

实验二 类和对象

一、实验目的：

熟悉类和对象的定义及使用；熟悉 static 关键字、this 关键字，访问权限等。

二、实验内容：

1. 编写一个 Java 应用程序，该程序有 2 个类：Vehicle(机动车)和 User(主类)。

Vehicle 类的成员变量有：speed(速度)、power(功率)。成员方法有：speedUp(int s)—加速，speedDown(int d)—减速，setPower(int p)—设置功率，getPower()—获取功率

请按下面给出的程序模版，填写缺少的代码，完成程序。

Vehicle.java

```
public class Vehicle{
    代码 1    //声明 double 型变量 speed
    代码 2    //声明 int 型变量 power
    void speedUp(int s)
    {
        代码 3 //将 s 的值累加到 speed
    }
    void speedDown(int d)
    {
        代码 4 //从 speed 中减去 d
    }
    void setPower(int p)
    {
        代码 5 //将 p 赋给 power
    }
    int getPower()
    {
        代码 6 //返回 power 的值
    }
    double getSpeed()
    {
        return speed;
    }
}
```

User.java

```
public class User {
    public static void main(String args[]){
        Vehicle car1,car2;
        代码 7    //创建对象 car1
        代码 8    //创建对象 car2
        car1.setPower(128);
        car2.setPower(76);
        System.out.println("car1 的功率是： "+car1.getPower());
    }
}
```

```

        System.out.println("car2 的功率是: "+car2.getPower());
        代码 9    //car1 调用 speedUp 方法,速度增加 80
        代码 10   //car2 调用 speedUp 方法,速度增加 80
        System.out.println("car1 目前的速度: "+car1.getSpeed());
        System.out.println("car2 目前的速度: "+car2.getSpeed());
        car1.speedDown(10);
        car2.speedDown(20);
        System.out.println("car1 目前的速度: "+car1.getSpeed());
        System.out.println("car2 目前的速度: "+car2.getSpeed());
    }
}

```

功能扩展:

在类中定义构造方法，构造具有不同初始值的对象。

改进 speedUP()方法，限制 speed 的值不能超过 200。

改进 speedDown()方法，限制 speed 的值不能小于 0。

增加一个刹车方法 void brake()，调用它能将 speed 的值变为 0。

Vehicle.java

```

public class Vehicle{
    double speed;
    int power;
    Vehicle(){};
    Vehicle(double _speed,int _power){
        speed=_speed;
        power=_power;
    }
    Vehicle(double _speed){
        speed=_speed;
    }
    Vehicle(int _power){
        power=_power;
    }
    void speedUp(int s)
    {
        speed+=s;
        if (speed>200) speed=200;
    }
    void speedDown(int d)
    {
        speed-=d;
        if (speed<0) speed=0;
    }
    void setPower(int p)
    {
        power=p;
    }
}

```

```

    }
    int getPower()
    {
        return power;
    }
    void brake() {
        speed=0;
    }
    double getSpeed()
    {
        return speed;
    }
}

```

User.java

```

public class User {
    public static void main(String args[]){
        Vehicle car1,car2;
        car1 = new Vehicle();
        car2 = new Vehicle();
        car1.setPower(128);
        car2.setPower(76);
        System.out.println("car1 的功率是: "+car1.getPower());
        System.out.println("car2 的功率是: "+car2.getPower());
        car1.speedUp(80);
        car2.speedUp(80);
        System.out.println("car1 目前的速度: "+car1.getSpeed());
        System.out.println("car2 目前的速度: "+car2.getSpeed());
        car1.speedDown(10);
        car2.speedDown(20);
        System.out.println("car1 目前的速度: "+car1.getSpeed());
        System.out.println("car2 目前的速度: "+car2.getSpeed());
    }
}

