1. 获取构造方法

在一个类之中除了继承的关系之外，最重要的就是类中的结构处理，而类之中的结构里面首先观察的就是构造方法的使用问题，实际上在之前通过反射实例化对象就已经接触到构造方法的问题了---当我们调用newInstance的时候我们就必须调用构造方法完成（getDeclaredConstructor）

·实例化方法代替:clazz.getDeclaredConstructor().newInstance();

所有类的构造方法可以通过Class类完成，该类中定义有如下几个方法:

|  |  |
| --- | --- |
| ·获取所有构造方法 | public Constructor<?>[] getDeclaredConstructors​()  throws SecurityException |
| ·获取指定构造方法(根据参数类型) | public Constructor<T> getDeclaredConstructor​(Class <?>... parameterTypes) throws NoSuchMethodException, SecurityException |
| ·获取所有构造方法 | public Constructor<?>[] getConstructors()throws SecurityException |
| ·获取指定构造方法 | public Constructor<T> getConstructor(Class<?>... parameterTypes) throws NoSuchMethodException, SecurityException |

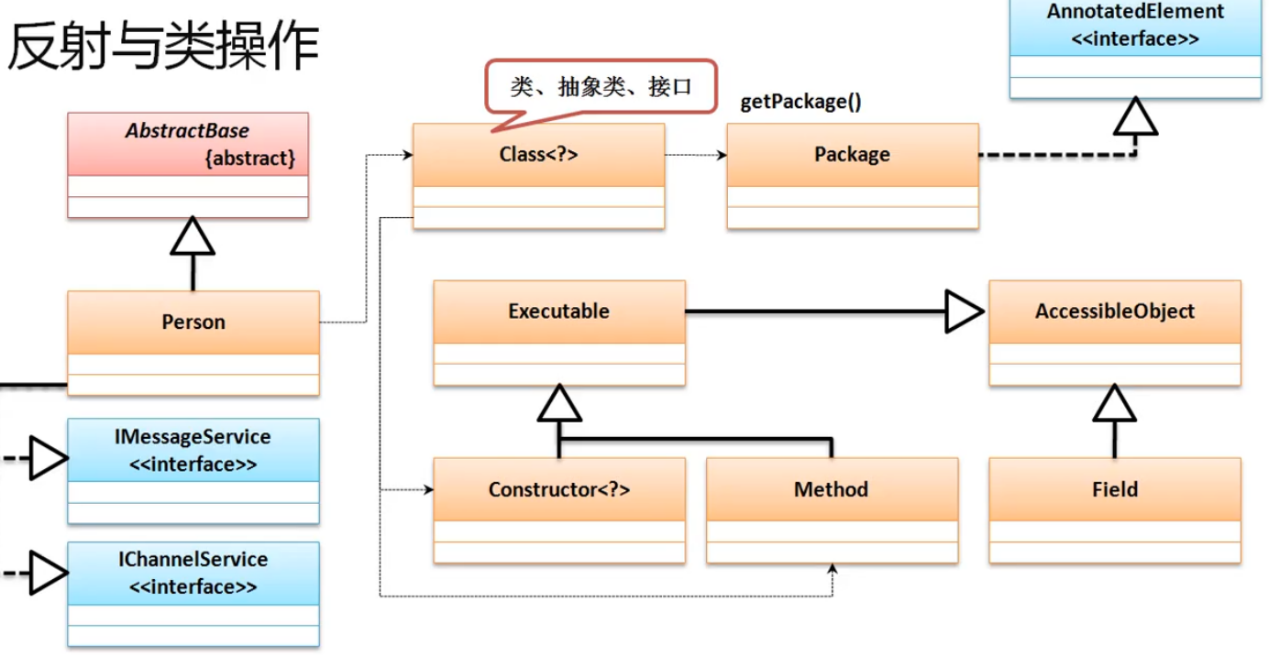
Declared是本类的意思。如果此时程序类之中的有参构造方法进行Person类对象的实例化处理，这个时候必须要指明要调用的构造，而后通过constructor类之中提供的实例化方法操作，因为一旦指向了有参构造那么在整个程序之中就必须在构造调用的时候传递参数:

·实例化方法: public T newInstance​(Object... initargs)

throws InstantiationException, IllegalAccessException, IllegalArgumentException, InvocationTargetException

范例:调用指定构造实例化对象

|  |
| --- |
|  |

 虽然程序允许调用有参构造，但是实际开发中所有使用反射的类中最好提供无参构造，因为这样的实例化可可以达到统一性。