# JAVA编程进阶上机报告

****

**学 院 智能与计算**

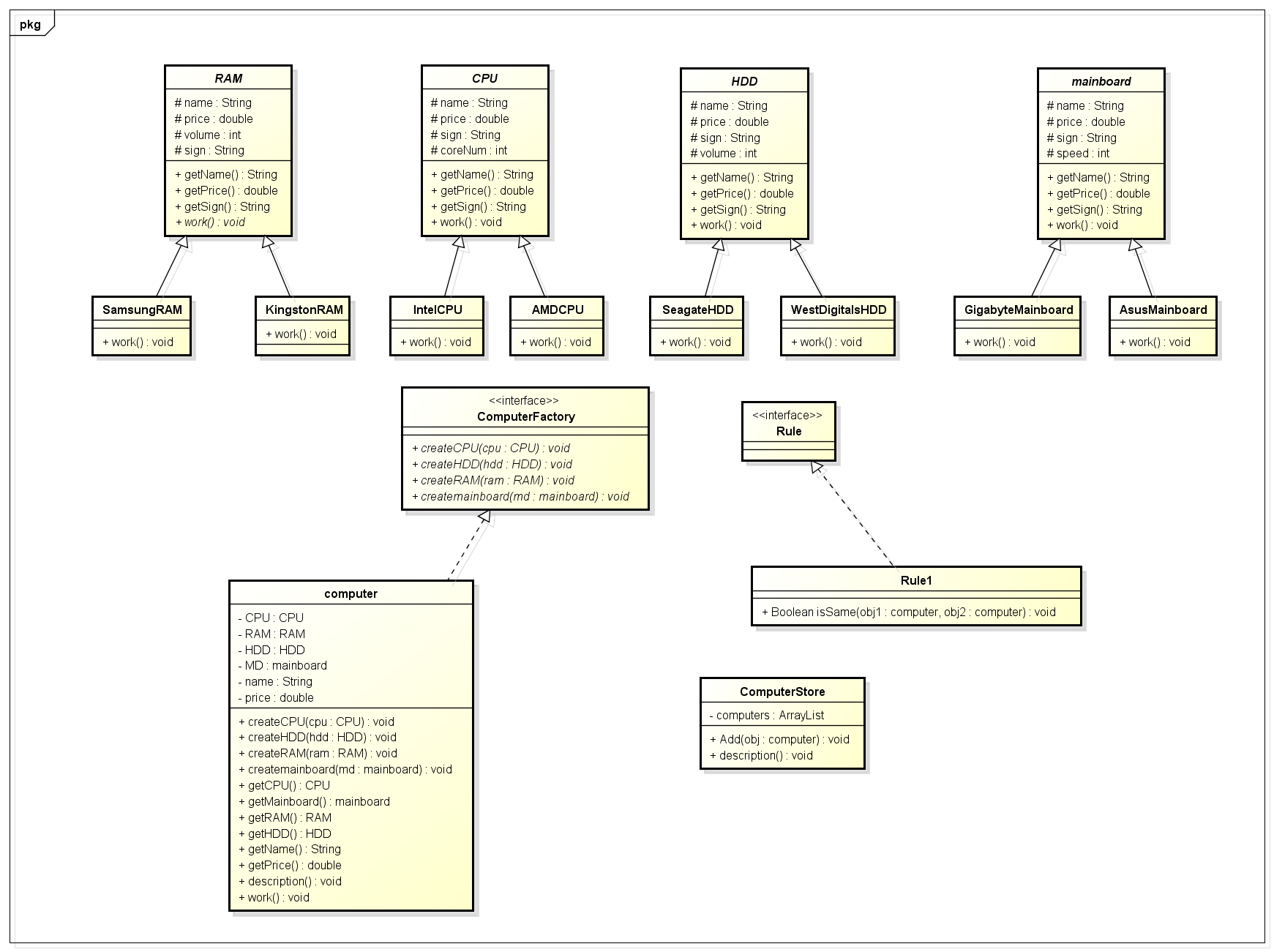
**专 业 软件工程**

**班 级 三班**

**学 号 3018216144**

**姓 名 王文君**

1. **实验要求**
2. 针对每个组件的每个品牌，设计一个类，并画成整体的类图
3. 设计计算机类（Computer.java），由上述四类组件组装而成，包括计算机的名称、计算机的描述（包括各个组件名）以及总价格等
4. 设计计算机销售主类（ComputerStore.java），包括3个由不同组件组装在一起的计算机实例，可实现计算机商品一览表，可展示每台计算机的描述、价格、工作等。
5. 设计时基于抽象类和接口，要尽可能的实现高内聚、低耦合。
6. **源代码**



