

Lab 06 实验报告

班级: 222111

学号: 22371001

姓名: 卜欣然

Question 1

- 运行 `Test` 的 `main` 方法, 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。不能用private方法重写父类中的public方法。

Question 2

- 运行 `Test` 的 `main` 方法, 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?
不能。

Question 3

- 运行 `Test` 的 `main` 方法, 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?
 - 对于静态方法 `staticGet()`:
 - 在 `StaticSuper` 类中调用 `StaticSuper.staticGet()`, 会访问到 `StaticSuper` 类中的静态方法, 输出结果为 `"Base staticGet()"`。
 - 在 `StaticSub` 类中调用 `StaticSub.staticGet()`, 由于静态方法不存在多态性, 所以会直接调用 `StaticSub` 类中的静态方法, 输出结果为 `"Derived staticGet()"`。
 - 对于实例方法 `dynamicGet()`:
 - `sup.dynamicGet()` 中的 `dynamicGet()` 是一个实例方法, 由于此时 `sup` 引用的是 `StaticSuper` 类的对象, 因此会调用 `StaticSub` 类中覆盖的 `dynamicGet()` 方法, 输出结果为 `"Derived dynamicGet()"`。这体现了多态性, 调用的是实际对象的方法实现。

Question 4

- 运行 `Test` 的 `main` 方法，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。
 - 调用 `B` 构造方法，由于 `B` 没有显式调用父类构造方法，因此会隐式调用父类无参构造方法 `A()`。
 - 在调用父类构造方法 `A()` 之前，会先输出 "A() before draw()"。
 - 在父类构造方法 `A()` 中，会调用 `draw()` 方法。由于此处调用的是子类 `B` 的对象，因此会调用 `B.draw()` 方法，输出 "B.draw(), b = 0"，其中 `b` 的值为默认值 0，因为此时还未执行到 `B` 构造方法的赋值语句。
 - 在父类构造方法 `A()` 中，会接着输出 "A() after draw()"。
 - 转到 `B` 构造方法中，执行 `this.b = b;` 语句，将 `b` 的值改为传入的参数 5。
 - 最后，输出 "B(), b = 5"。

Question 6

- 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。
 - `Test01` 类中的 `f()` 方法和 `I0`，`I1` 接口中的 `f()` 方法出现冲突，需要重写该方法并明确指定实现哪个接口中的 `f()` 方法。
 - `Test02` 类中的 `f()` 方法和 `I0`，`I2` 接口中的 `f()` 方法出现冲突，同样需要重写该方法并明确指定实现哪个接口中的 `f()` 方法。此外，由于 `I2` 中的 `f()` 方法返回类型为 `int`，所以在 `Test02` 类中也需要将其返回类型改为 `int`。
 - `Test23` 类中 `f()` 方法重写了 `I2`，`I3` 接口中的 `f()` 方法，但两个接口中的 `a` 变量名相同，会导致不明确引用的编译错误，需要显式地通过接口名来访问变量。
- 如果 `I1 extends I0`，会引入新的错误吗？`I2`、`I3` 也 `extends I0` 呢？

如果 `I1`，`I3` 都继承自 `I0`，是不会引入新的错误的，因为子接口继承了父接口中的抽象方法和常量，并不会影响到其他的接口和实现类。但需要注意的是，如果多个接口中定义了同名的方法或常量，则需要在实现类中进行明确的指定和实现，否则将会出现编译错误。

`I2` 会。