```

2. 根据下面要求，编写一个 Java 应用程序：

①定义类 `StuCard`，用到类成员变量(人数)和实例成员变量：学号、姓名、性别、专业、年级、籍贯等

②初始化的时候用到构造方法，提供两个以上构造方法

③多个类（业务类 `StuCard` 和测试主类），用到包的概念，且分别放到不同的包中

④用到类成员访问权限，要求类成员变量为 `Public`，实例成员变量为 `Private`

⑤制作公有的 `set` 和 `get` 方法与外界通过消息调用的方式通信

⑥在主类中实现对业务类 `StuCard` 的读、写、修改属性等功能

在完成以上功能要求的情况下，可以进一步扩展。

`StuCard.java`

```

package java_work.stu.stu;

public class StuCard{
    public static int number;
    private String No,name,sex,profession,grade,Birthplace;
    public StuCard(){ };
    public StuCard(String _No,String _name,String _sex,String _profession,String _grade,String
_Birthplace){
        No=_No;
        name=_name;
        sex=_sex;
        profession=_profession;
        grade=_grade;
        Birthplace=_Birthplace;
    }
    public void set(String _No,String _name,String _sex,String _profession,String _grade,String
_Birthplace){
        No=_No;
        name=_name;
        sex=_sex;
        profession=_profession;
        grade=_grade;
        Birthplace=_Birthplace;
    }
    public void get() {
        System.out.println(No);
        System.out.println(name);
        System.out.println(sex);
        System.out.println(profession);
        System.out.println(grade);
        System.out.println(Birthplace);
    }
}

```

Main.java

```

package java_work.stu.main;

import java_work.stu.stu.StuCard;

public class Main {

    public static void main(String[] args) {
        StuCard stu = new StuCard("1804030401","张世琛","男","计算机科学与技术","大二
","山东省");
    }
}

```

```

        stu.set("1803010121","徐宁","男","计科","大一","江西");
        stu.get();
    }
}

```

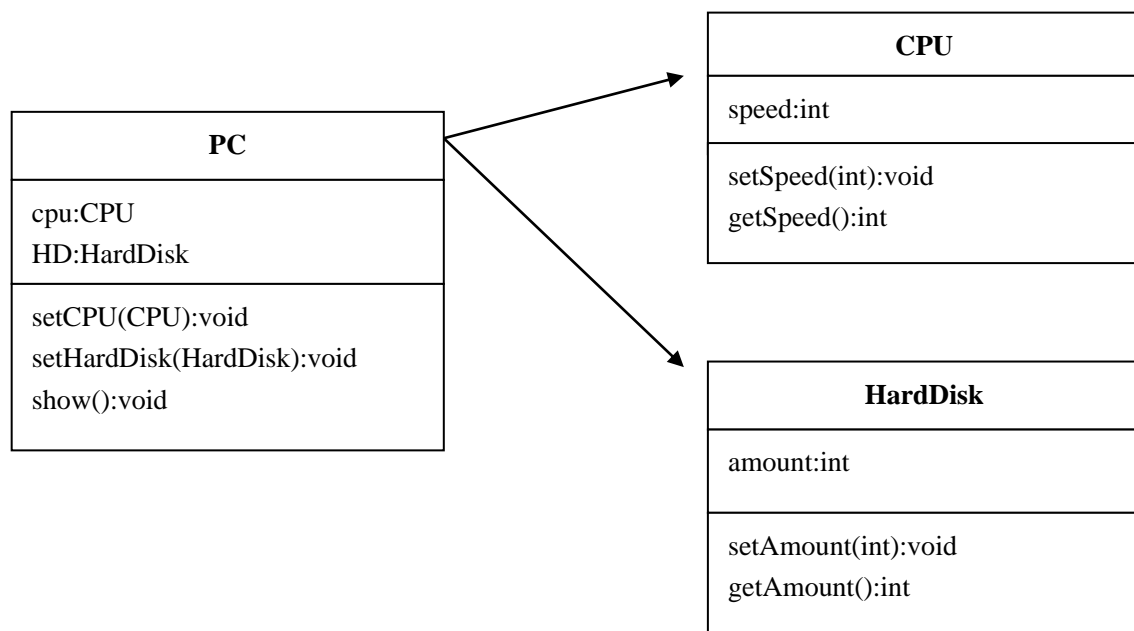
3. 用类描述计算机中 CPU 的速度和硬盘的容量。要求 Java 应用程序有 4 个类，名字分别是 PC、CPU、HardDisk 和 Test，其中 Test 是主类。

- PC 类与 CPU 和 HardDisk 类关联的 UML 图如下：

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 `amount` 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 `disk`，`disk` 将自己的 `amount` 设置为 200；
- (3) main 方法中创建一个 PC 对象 `pc`；
- (4) `pc` 调用 `setCPU(CPU c)` 方法，调用时实参是 `cpu`；
- (5) `pc` 调用 `setHardDisk(HardDisk h)` 方法，调用时实参是 `disk`；
- (6) `pc` 调用 `show()` 方法。



Test.java

```

class PC{
    CPU cpu;
    HardDisk HD ;
    void setCPU(CPU c) {
        cpu=c;
    }
}