# 思路：建了四个抽象类，再分别针对每个组件的每个品牌，设计一个类，各自实现work方法。建了接口ComputerFactory,用来组装电脑。建了一个接口Rule，表示设定的一些规则，类Rule1，确定两个电脑配置是否完全相同，最后ComputerStore实现要求功能

**代码：**

**package** MyComputer;

**public** **abstract** **class** HDD {

**protected** String name="HDD";

**protected** **double** price;

**protected** String sign;

**protected** **int** volume;

**public** **double** getPrice()

{

**return** price;

}

**public** String getName()

{

**return** name;

}

**public** String getSign()

{

**return** sign;

}

**public** **abstract** **void** work();

}

**package** MyComputer;

**public** **abstract** **class** mainboard {

**protected** String name="mainboard";

**protected** **double** price;

**protected** String sign;

**protected** **int** speed;

**public** String getName()

{

**return** name;

}

**public** String getSign()

{

**return** sign;

}

**public** **double** getPrice()

{

**return** price;

}

**public** **abstract** **void** work();

}

**package** MyComputer;

**public** **abstract** **class** CPU {

**protected** String name="CPU";

**protected** **double** price;

**protected** **int** coreNum;

**protected** String sign;

**public** **double** getPrice()

{

**return** price;

}

**public** String getName()

{

**return** name;

}

**public** String getSign()

{

**return** sign;

}

**public** **abstract** **void** work();

}

**package** MyComputer;

**public** **abstract** **class** RAM {

**protected** String name="RAM";

**protected** **double** price;

**protected** **int** volume;

**protected** String sign;

**public** **double** getPrice()

{

**return** price;

}

**public** String getName()

{

**return** name;

}

**public** String getSign()

{

**return** sign;

}

**public** **abstract** **void** work();

}

**package** MyComputer;

**public** **class** AMDCPU **extends** CPU {

AMDCPU(**double** price,**int** coreNum)

{

**this**.coreNum=coreNum;

**this**.price=price;

**this**.sign="AMD";

}

**public** **void** work()

{

System.***out***.println(**this**.getName()+"work");

}

}

**package** MyComputer;

**public** **class** AsusMainboard **extends** mainboard{

AsusMainboard(**double** price,**int** speed)

{

**this**.speed=speed;

**this**.price=price;

**this**.sign="Asus";

}

**public** **void** work()

{

System.***out***.println(**this**.getName()+"work");

}

}

**package** MyComputer;

**public** **class** GigabyteMainboard **extends** mainboard {

GigabyteMainboard(**double** price,**int** speed)

{

**this**.speed=speed;

**this**.price=price;

**this**.sign="Gigabyte";

}

**public** **void** work()

{

System.***out***.println(**this**.getName()+"work");

}

}

**package** MyComputer;

**public** **class** IntelCPU **extends** CPU{

IntelCPU(**double** price,**int** coreNum)

{

**this**.coreNum=coreNum;

**this**.price=price;

**this**.sign="Intel";

}

**public** **void** work()

{

System.***out***.println(**this**.getName()+"work");

}

}

**package** MyComputer;

**public** **class** KingstonRAM **extends** RAM {

KingstonRAM (**double** price,**int** volume)

{

**this**.volume=volume;;

**this**.price=price;

**this**.sign="Kingston";

}

**public** **void** work()

{

System.***out***.println(name+"work");

}

}

**package** MyComputer;

**public** **class** SamsungRAM **extends** RAM{

SamsungRAM (**double** price,**int** volume)

{

**this**.volume=volume;;

**this**.price=price;

**this**.sign="Samsung";

}

**public** **void** work()

{

System.***out***.println(**this**.getName()+"work");

}

}

**package** MyComputer;

**public** **class** SeagateHDD **extends** HDD{

SeagateHDD (**double** price,**int** volume)

{

**this**.volume=volume;;

**this**.price=price;

**this**.sign="SeagateHDD";

}

**public** **void** work()

{

System.***out***.println(**this**.getName()+"work");

}

}

**package** MyComputer;

**public** **class** WestDigitalsHDD **extends** HDD{

WestDigitalsHDD (**double** price,**int** volume)

{

**this**.volume=volume;;

**this**.price=price;

**this**.sign="WestDigitals";

}

**public** **void** work()

{

System.out.println(**this**.getName()+"work");

