学号 E21714049 专业 计算机科学与技术 姓名 梅世祺
实验日期 2019.09.04 教师签字 成绩

实验报告

【实验名称】 Java 的类和对象

【实验目的】

1. 学习 Java 类和对象的基本知识（构造函数，方法，成员变量，类方法等，值类型和引用类型，引用类型的内存模型（栈内存和堆内存，Java 中只有值传递没有引用传递…）
2. 方法的重载与多态

【实验原理】

用类描述计算机中 CPU 的速度和硬盘的容量。要求 Java 应用程序有 4 个类，名字分别是 PC，CPU，HardDisk 和 Test，其中 Test 是主类。

- PC 类与 CPU 和 HardDisk 类关联的 UML 图(见图 4.34)。

其中,CPU 类要求 getSpeed()返回 speed 的值,要求 setSpeed(int

m)方法将参数 m 的值赋值给 speed; HardDisk 类要求 getAmount(返回 mount 的值,要求 setAmount(int m)方法将参数 m 的值赋值给 amount; PC 类要求 setCPU(CPU c)将参数 c 的值赋值给 CPU,要求 setHardDisk(HardDisk h)方法将参数 h 的值赋值给 HD,要求 show()方法能显示 CPU 的速度和硬盘的容量。

- 主类 Test 的要求

(1)main 方法中创建一个 CPU 对象 cpu,cpu 将自己的 speed 设置为 2200;

(2)main 方法中创建一个 HardDisk 对象 dis,disk 将自己的 amountT 设置为 200;

(3)main 方法中创建一个 PC 对象 pc(4)pc 调用 setCPU(CPuc)方法,调用时实参是 cpu(5)pc 调用 setHardDisk(HardDiskh)方法,调用时实参是 disk(6)pc 调用 show 方法。

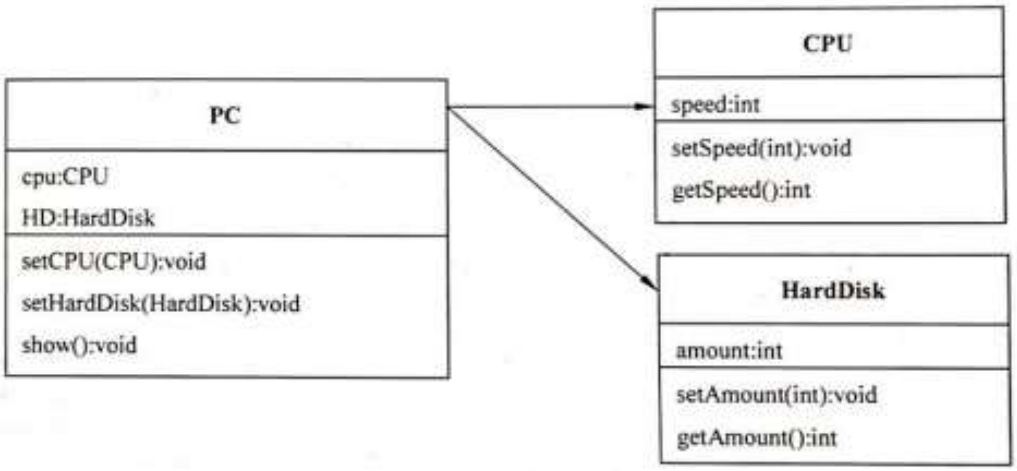

图 4.34 PC 与 CPU 和 HardDisk 关联 UML 图

【实验内容】

HardDisk.java > ...

```java
public class HardDisk {
    HardDisk(int amount) {
        this.amount = amount;
    }

    /**
     * @return the amount
     */
    public int getAmount() {
        return amount;
    }

    /**
     * @param amount the amount to set
     */
    public void setAmount(int amount) {
        this.amount = amount;
    }

    private int amount;
}
```

HardDisk 类的声明与实现如上。

CPU.java > ...

```java
public class CPU {
    CPU(int frequency) {
        this.speed = frequency;
    }

    public int getSpeed() {
        return this.speed;
    }

    public void setSpeed(int m) {
        this.speed = m;
    }

    private int speed;
}
```

