

第 5 组

一、选择题

1. 已知有下面类的说明:

```
public class X5_1_1 extends x{  
    private float f =10.6f;  
    int i=16;  
    static int si=10;  
    public static void main(String[] args) {  
        X5_1_1 x=new X5_1_1();  
    }  
}
```

在 main() 方法中, 下面哪条语句的用法是正确的? (A)

A. x.f B. this.si C. X5_1_1.i D. X5_1_1.f

【解析】由于 x 是在 main 方法内部定义的对象, 因此它可以引用类的非 static 类型的属性, 因此选 A,

static

而 this 和 super 不能在 main 方法 中使用，使用类名只能引用本类的 静态属性，因此 B、C、D 不对。

2. 下列程序的运行结果是 (C)。

```
public class X5_1_2 extends x {  
    int ab() { 应改成 void.  
    static int aa=10;  
    aa++;  
    System.out.println(aa);  
}  
  
public static void main(String[] args) {  
    X5_1_2 x=new X5_1_2();  
    x.ab();  
}  
}
```

A. 10 B. 11

C. 编译错误 D. 运行成功，但不输出

【解析】方法体内的 局部变量 不能使用 static 修饰符。

3. 下面关于接口的说法中不正确的是 (C)。

- A. 接口中 所有的方法 都是抽象的
- B. 接口中 所有的方法 都是 public 访问权限
- C. 子接口继承父接口所用的关键字是 implements *extends*
- D. 接口是 Java 中的特殊类，包含常量和抽象方法

【解析】子接口 继承 父接口所用的关键字也是 extends，只有类实现接口时才使用 implements。

4. 区分类中重载方法的依据是 (A)。④ 重载 *overload*

- A. 形参列表的类型和顺序
- B. 不同的形参名称
- C. 返回值的类型不同
- D. 访问权限不同

【解析】形参表的类型和顺序不同时区分重载方法的唯一标志。

5. 子类对象能否直接向其父类赋值？父类对象能否向其子类赋值？ (B)

- A. 能，能 B. 能，不能
- C. 不能，能 D. 不能，不能

【解析】子类对象可以直接赋值给父类对象，而父类对象不能直接赋值给子类对象。

⑤ 举例：

① *Animal dog = new Dog();* ✓
子类能直接向父类赋值，具体地说：即声明了动物，具体地说该动物是一条狗。

② *Dog animal = new Animal();* ✗
父类不能直接向子类赋值。

6. Java 语言类间的继承关系是 (A)。

- A. 单继承 B. 多继承 C. 不能继承 D. 不一定

【解析】Java 语言类间的继承关系是单继承，但一个类可以实现多个接口。

7. Java 语言接口间的继承关系是 (B)。

A. 单继承 B. 多重继承 C. 不能继承 D. 不一定

【7. 解析】Java 语言接口间的继承关系允许多重继承。

8. 一个类实现接口的情况是 (A)。

A. 一次可以实现多个接口

B. 一次只能实现一个接口

C. 不能实现接口

D. 不一定

【解析】Java 语言允许一个类一次实现多个接口。

9. 定义外部类的类头时, 不可用的关键字是 (C)。

A. public B. final

C. protected D. abstract

【解析】定义外部类时不能使用 protected 关键字。

10. 如果局部变量和成员变量同名, 如何在局部变量作用域内引用成员变量? (B)

A. 不能引用, 必须改名, 使它们的名称不相同

B. 在成员变量前加 this, 使用 this 访问该成员变量

C. 在成员变量前加 super, 使用 super 访问该成员变量

D. 不影响, 系统可以自己区分

【解析】this 可以用来引用本类对象。

11. 下面说法不正确的是 (B)。

A. 抽象类既可以做父类, 也可以做子类

B. abstract 和 final 能同时修饰一个类

C. 抽象类中可以没有抽象方法, 有抽象方法的类一定是抽象类或接口

D. 声明为 final 类型的方法不能在其子类中重新定义 (即不能重写)

【解析】abstract 和 final 不能同时修饰一个类, 因为 abstract 类需要子类, 而 final 类不能有子类。

二、填空题

1. 消息就是向对象发出 服务请求, 是对 数据成员 和 成员方法 的引用。

2. 在面向对象系统中, 消息分为两类 公有消息 和 私有消息。

3. 在面向对象程序设计中, 采用 继承 机制可以有效地组织程序结构。充分利用已有的类来创建更复杂的类, 大大提高程序开发的效率, 提高代码的复用率, 降低维护的工作量。

