

Java.lang包下：

Class<T>：表示正在运行的Java应用程序中的接口和类。

Java.lang.reflect包下：

Field类：代表类的成员变量（属性）。

Method类：代表类的方法。

Constructor类：代表类的构造方法。

Array类：提供动态创建数组，及访问数组元素的静态方法。

实例化对象方法：

- 无参构造方法：Class.newInstance();

```
eg. Class c = String.class;  
    Object str = c.newInstance();
```

- 先通过Class对象获取指定的Constructor对象，再调用Constructor对象的newInstance()方法来创建实例。可用指定构造器构造类实例。无参构造：Class.getConstructor(new Class[]{}).newInstance(new Object[]{});

```
eg. Class<?> c = String.class;  
    Constructor constructor =  
c.getConstructor(String.class);  
    Object obj = constructor.newInstance("2333");
```

获取并调用方法：

- 获取当前类声明的所有Method。不包括继承的方法。

```
Method[] arrMethods = classType.getDeclaredMethods();
```

- 获取当前类及超类的public Method。

```
Method[] arrMethods = classType.getMethods();
```

- 获取当前类及超类指定的public Method。

```
Method method = classType.getMethod(String name, Class<?>... parameterTypes);
```

第一个参数为指定方法名称，后面的参数为方法的参数对应的class对象。

- 获取当前类指定的Method。

```
Method method = classType.getDeclaredMethods();
```

- 通过反射动态运行指定Method，私有方法或属性需method的方法设定后才可访问

```
Object obj = method.invoke(Object obj, Object... args);
```

获取并调用属性：

- `getFiled`：访问公有的成员变量
- `getDeclaredField`：所有已声明的成员变量，但不能得到其父类的成员变量

`getFiled`s 和 `getDeclaredFields` 方法用法同上。eg.

```
Field[] arrFields = classType.getFields();
```

```
Field field = classType.getField(String name);
```

- 通过反射动态设定Field的值

```
field.set(Object obj,Object value);
```

- 通过反射动态获取Field的值

```
Object obj = fiald.get(Object obj);
```

incoke方法详解: <https://www.sczyh30.com/posts/Java/java-reflection-2/>