CPU 类的声明与实现如上。

```java
public class PC {
    PC(CPU cpu, HardDisk hardDisk) {
        this.cpu = cpu;
        this.hardDisk = hardDisk;
    }
    /**
     * @param cpu the cpu to set
     */
    public void setCpu(CPU cpu) {
        this.cpu = cpu;
    }
    /**
     * @param hardDisk the hardDisk to set
     */
    public void setHardDisk(HardDisk hardDisk) {
        this.hardDisk = hardDisk;
    }
    public void show() {
        System.out.println(cpu.getSpeed());
    }
    private CPU cpu;
    private HardDisk hardDisk;
}
```

PC 类的声明与实现如上。

```java
Test.java > Test
public class Test {
    Run | Debug
    public static void main(String[] args) {
        CPU cpu = new CPU(2200);
        HardDisk hardDisk = new HardDisk(200);
        PC pc = new PC(cpu, hardDisk);

        pc.setCpu(cpu);
        pc.setHardDisk(hardDisk);
        pc.show();
    }
}
```

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL
/usr/lib/jvm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /home
b875548ed2de981a2239da2/redhat.java/jdt_ws/jdt.ls-java-project/bin Test
$ /usr/lib/jvm/java-11-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /ho
95b875548ed2de981a2239da2/redhat.java/jdt_ws/jdt.ls-java-project/bin Test
2200
$ 输出结果：2200 表示CPU的主频
```

Test 类的实现如上，运行结果：2200。

【小结或讨论】

1. 在 Java 还存在**可变参数**的用法，用法如下：

```java
DynamicParameters.java > ...
public class DynamicParameters {
    Run | Debug
    public static void main(String[] args) {
        DynamicParameters dynamicParameters = new DynamicParameters();
        int sum = dynamicParameters.getSum(1, 2, 3, 4, 5);
        System.out.println(sum);
    }

    public int getSum(int ...x) {
        int sum = 0;

        for (int i=0; i < x.length; i++) {
            sum += x[i];
        }

        return sum;
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
$ cd /home/lolimay/Code/Java/experiment2 ; /usr/lib/jvm/java-11-openjdk-amd64/bin/java
vscode-server/data/User/workspaceStorage/399a81e95b875548ed2de981a2239da2/redhat.java/
meters
DynamicParameters.java  Java实验报告2-E21714049-梅世祺.docx
15
$ 运行结果：15
```

由上图可知，可以通过 length 属性获取可变参数 x 的个数，且可变参数本身可以通过类似数组引用的方式获取每个参数。如 x[0] 就表示可变参数集合中的第一个参数。

2. 对象的组合

在 Java 中，如果一个对象 a 组合了对象 b,那么对象 a 就可以委托对象 b 调用其方法，即对象 a 可以通过组合的方式复用对象 b 的方法。

3. 关联关系和依赖关系

关联关系：一个类的对象是另一个类的成员；

依赖关系：一个类的对象变量是另一个类方法的参数或返回值。

4. 实例成员和类成员

用关键字 static 修饰的称为类变量，类变量与实例变量的区别：

- 不同对象的实例变量互不相同
- 所有的对象共享类变量
- 通过类名直接访问类变量

5. 方法的重载与多态

参数的个数不同或者参数的类型不同即重载

6. this 关键字

this 是 Java 的一个关键字，表示某个对象。this 可以出现在实例方法和构造方法中，但**不可以出现在类方法中。**

a) this 关键字出现在类的构造方法中，表示使用该构造方法所创建的对象

b) this 关键字出现在实例方法中，this 就表示正在调用该方法的对象。

7. 友好变量和友好方法

所谓“友好”，即指不被 public，protected 和 private 关键字修饰的变量和方法。

	同一个类	同一个包	不同包的子类	不同包的非子类
public	√	√	√	√
protected	√	√	√	
默认(default)(friendly)		√		
private	√			

Java 访问权限表

总结：

- 类是组成 Java 源文件的基本元素。
- 类体可以有两种重要的成员：成员和方法。

- 成员变量分成实例变量和类变量。类变量被该类的所有对象共享；不同对象的实例变量互不相同。
- 除构造方法外，其它方法分为实例方法和类方法。
- 实例方法既可以操作实例变量也可以操作类变量，当对象调用实例方法时，方法中的成员变量就是指分配给该对象的成员变量，其中的实例变量和其它对象的不相同，即占有不同的内存空间；类变量和其它对象的相同，即占有相同的空间。
- 类方法只能操作类变量，当对象调用类方法时，方法中的成员变量一定都是类变量，也就是说该对象和所有的对象共享类变量。
- 对象访问自己的变量以及调用方法受访问权限的限制。