



Java 核心技术(高阶)

第三章 反射

第一节 反射入门

华东师范大学 陈良育



问题

- 问题：如何知道一个类有多少个方法？

```
public class A {  
    public void hello()  
    {  
        System.out.p  
    }  
}
```

print(boolean b) : void - PrintStream
print(char c) : void - PrintStream
print(char[] s) : void - PrintStream
print(double d) : void - PrintStream
print(float f) : void - PrintStream
print(int i) : void - PrintStream
print(long l) : void - PrintStream
print(Object obj) : void - PrintStream
print(String s) : void - PrintStream
printf(String format, Object... args) : PrintStream

Press 'Alt+/' to show Template Proposals

```
public class Tiger {  
  
    public void move()  
    {  
        System.out.println("I can move fast");  
    }  
  
    public static void main(String[] args) {  
        Tiger t = new Tiger();  
        t.m  
    }  
}
```

move() : void - Tiger
main(String[] args) : void - Tiger



反射(1)

- 反射: reflection
 - 程序可以访问、检测和修改它本身状态或行为的能力, 即自描述和自控制。
 - 可以在运行时加载、探知和使用编译期间完全未知的类。
 - 给Java插上动态语言特性的翅膀, 弥补强类型语言的不足。
 - java.lang.reflect包, 在Java 2时代就有, 在Java 5得到完善



反射(2)

- 反射: reflection
 - 在运行中分析类的能力
 - 在运行中查看和操作对象
 - 基于反射自由创建对象
 - 反射构建出无法直接访问的类
 - set或者get到无法访问的成员变量
 - 调用不可访问的方法
 - 实现通用的数组操作代码
 - 类似函数指针的功能



创建对象(1)

- 问题：如何(自由的)创建一个对象来调用它的方法？
- 方法1：静态编码&编译

```
public class A {  
    public void hello()  
    {  
        System.out.println("hello from A");  
    }  
}
```

```
//第一种 直接new 调用构造函数  
A obj1 = new A();  
obj1.hello();
```



创建对象(2)

- 方法2: 克隆(clone)

```
public class B implements Cloneable {  
    public void hello()  
    {  
        System.out.println("hello from B");  
    }  
  
    protected Object clone() throws CloneNotSupportedException  
    {  
        return super.clone();  
    }  
}
```

//第二种 clone
//obj3 是obj2的克隆对象 没有调用构造函数
B obj2 = new B();
obj2.hello();

B obj3 = (B) obj2.clone();
obj3.hello();



创建对象(3)

- 方法3: 序列化(serialization)和反序列化(deserialization)

```
public class C implements Serializable {  
    private static final long serialVersionUID = 1L;  
  
    public void hello() {  
        System.out.println("hello from C");  
    }  
}
```

//第三种 序列化 没有调用构造函数

//序列化会引发安全漏洞, 未来将被移除出JDK, 请谨慎使用!!!

```
C obj4 = new C();
```

```
ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("data.obj"));
```

```
out.writeObject(obj4);
```

```
out.close();
```

```
ObjectInputStream in = new ObjectInputStream(new FileInputStream("data.obj"));
```

```
C obj5 = (C) in.readObject();
```

```
in.close();
```

```
obj5.hello();
```




创建对象(4)

- 方法4和5: 反射

//第四种 **newInstance** 调用构造函数

```
Object obj6 = Class.forName("A").newInstance();  
Method m = Class.forName("A").getMethod("hello");  
m.invoke(obj6);
```

```
A obj7 = (A) Class.forName("A").newInstance();
```

//第五种 **newInstance** 调用构造函数

```
Constructor<A> constructor = A.class.getConstructor();  
A obj8 = constructor.newInstance();  
obj8.hello();
```


总结



- 初步了解Java反射的作用
- 了解Java中创建对象的多种方法



谢谢!