



持续更新中

回复“1024”获取Java架构师资源



微信搜一搜

Java研发军团

Java研发军团《Java面试手册》V1.0
公众号后台回复“面试手册”

Java反射面试题

1、除了使用new创建对象之外，还可以用什么方法创建对象？

使用Java反射可以创建对象!

2、Java反射创建对象效率高还是通过new创建对象的效率高？

通过new创建对象的效率比较高。通过反射时，先找查找类资源，使用类加载器创建，过程比较繁琐，所以效率较低

3、java反射的作用

反射机制是在运行时，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意个对象，都能够调用它的任意一个方法。在java中，只要给定类的名字，就可以通过反射机制来获得类的所有信息。

这种动态获取的信息以及动态调用对象的方法的功能称为Java语言的反射机制。

4、哪里会用到反射机制？

jdbc就是典型的反射

```
Class.forName('com.mysql.jdbc.Driver.class');//加载MySQL的驱动类
```

这就是反射。如hibernate，struts等框架使用反射实现的。

5、反射的实现方式：

第一步：获取Class对象，有4中方法：

- 1) Class.forName("类的路径");
- 2) 类名.class
- 3) 对象名.getClass()
- 4) 基本类型的包装类，可以调用包装类的Type属性来获得该包装类的Class对象

6、实现Java反射的类：

- 1) Class：表示正在运行的Java应用程序中的类和接口

注意：所有获取对象的信息都需要Class类来实现。

- 2) Field：提供有关类和接口的属性信息，以及对它的动态访问权限。
- 3) Constructor：提供关于类的单个构造方法的信息以及它的访问权限
- 4) Method：提供类或接口中某个方法的信息

7、反射机制的优缺点：

优点：

- 1) 能够运行时动态获取类的实例，提高灵活性；
- 2) 与动态编译结合

缺点：

- 1) 使用反射性能较低，需要解析字节码，将内存中的对象进行解析。

解决方案：

- 1、通过setAccessible(true)关闭JDK的安全检查来提升反射速度；
- 2、多次创建一个类的实例时，有缓存会快很多
- 3、ReflfectASM工具类，通过字节码生成的方式加快反射速度
- 2) 相对不安全，破坏了封装性（因为通过反射可以获得私有方法和属性）

8、Java 反射 API

反射 API 用来生成 JVM 中的类、接口或则对象的信息。

1. Class 类：反射的核心类，可以获取类的属性，方法等信息。
2. Field 类：Java.lang.reflec 包中的类，表示类的成员变量，可以用来获取和设置类之中的属性值。

1. Method 类：Java.lang.reflec 包中的类，表示类的方法，它可以用来获取类中的方法信息或者执行方法。

1. Constructor 类：Java.lang.reflec 包中的类，表示类的构造方法。

9、反射使用步骤（获取 Class 对象、调用对象方法）

1. 获取想要操作的类的 Class 对象，他是反射的核心，通过 Class 对象我们可以任意调用类的方法。
2. 调用 Class 类中的方法，既就是反射的使用阶段。
3. 使用反射 API 来操作这些信息。

10、获取 Class 对象有几种方法

调用某个对象的 getClass()方法

```
Person p=new Person();  
  
Class clazz=p.getClass();
```

调用某个类的 class 属性来获取该类对应的 Class 对象

```
Class clazz=Person.class;
```

使用 Class 类中的 forName()静态方法(最安全/性能最好)

```
Class clazz=Class.forName("类的全路径");（最常用）
```

当我们获得了想要操作的类的 Class 对象后，可以通过 Class 类中的方法获取并查看该类中的方法和属性。

```
//获取 Person 类的 Class 对象  
  
Class clazz=Class.forName("reflection.Person");  
//获取 Person 类的所有方法信息  
Method[] method=clazz.getDeclaredMethods();  
for(Method m:method){  
    System.out.println(m.toString());  
}  
//获取 Person 类的所有成员属性信息  
Field[] field=clazz.getDeclaredFields();  
for(Field f:field){  
    System.out.println(f.toString());  
}  
//获取 Person 类的所有构造方法信息  
Constructor[] constructor=clazz.getDeclaredConstructors();  
for(Constructor c:constructor){  
    System.out.println(c.toString());  
}
```

11、利用反射动态创建对象实例

Class 对象的 newInstance()

1. 使用 Class 对象的 newInstance()方法来创建该 Class 对象对应类的实例，但是这种方法要求该 Class 对象对应的类有默认的空构造器。

调用 Constructor 对象的 newInstance()

1. 先使用 Class 对象获取指定的 Constructor 对象，再调用 Constructor 对象的 newInstance() 方法来创建 Class 对象对应类的实例,通过这种方法可以选定构造方法创建实例。

```
//获取 Person 类的 Class 对象

Class clazz=Class.forName("reflection.Person");

//使用 .newInstance 方法创建对象

Person p=(Person) clazz.newInstance();

//获取构造方法并创建对象

Constructor
c=clazz.getDeclaredConstructor(String.class,String.class,int.class);

//创建对象并设置属性13/04/2018

Person p1=(Person) c.newInstance("李四","男",20);
```