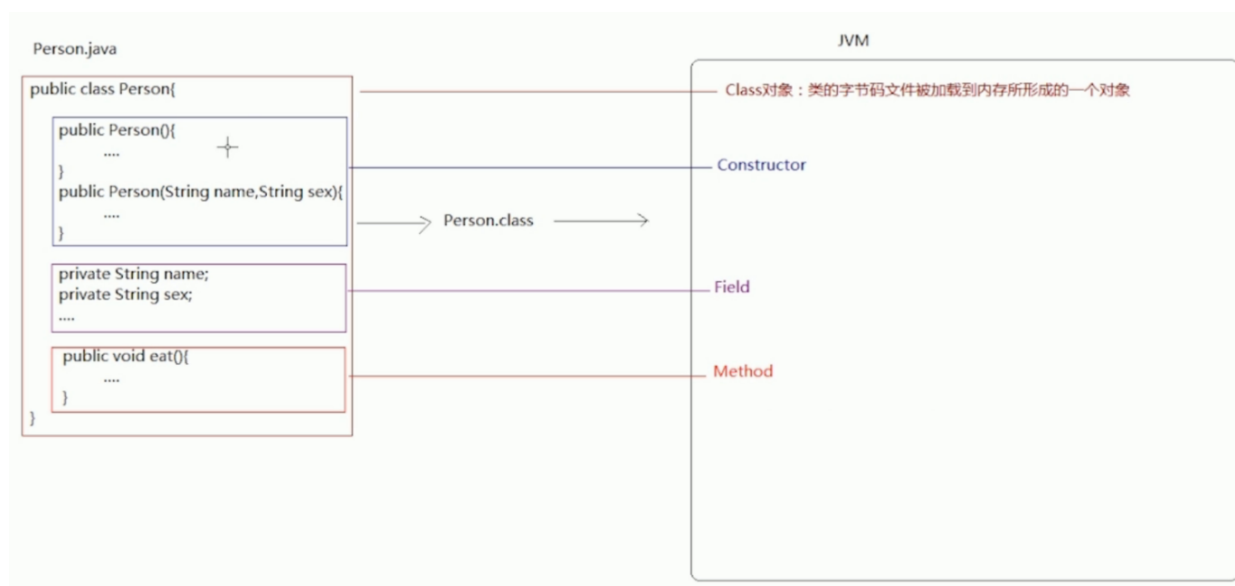


什么是Java的反射机制

- java反射机制是在运行状态中，对于任意一个类，都能够知道这个类所有属性和方法；对于任意一个对象，都能够调用它的任意方法和属性；这种动态获取信息以及动态调用对象方法的功能称为java语言的反射机制
- 作用：用来编写一些通用性比较高的代码或者框架的时候使用

反射常用对象的概述

- **Class**
 - Class类的实例表示正在运行的Java应用程序中的类和接口
- **Constructor**
 - 关于类的单个构造方法的信息以及对它的访问权限
- **Field**
 - Field提供有关类或接口的单个字段的信息，以及对它的动态访问权限
- **Method**
 - Method提供关于类或接口上单独某个方法的信息
-



Class类

- Java中java.lang.Class类用于表示一个类的字节码(.class)文件
- 如何得到某个class文件对应的Class对象
 - 已知类和对象的情况下

- - 1 类名.class
 - 2 对象.getClass() ---Object类提供

- 未知类和对象的情况下(通常使用的这个, **推荐**)

```
1 Class.forName("包名.类名")
```

- Class类代表某个类的字节码, 并提供了加载字节码的方法: `forName("包名.类名")`, `forName`方法用于加载类字节码到内存中, 并封装成一个Class对象

Constructor类

- Constructor类的实例对象代表类的一个构造方法
- 得到某个类所有的构造方法

- ```
1 Constructor []constructors = Class.forName("java.lang.String").getConstructors();
```

- 得到指定的构造方法并调用

- ```
1 Constructor constructor =  
  Class.forName("java.lang.String").getConstructor(String.class);  
2 String str = (String)constructor.newInstance("abc");
```

- Class类的`newInstance()`方法用来调用类的默认构造方法

- ```
1 String obj =(String)Class.forName("java.lang.String").newInstance();
```

---

## Field类

- Field类代表某个类中的一个成员变量, 并提供动态的访问权限
- Field对象的获得
  - 得到所有的成员变量

- ```
1 Field[] fields =c.getFields(); //取得所有public属性(包括父类继承)  
2 Field[] fields =c.getDeclaredFields(); //取得所有声明的属性
```

- 得到指定的成员变量

- ```
1 Field name = c.getField("name");
2 Field name = c.getDeclaredField("name");
```

- 设置Filed变量是否可以访问

- ```
1 field.setAccessible(boolean);
```

- Field变量值的读取、设置

- ```
1 field.get(obj)
2 field.set(obj,value);
```

---

## Method类

- Method类代表某个类中的一个成员方法
- Method对象的获得
  - 获得所有方法

- ```
1 getDeclaredMethods()
2 getMethods()
```

- 获得指定方法

- ```
1 getDeclaredMethod(String name,Class<?>...parameterTypes)
2 getMethod(String name,Class<?>...parameterTypes)
```

- 通过反射执行方法

- ```
1 invoke(Object obj,Object...args)
```