

Java程序设计 LAB05

```
/**
 * 学号:19373073
 * 姓名:何潇龙
 * JDK版本:jdk1.8.0_91
 * 代码文件编码方式:UTF-8
 * IDE:IDEA
 */
```

1. 多态1

- 运行 `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，在重写父类方法的时候不能降低方法的访问权限。

2. 多态2

- 运行 `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
```

- 类的非静态属性能体现多态性吗？

可以，体现在非静态属性在子类中的隐藏。向上转型时使用的仍然是父类中的非静态属性，但通过子类重写的方法能获得同名的子类非静态属性。

3. 多态3

- 运行 `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能体现多态性吗？

不能。如果有多态性的话在向上转型后应该能输出子类重写的方法中的**Derived**但并没有。静态方法与类相关联，而不是与单一的对象相关联

4. 多态4

- 运行 `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。

在使用子类创建对象时，要先加载父类的构造方法，执行
`System.out.println("A() before draw()");`
`draw();`
`System.out.println("A() after draw()");`
在执行`draw();`的时候，因为创建的是B的对象，所以执行B重写的方法`draw()`，执行时B的对象属性尚未初始化，`int`变量默认为0。之后加载子类的构造方法，B的属性被初始化，因此打出**b=5**。

6. Overload? Override?

- 这段程序是无法通过编译的，都有哪些原因呢？尝试从继承、覆盖、重载的角度考虑。

首先接口中的方法默认都是`public`的，在实现方法的时候需要显式的注出所有方法都是`public`，否则就缩小了访问权限。`Test02`中的两个`f()`只更改了返回值，参数列表没有发生变化，这不属于重写也不属于重载，是不合法的。`Test23`中出现了歧义的变量引用，因为`implements I2, I3`中都有变量`a`。

- 如果 `I1 extends I0`，会引入新的错误吗？`I2`、`I3` 也 `extends I0` 呢？

`I1`不会，可以视作是重写了`f()`方法。`I2`会有问题，因为改变了返回值但没改变参数列表，既不是重写也不是重载。而`I3`更改了参数列表，属于方法重载，也不会有问题。