1. **本章的主要內容**



1. **關於反射的理解**  
   Reflection （反射）是被視為動態語言的關鍵，反射機制允許程序在執行期借助於 Reflection API 取得任何類的內部信息，並能直接操作任意對象的內部屬性及方法體會反射機制的”動態性”  
   框架 = 反射 + 註解 + 設計模式
2. **體會反射機制的”動態性”**

@Test

public void test2(){

for(int i = 0;i < 100;i++){

int num = new Random().nextInt(3);//0,1,2

String classPath = "";

switch(num){

case 0:

classPath = "java.util.Date";

break;

case 1:

classPath = "java.lang.Object";

break;

case 2:

classPath = "com.atguigu.java.Person";

break;

}

try {

Object obj = getInstance(classPath);

System.out.println(obj);

} catch (Exception e) {

e.printStackTrace();

}

}

}

/\*

創建一個指定類的對象。 classPath:指定類的全類名

\*/

public Object getInstance(String classPath) throws Exception {

Class clazz = Class.forName(classPath);

return clazz.newInstance();

}

1. **反射機制提供的功能**

* 在運行時判斷任意一個對象所屬的類
* 在運行時構造任意一個類的對象
* 在運行時判斷任意一個類所具有的成員變量和方法
* 在運行時獲取泛型信息
* 在運行時調用任意一個對象的成員變量和方法
* 在運行時處理注解
* 生成動態代理

1. **相關的API**java.lang.Class:反射的源頭  
   java.lang.reflect.Method  
   java.lang.reflect.Field  
   java.lang.reflect.Constructor  
   **……**