JAVA 编程进阶上机报告

学　院　智能与计算学部

专　业　软件工程

班　级　二班

学　号　3018216084

姓　名　田家硕

一、实验要求

某计算机组装公司主要销售各类组装计算机，计算机一般由CPU、内存、主板、硬盘等组件构成。具体组件信息如下：

组件名	组件品牌	组件属性
CPU	Intel、AMD	Name，coreNum，price
内存	Samsung, Kingston	Name, volume, price
硬盘	Seagate, WestDigitals	Name, volume, price
主板	Asus、Gigabyte	Name，speed, price

每个组件都有自己的工作方式，简单起见，每个组件的工作内容为打印“组件名+work”。

具体要求：

1) 针对每个组件的每个品牌，设计一个类，并画成整体的类图

2) 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

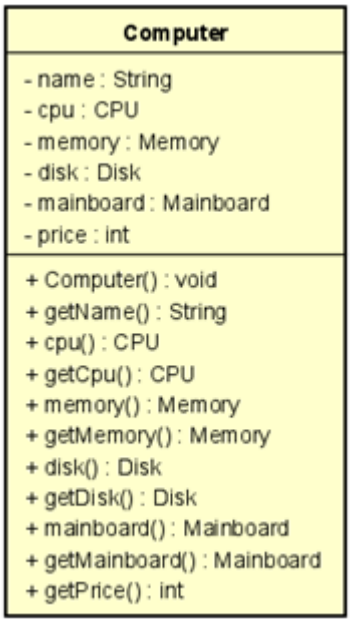
3) 设计计算机销售主类（ComputerStore.java），包括3个由不同组件组装在一起的计算机实例，可实现计算机商品一览表，可展示每台计算机的描述、价格、工作等。

4) 设计时基于抽象类和接口，要尽可能的实现高内聚、低耦合。

二、源代码

computer

computer

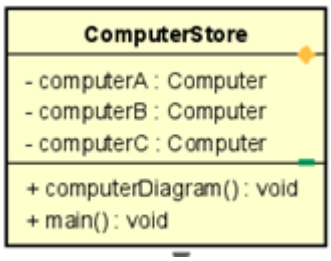



```
package computer;

import element.*;

public class Computer {
    private String name;
    private CPU cpu;
    private Memory memory;
    private Disk disk;
    private Mainboard mainboard;
    private int price;

    Computer(String name,String cpu,String memory,String disk,String
mainboard){
        this.name=name;
        this.cpu=cpu(cpu);
        this.disk=disk(disk);
        this.memory=memory(memory);
        this.mainboard=mainboard(mainboard);

 this.price=this.cpu.getPrice()+this.disk.getPrice()+this.memory.getPrice(
)+this.mainboard.getPrice();
    }

    public String getName() {
        return name;
```

```
    }

    public void setName(String name) {
        this.name = name;
    }

    public CPU cpu(String cpu) {
        if(cpu.equals("intel")){
            return new IntelCPU();
        }
        else if(cpu.equals("AMD")){
            return new AmdCPU();
        }
        else{
            return null;
        }
    }

    public CPU getCpu() {
        return cpu;
    }

    public Memory memory(String memory){
        if(memory.equals("Kingston")){
            return new KingstonMemory();
        }
        else if(memory.equals("Samsung")){
            return new SamsungMemory();
        }
        else{
            return null;
        }
    }

    public Memory getMemory() {
        return memory;
    }

    public Disk disk(String disk){
        if(disk.equals("Seagate")){
```

```
            return new SeagateDisk();
        }
        else if(disk.equals("WestDigitals")){
            return new WestDigitalsDisk();
        }
        else{
            return null;
        }
    }

    public Disk getDisk() {
        return disk;
    }

    public Mainboard mainboard(String mainboard){
        if(mainboard.equals("Gigabyte")){
            return new GigabyteMainboard();
        }
        else if(mainboard.equals("Asus")){
            return new AsusMainboard();
        }
        else{
            return null;
        }
    }

    public Mainboard getMainboard() {
        return mainboard;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public int getPrice() {
        return price;
    }


}
```

