

Java 程序设计 LAB05

1. 多态 1

执行 `java Test`，程序的输出是什么？

```
private f()
```

如果将父类中的方法声明为 `public`，而子类为 `private`，编译能通过吗？如果能，最后会输出什么？

不能，子类覆盖父类方法不能缩小父类方法的访问权限。

2. 多态 2

执行 `java Test`，程序的输出是什么？

```
sup.field = 0, sup.getField() = 1
```

```
sub.field = 1, sub.getField() = 1, sub.getSuperField() = 0
```

类的非静态属性不能体现多态性

3. 多态 3

执行 `java Test`，程序的输出是什么？

```
Base staticGet()
```

```
Derived dynamicGet()
```

类的静态属性和静态方法不能体现多态性

4. 多态 4

执行 `java Test`，程序的输出是什么？

```
A() before draw()
```

```
B.draw(), b = 0
```

```
A() after draw()
```

```
B(), b = 5
```

结合之前实验的初始化顺序和多态，给出程序这样输出的解释。

在 `main` 方法中 `new B(5)`；开始初始化类 `B`，但是类 `B` 是类 `A` 的子类，所以先初始化类 `A`。在类 `A` 和类 `B` 中都没有静态变量、静态方法以及静态代码块，所以程序首先进入到类 `A` 的构造器中（即父类构造器）进行初始化，所以输出：

```
A() before draw()
```

```
B.draw(), b = 0
```

```
A() after draw()
```

其中输出 `B.draw(), b = 0` 是因为在子类中覆盖了父类的非静态 `draw()` 方法并且初始化的对象类型是 `B`，根据多态的作用原理，应该调用子类中的非静态 `draw()` 方法。父类初始化之后开始初始化子类，即进入子类的构造器，由于在 `main` 方法中传入的参数是 `5`，故子类中的 `b=5`，输出 `B(), b = 5`

5. ShapeFactory1

代码放哪儿了？

```
oo\05
```

6. 多态 7

程序无法编译的原因有哪些？

1. 在这个地方有一个错，由于接口 **I0**, **I1** 中的方法都是 **public** 的，当一个类实现这些接口时会重写接口的方法，并且不能缩小接口中方法的访问权限，所以在重写方法时方法的权限要为 **public**。

```
class Test01 implements I0, I1{
    @Override
    void f() {}
}
```

2. 在这个地方有两个错误：由于接口 **I0**, **I2** 中的方法都是 **public** 的，当一个类实现这些接口时会重写接口的方法，并且不能缩小接口中方法的访问权限，所以在重写方法时方法的权限要为 **public**；另外，在 **I0**, **I2** 两个接口中有同名的方法并且参数也一样，但是它们的返回值不一样，这是无法用一个类同时来实现这两个接口的，因为这对于这个类来说这两个方法之间并不是重载。

```
class Test02 implements I0, I2{
    @Override
    void f() {}

    @Override
    int f() {return 0;}
}
```

3. 这个地方有两个错：由于接口 **I2**, **I3** 中的方法都是 **public** 的，当一个类实现这些接口时会重写接口的方法，并且不能缩小接口中方法的访问权限，所以在重写方法时方法的权限要为 **public**。由于接口 **I2**, **I3** 中的方法同名，但是参数不一样，所以对于这个类来说这两个方法之间是重载关系；由于在接口 **I2**, **I3** 中都定义了 **static public final** 类型的成员变量 **a**，在这个类中实现 **f()** 方法时返回 **a**，这个 **a** 不知道是接口 **I2** 还是 **I3** 中的所定义的，并且这是一个常量，不能有多个值。

```
class Test23 implements I2, I3{
    @Override
    int f() {return a;}

    @Override
    int f(int i) {return i;}
}
```

I1 extends I0 会引入新的错误吗？**I2** 呢？**I3** 呢？

I1 extends I0 不会出现新的错误；**I2 extends I0** 会出现新的错误；**I3 extends I0** 会出现新的错误

7. 策略模式

代码放哪儿了？

oo\07

8. ShapeFactory1

代码放哪儿了？

oo\08

