени модификатор доступа”. И вы будете правы, но, что если **MyClass** находится в скомпилированной aar библиотеке или в другом закрытом модуле без доступа к редактированию, а на практике такое случается крайне часто. И какой-то невнимательный программист просто забыл написать **getter**. Самое время вспомнить о рефлексии!
3. Узнать / определить класс объекта.
4. Получить информацию о модификаторах класса, полях, методах, константах, конструкторах и суперклассах.
5. Выяснить, какие методы принадлежат реализуемому интерфейсу / интерфейсам.
6. Создать экземпляр класса, когда имя класса неизвестно до момента выполнения программы.
7. Получить и установить значение поля объекта по имени.
8. Вызвать метод объекта по имени.

# Задача 9 (10)

Имеется класс ExampleClass.

|  |
| --- |
| package org.example;  public class ExampleClass {  private int privateVariable=5;  protected float protectedVariable=6.0f;  public String publicVariable="whatever!";  double noModifierVariable=1111.000000;   public ExampleClass()  {   }   public ExampleClass(int privateVariable, float protectedVariable, String publicVariable, double noModifierVariable) {  this.privateVariable = privateVariable;  this.protectedVariable = protectedVariable;  this.publicVariable = publicVariable;  this.noModifierVariable = noModifierVariable;  }   public int getPrivateVariable() {  return privateVariable;  }   public void setPrivateVariable(int privateVariable) {  this.privateVariable = privateVariable;  }    public float getProtectedVariable() {  return protectedVariable;  }   public void setProtectedVariable(float protectedVariable) {  this.protectedVariable = protectedVariable;  }   public String getPublicVariable() {  return publicVariable;  }   public void setPublicVariable(String publicVariable) {  this.publicVariable = publicVariable;  }  @Deprecated()  public double getNoModifierVariable() {  return noModifierVariable;  }   public void setNoModifierVariable(double noModifierVariable) {  this.noModifierVariable = noModifierVariable;  }   private void privateMethod()  {   } } |

Используя рефлексию в Java:

1. Вывести все переменные класса вместе с типом и модификатором доступа.
2. Вывести все методы класса.
3. Вывести все конструкторы класса.
4. Вывести всех значений переменных класса.
5. Вывести всех аннотацией в классе.
6. Создать объект типа ExampleClass.