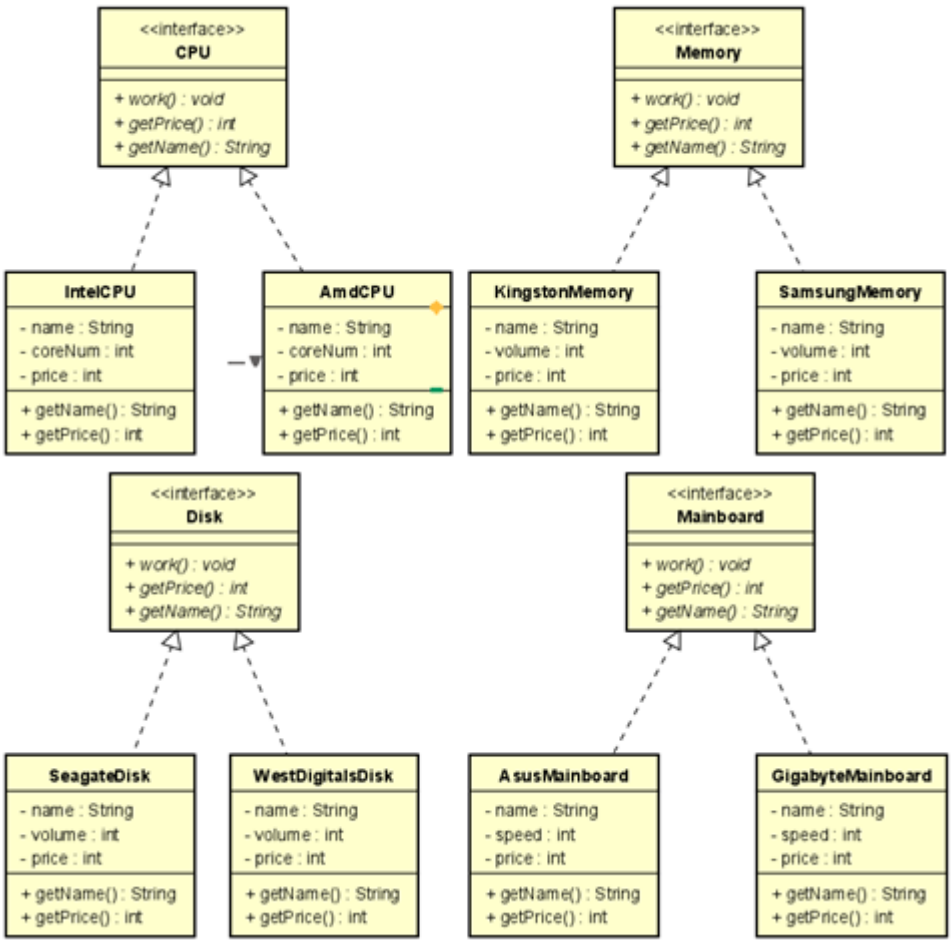
computerStore




```
package computer;

public class ComputerStore {
    private static Computer computerA=new
Computer("A","intel","Kingston","Seagate","Gigabyte");
    private static Computer computerB=new
Computer("B","AMD","Samsung","WestDigitals","Gigabyte");
    private static Computer computerC=new
Computer("C","intel","Samsung","Seagate","Asus");;

    public static void computerDiagram(Computer computer){
        System.out.println(computer.getName());
        System.out.println("CPU"+":"+computer.getCpu().getName()+" ");
        computer.getCpu().work();
        System.out.println("Memory:"+computer.getMemory().getName()+" ");
        computer.getMemory().work();
        System.out.println("Disk"+":"+computer.getDisk().getName()+" ");
        computer.getDisk().work();

 System.out.println("Mainboard"+":"+computer.getMainboard().getName()+"
");
        computer.getMainboard().work();
        System.out.println("price:"+computer.getPrice()+"\n");
    }

    public static void main(String[] args) {
        computerDiagram(computerA);
        computerDiagram(computerB);
        computerDiagram(computerC);
    }
```

```
}
```

element



CPU

```
package element;

public interface CPU {
    default void work(){
        System.out.println("CPU work");
    }

    int getPrice();
    String getName();
}
```

AmdCPU

```
package element;

public class AmdCPU implements CPU {
    private static String name="AmdCPU";
    private static int coreNum=50;
    private static int price=50;

    public String getName(){
        return name;
    }
    public int getCoreNum(){
        return coreNum;
    }
    public void setCoreNum(int coreNum){
        this.coreNum=coreNum;
    }
    public int getPrice(){
        return price;
    }
    public void setPrice(int price){
        this.price=price;
    }
}
```