4. 数据成员的隐藏 是指在子类中重新定义一个与父类中已定义的数据成员名完全相同的数据成员。

5. 子类可以重新定义与父类同名的成员方法, 实现对父类方法的 覆盖 (即重写) override

6. 子类在重新定义父类已有的方法时, 应保持与父类完全相同的 方法名、返回值类型 和 参数列表, 否则就不是方法的覆盖, 而是子类定义自己特有的方法, 与父类的方法无关。

7. this 代表了 当前对象 的一个引用, super 表示的是当前对象的 直接父类对象 的引用。

⑩ 构造方法, 类方法 (static 修饰的方法) 不能声明为抽象方法。

即: 抽象类不能实例化。

8. 抽象类不能 创建 对象, 该工作由抽象类派生的非抽象子类来实现。

★ 如果父类中已有同名的 abstract 方法, 则子类中就 不能 (能/不能) 再有同名的抽象方法。

10. abstract 类中不能有 private 访问权限的数据成员或成员方法。

11. interface 是声明接口的关键字, 可以把它看成一个特殊类。接口中的数据成员默认的修饰符是 public static final, 接口中的成员方法默认的修饰符是 public abstract。

12. 如果实现某接口的类不是 abstract 的抽象类, 则在类的定义部分必须 实现 该接口的所有抽象方法; 如果实现某接口的类是 abstract 的抽象类, 则它可以 不实现 该接口所有的方法。但是对于这个抽象类任何一个非抽象的子类而言, 它们父类所实现的接口中的所有抽象方法以及自身所实现接口中的抽象方法都必须有实在的 方法体。

有一个类实现了多个接口的考虑

13. 包的作用有两个, 一是 划分类名空间, 二是 控制类之间的访问。

14. 封装 也称 信息隐藏, 是指类的设计者只为类的使用者提供类的可以访问的部分 (包括类的数据成员和成员方法), 而把类中的其他成员 隐藏 起来, 使用户不能访问的机制。

15. Java 提供了 4 种访问权限来实现封装机制, 即 private、protected、默认 和 public。

16. Java 中提供两种多态机制, 重载 与 覆盖。→ 覆盖即重写。

17. 当一个构造方法需要调用另一个构造方法时, 可以使用关键字 this, 同时这个调用语句应该是整个构造方法的 第一条 可执行语句。

前半句完全是错的, 详见文档中的解释。

18. 如果子类自己没有构造方法, 那么父类也一定没有 (有/没有) 带参的构造方法; 此时它将继承父类的 无参构造方法 作为自己的构造方法; 如果子类自己定义了构造方法, 则在创建新对象时, 它将先执行 父类 的构造方法, 然后再执行自己的 构造方法。

19. 对于父类的含参数构造方法, 子类可以通过在自己的构造方法中使用 super 关键字来调用它, 但这个调用语句必须是子类构造方法的 第一条 可执行语句。

20. 创建一个名为 myPackage 的包的语句为 package myPackage;, 该语句应该放在程序 第一行 位置。

三、判断题

1. Java 语言中, 构造方法是不可继承的。(✓)

2. 子类的成员变量和成员方法的数目 一定大于 等于父类的成员变量和成员方法的数目。(×) 不一定。

3. 抽象方法是一种只有说明而无具体实现的方法。(✓)

4. Java 语言中, 所创建的子类都应有一个父类。(✓)

★ 5. 调用 this 或 super 构造方法 的语句 必须放在第一条语句。(✓) 利用 this 或 super 调用构造方法的语句必须放在第一条。

6. 一个类可以实现多个接口, 接口可以实现 “多继承”。(✓)

7. 实现接口的类不能是抽象类。(×) × 可以是抽象类来实现接口, 详见文档。

8. 使用构造方法只能给实例成员变量赋初值。?(✓)?

9. Java 语言 不允许 同时继承一个类并实现一个接口。(×) 允许

语法: class ... extends ... implements ...
class

第 6 组

一、选择题