```

```

        void setHardDisk(HardDisk h) {
            HD=h;
        }
        void show() {
            System.out.println(cpu.speed);
            System.out.println(HD.amount);
        }
    }
    class HardDisk{
        int amount;
        int getAmount() {
            return amount;
        }
        void setAmount(int m) {
            amount=m;
        }
    }

    class CPU{
        int speed;
        int getSpeed() {
            return speed;
        }
        void setSpeed(int m) {
            speed=m;
        }
    }

    }
    public class Test {
        public static void main(String[] args) {
            CPU cpu = new CPU();
            cpu.setSpeed(2200);
            HardDisk disk=new HardDisk();
            disk.setAmount(200);
            PC pc=new PC();
            pc.setCPU(cpu);
            pc.setHardDisk(disk);
            pc.show();
        }
    }
}

```

4. 共饮同井水：编写程序模拟两个村庄公用同一口井水。编写一个 **Village** 类，该类有一个静态的成员变量 **waterAmount**，用于模拟井水的水量；实例变量 **peopleNumber** 表示村庄的人数，实例变量 **name** 表示村庄的名称。定义相应的成员方法操作成员变量。在主类 **Land** 的 **main()**方法中创建两个村庄，设置村庄的名称、人数，通过一个村庄改变 **waterAmount**

的值，另一个村庄查看该值的结果。

提示：类变量是该类所有对象的共享数据；一般通过类方法操作类变量。

Land.java

```
class Village{
    static int waterAmount;
    int peopleNumber;
    String name;
    void setname(String s) {
        name=s;
    }
    void setnum(int num) {
        peopleNumber=num;
    }
    static void set(int water) {
        waterAmount=water;
    }
    static int get() {
        return waterAmount;
    }
}

public class Land {
    public static void main(String[] args) {
        Village land1=new Village();
        Village land2=new Village();
        land1.setname("zhangcun");
        land1.setnum(300);
        land1.set(100);
        System.out.println(land2.get());
    }
}
```

5. 分析程序，给出运行结果

```
class B {
    int x=100,y=200;
    public void setX(int x){
        x=x;
    }
    public void setY(int y){
        this.y=y;
    }
    public int getXYSum(){
        return x+y;
    }
}
```

```

    }
}
public class A{
    public static void main(String args[]){
        B b=new B();
        b.setX(-100);
        b.setY(-200);
        System.out.println("sum="+b.getXYSum());
    }
}

```

Sum=-100

6. 分析程序，给出输出结果

```

public class A{
    public static void main(String args[]){
        B b=new B(20);
        add(b);
        System.out.println(b.intValue());
    }
    public static void add(B m){
        int t=777;
        m.setIntValue(t);
    }
}
class B{
    int n;
    B(int n){
        this.n=n;
    }
    public void setIntValue(int n){
        this.n=n;
    }
    int intValue(){
        return n;
    }
}

```

777

7. 分析程序，给出输出结果

```

class B{
    int n;
    static int sum=0;
}

```



```

        void setN(int n){
            this.n=n;
        }
        int getSum(){
            for(int i=1;i<=n;i++)
                sum=sum+i;
            return sum;
        }
    }
    public class A{
        public static void main(String args[]){
            B b1=new B(),b2=new B();
            b1.setN(3);
            b2.setN(5);
            int s1=b1.getSum();
            int s2=b2.getSum();
            System.out.println(s1+s2);
        }
    }

```

27

8. 程序改错：将下面给出的程序修改成正确的代码

```

class A{
    int x=100;
    static long y;
    y=200;
    public void f(){
        y=300;
    }
    public static void g(){
        x=-23;
    }
}

```

```

class A{
    static int x=100;
    static long y;
    //y=200;
    public static void f(){
        y=300;
    }
    public static void g(){
        x=-23;
    }
}

```

```
    }  
}
```

9. 程序改错：将下面给出的程序修改成正确的代码

```
class Tom{  
    private int x=120;  
    protected int y=20;  
    int z=11;  
    private void f(){  
        x=200;  
        System.out.println(x);  
    }  
    void g(){  
        x=200;  
        System.out.println(x);  
    }  
}  
public class A{  
    public static void main(String args[]){  
        Tom tom=new Tom();  
        tom.x=22;  
        tom.y=33;  
        tom.z=55;  
        tom.f();  
        tom.g();  
    }  
}
```

```
class Tom{  
    int x=120;  
    int y=20;  
    int z=11;  
    void f(){  
        x=200;  
        System.out.println(x);  
    }  
    void g(){  
        x=200;  
        System.out.println(x);  
    }  
}  
public class A{  
    public static void main(String args[]){
```

```
    Tom tom=new Tom();  
    tom.x=22;  
    tom.y=33;  
    tom.z=55;  
    tom.f();  
    tom.g();  
}  
}
```