}

}

**package** MyComputer;

**public** **interface** ComputerFactory {

**public** **void** createCPU(CPU cpu);

**public** **void** createHDD(HDD hdd);

**public** **void** createRAM(RAM ram);

**public** **void** createmainboard(mainboard md);

}

**package** MyComputer;

**public** **class** computer **implements** ComputerFactory{

**private** CPU CPU ;

**private** RAM RAM;

**private** mainboard MD;

**private** HDD HDD;

**private** String Name;

**private** **double** price;

computer(String name)

{

Name=name;

price=0;

}

**public** **void** createCPU(CPU cpu) {

// **TODO** Auto-generated method stub

CPU=cpu;

price+=cpu.getPrice();

}

@Override

**public** **void** createRAM(RAM ram) {

// **TODO** Auto-generated method stub

RAM=ram;

price+=RAM.getPrice();

}

@Override

**public** **void** createmainboard(mainboard md) {

// **TODO** Auto-generated method stub

MD=md;

price+=MD.getPrice();

}

@Override

**public** **void** createHDD(HDD hdd) {

// **TODO** Auto-generated method stub

HDD=hdd;

price+=HDD.getPrice();

}

**public** CPU getCPU()

{

**return** CPU;

}

**public** mainboard getMainboard()

{

**return** MD;

}

**public** RAM getRAM()

{

**return** RAM;

}

**public** HDD getHDD()

{

**return** HDD;

}

**public** String getName() {

**return** Name;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** description()

{

System.***out***.println("NAME:"+Name);

System.***out***.println("RAM:"+RAM.getSign());

System.***out***.println("HDD:"+HDD.getSign());

System.***out***.println("mainboard:"+MD.getSign());

System.***out***.println("CPU:"+CPU.getSign());

}

**public** **void** work()

{

RAM.work();

HDD.work();

MD.work();

CPU.work();

}

}

**package** MyComputer;

**public** **interface** Rule {

}

**package** MyComputer;

**public** **class** Rule1 **implements** Rule{

**public** Boolean isSame(computer obj1, computer obj2) {

// **TODO** Auto-generated method stub

**if**(obj1.getCPU().getSign().equals(obj2.getCPU().getSign())

&&obj1.getHDD().getSign().equals(obj2.getHDD().getSign())

&&obj1.getMainboard().getSign().equals(obj2.getMainboard().getSign())

&&obj1.getRAM().getSign().equals(obj2.getRAM().getSign()))

**return** **true**;

**else** **return** **false**;

}

}

**package** MyComputer;

**import** java.util.ArrayList;

**public** **class** ComputerStore {

ArrayList computers;

ComputerStore ()

{

computers=**new** ArrayList();

}

**void** Add(computer obj) **throws** Exception

{

**if**(computers.size()==3)

{

**throw** **new** Exception("已有三台电脑");

}

**for**(**int** i=0;i<computers.size();i++)

{

computer computer=(computer) computers.get(i);

**if**(**new** Rule1().isSame(obj, computer))

{

**throw** **new** Exception("重复错误");

}

}

computers.add(obj);

}

**public** **void** description()

{

**for**(**int** i=0;i<computers.size();i++)

{

computer computer=(computer) computers.get(i);

computer.description();

System.***out***.println("Price:"+computer.getPrice());

computer.work();

System.***out***.println();

}

}

}

**package** MyComputer;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

computer computer1=**new** computer("1");

computer computer2=**new** computer("2");

computer computer3=**new** computer("3");

AsusMainboard AsusMainboard=**new** AsusMainboard(100,20);

AMDCPU AMDCPU =**new** AMDCPU(50,1);

IntelCPU IntelCPU =**new** IntelCPU(50,1);

SamsungRAM SamsungRAM =**new** SamsungRAM (200,30);

SeagateHDD SeagateHDD=**new** SeagateHDD(250,100);

WestDigitalsHDD WestDigitalsHDD=**new** WestDigitalsHDD(300,50);

computer1.createCPU(AMDCPU);

computer1.createHDD(SeagateHDD);

computer1.createmainboard(AsusMainboard);

computer1.createRAM(SamsungRAM);

computer2.createCPU(IntelCPU);

computer2.createHDD(SeagateHDD);

computer2.createmainboard(AsusMainboard);

computer2.createRAM(SamsungRAM);

computer3.createCPU(IntelCPU);

computer3.createHDD( WestDigitalsHDD);

computer3.createmainboard(AsusMainboard);

computer3.createRAM(SamsungRAM);

ComputerStore ComputerStore=**new** ComputerStore();

**try** {

ComputerStore.Add(computer1);

ComputerStore.Add(computer2);

ComputerStore.Add(computer3);

ComputerStore.description();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

1. **实验结果**