IntelCPU

```
package element;

public class IntelCPU implements CPU{
    private static String name="IntelCPU";
    private static int coreNum=100;
    private static int price=100;

    public String getName(){
        return name;
    }
    public int getCoreNum(){
```

```
        return coreNum;
    }
    public void setCoreNum(int coreNum){
        this.coreNum=coreNum;
    }
    public int getPrice(){
        return price;
    }
    public void setPrice(int price){
        this.price=price;
    }
}
```

Disk

```
package element;

public interface Disk {
    default void work(){
        System.out.println("Disk work");
    }

    int getPrice();
    String getName();
}
```

SeagateDisk

```
package element;

public class SeagateDisk implements Disk {
    private String name = "SeagateDist";
    private int volume = 1000;
    private int price = 100;
    public String getName(){
        return name;
    }
    public int getVolume(){
```

```
        return volume;
    }
    public void setVolume(int volume){
        this.volume=volume;
    }
    public int getPrice(){
        return price;
    }
    public void setPrice(int price){
        this.price=price;
    }
}
```

WestDigitalsDisk

```
package element;

public class WestDigitalsDisk implements Disk {
    private String name = "SeagateDist";
    private int volume = 2000;
    private int price = 200;
    public String getName(){
        return name;
    }
    public int getVolume(){
        return volume;
    }
    public void setVolume(int volume){
        this.volume=volume;
    }
    public int getPrice(){
        return price;
    }
    public void setPrice(int price){
        this.price=price;
    }
}
```

Memory

```
package element;

public interface Memory {
    default void work(){
        System.out.println("Memory work");
    }

    int getPrice();
    String getName();
}
```

SamsungMemory

```
package element;

public class SamsungMemory implements Memory {
    private String name = "SamsungMemory";
    private int volume = 100;
    private int price = 100;
    public String getName(){
        return name;
    }
    public int getVolume(){
        return volume;
    }
    public void setVolume(int volume){
        this.volume=volume;
    }
    public int getPrice(){
        return price;
    }
    public void setPrice(int price){
        this.price=price;
    }
}
```

KingstonMemory

```
package element;
```

```
public class KingstonMemory implements Memory{
    private String name = "KingstonMemory";
    private int volume = 200;
    private int price = 200;
    public String getName(){
        return name;
    }
    public int getVolume(){
        return volume;
    }
    public void setVolume(int volume){
        this.volume=volume;
    }
    public int getPrice(){
        return price;
    }
    public void setPrice(int price){
        this.price=price;
    }
}
```

Mainboard

```
package element;

public interface Mainboard {
    default void work() {
        System.out.println("Mainboard work");
    }

    int getPrice();
    String getName();
}
```

AsusMainboard

```
package element;
```

```
public class AsusMainboard implements Mainboard {
    private String name = "AsusMainboard";
    private int speed = 1000;
    private int price = 100;
    public String getName(){
        return name;
    }
    public int getSpeed(){
        return speed;
    }
    public void setSpeed(int speed){
        this.speed=speed;
    }
    public int getPrice(){
        return price;
    }
    public void setPrice(int price){
        this.price=price;
    }
}
```

GigabyMainboard

```
package element;

public class GigabyteMainboard implements Mainboard {
    private String name = "GigabyteMainboard";
    private int speed = 2000;
    private int price = 200;
    public String getName(){
        return name;
    }
    public int getSpeed(){
        return speed;
    }
    public void setSpeed(int speed){
        this.speed=speed;
    }
    public int getPrice(){
```

```
        return price;
    }
    public void setPrice(int price){
        this.price=price;
    }
}
```

三、实验结果

运行ComputerStore输出：

```
A
CPU:IntelCPU
CPU work
Memory:KingstonMemory
Memory work
Disk:SeagateDist
Disk work
Mainboard:GigabyteMainboard
Mainboard work
price:600

B
CPU:AmdCPU
CPU work
Memory:SamsungMemory
Memory work
Disk:SeagateDist
Disk work
Mainboard:GigabyteMainboard
Mainboard work
price:550

C
CPU:IntelCPU
CPU work
Memory:SamsungMemory
Memory work
Disk:SeagateDist
```

```
Disk work
Mainboard:AsusMainboard
Mainboard work
price:400
```