**Java反射**

**一．选择题**

1. 如何获取一个类的字节码文件对象

A、对象名.class

B、类名.getClass()

C、Object类中的forName()方法

D、以上说法都不正确

2. 关于反射机制下列说法错误的是

A、反射可以获取类中所有的属性和方法

B、反射可以构造类的对象，并获取其私有属性的值

C、反射机制指的是在程序编译期间，通过.class文件加载并使用一个类的过程

D、暴力反射可以获取类中私有的属性和方法

3..下列关于反射和泛型的联系说法正确的是 （ ）

A、泛型只在编译时有效，无法在运行期获取泛型的具体类型

B、反射机制可以获取泛型的具体类型

C、可以通过反射绕过泛型检查，因为运行期泛型根本没有用

D、以上说法都不对

4. 使用反射机制获取一个类的属性，下列关于getField()方法说法正确的是 （ ）

A、该方法需要一个String类型的参数来指定要获取的属性名

B、该方法只能获取私有属性

C、该方法能够获取所有属性

D、该方法可以获取私有属性，但使用前必须先调用setAccessible(true)方法

5. 下列关于通过反射方式获取方法并执行的过程说法正确的是 （ ）

A、通过对象名.方法名(参数列表)的方式调用该方法

B、通过Class.getMethod(方法名，参数类型列表)的方式获取该方法

C、通过Class.getDeclaredMethod(方法名，参数类型列表)获取私有方法

D、通过invoke(对象名,参数列表)方法来执行一个方法

6. 反射能做什么 （ ）

A.在运行时判断任意一个对象所属的类

B.在运行时构造任意一个类的对象

C.在运行时判断任意一个类所具有的成员变量和方法

D.在运行时调用任意一个对象的方法

7. 假定Tester 类有如下test方法：

public int test(int p1, Integer p2)

以下哪段代码能正确编译并动态调用一个Tester 对象的test方法？ （ ）

A、Class classType=Tester.class;

Object tester=classType.newInstance();

Method addMethod=classType.getMethod("test", int.class, int.class);

Object result=addMethod.invoke(tester, 100, 200);

B、Class classType=Tester.class;

Object tester=classType.newInstance();

Method addMethod=classType.getMethod("test", int.class, int.class);

int result=addMethod.invoke(tester, 100, 200);

C、Class classType=Tester.class;

Object tester=classType.newInstance();

Method addMethod=classType.getMethod("test", int.class,Integer.class);

Object result=addMethod.invoke(tester, 100, 200);

D、Class classType=Tester.class;

Object tester=classType.newInstance();

Method addMethod=classType.getMethod("test", int.class,Integer.class);

Integer result=addMethod.invoke(tester, 100, 200);

8. 以下哪些方法在Class类中定义？（ ）

A．getConstructors()

B．getPrivateMethods()

C．getDeclaredFields()

D．getImports()

E．setField()

9. 假定有Tester类9.假定有Tester类以下那种方法能正确的构造出一个Tester对象（ ）

Class<Tester> classType = Tester.class;

Constructor<Tester> constructor = classType.getConstructor();以下那种方法能正确的构造出一个Tester对象（ ）

A.Class.newInstance();

B.constructor.newInstance();

C.Tester.newInstance();

D.classType.newInstance();

10. 以下代码运行的结果是（ ）

public class Person {

public String eat(Integer num, String food) {

System.out.print("吃" + num + "个" + food + " ");

return food;

}

public static void main(String[] args) {

Class clazz = Person.class;

Person p = (Person) clazz.newInstance();

Method method = clazz.getMethod("eat", Integer.class, String.class);

Class type = method.getReturnType();

method.invoke(p, 1, "苹果");

System.out.print(type.getSimpleName() + " ");

System.out.print(method.getParameterTypes()[1] instanceof Class);

}

}

A.吃一个苹果 java.lang.String true

B.吃一个苹果 String true

C.吃一个苹果 java.lang.String false

D.吃一个苹果 String false

二．判断题

11. JAVA 反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法；这种动态获取的信息以及动态调用对象的方法的功能称为java 语言的反射机制。（ ）

A.对

B.错

12. Class类的对象用于表示当前运行的 Java 应用程序中的类和接口，Class类是一个未继承Object类的特殊类。（ ）

A.对

B.错

1. 获取到一个类的Class对象后，可以实现类的动态代理、获取父类的泛型类型、获取当前类实现的接口等操作。 （ ）

A.对

B.错

14. 在反射机制中，把类中的成员（构造方法、成员方法、成员变量）都封装成了对应的类进行表示。 （ ）

A.对

B.错