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　6 班

学　号　3018216281

姓　名 朱明煊

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、整体类图与设计思想

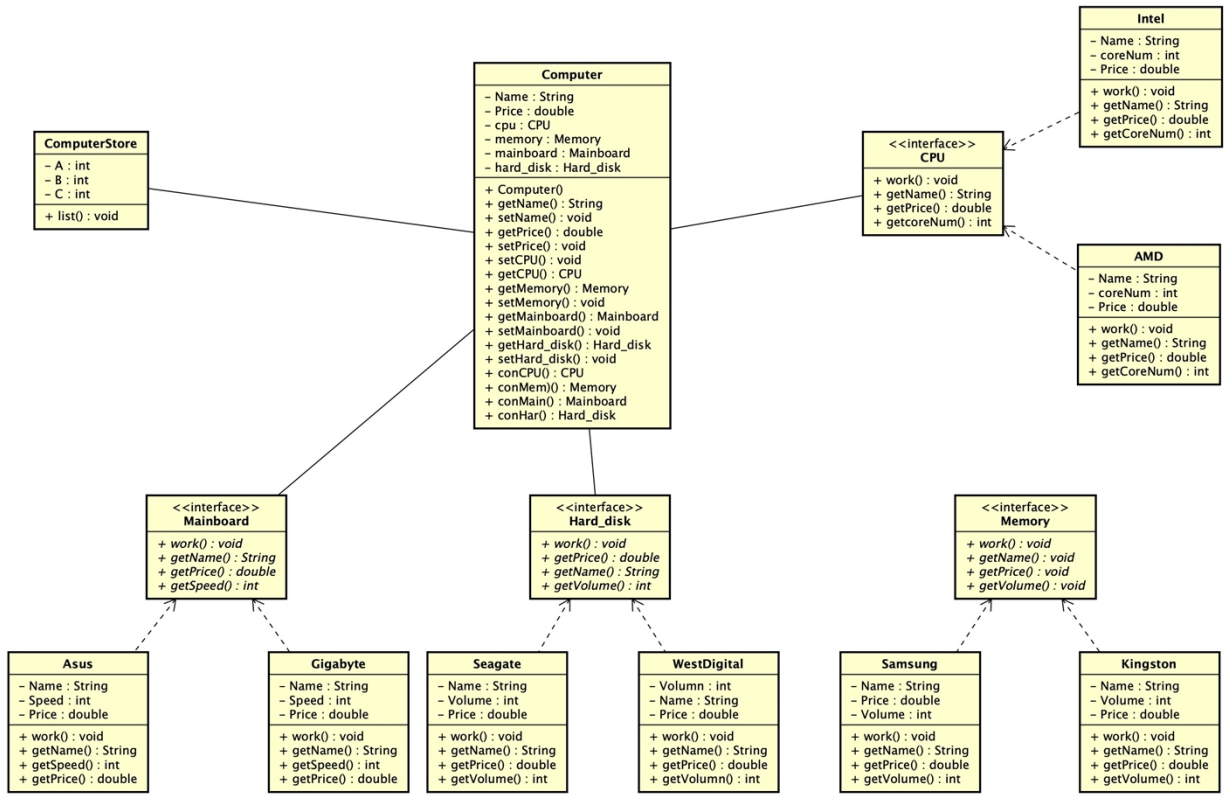
将每个组件作为接口，提供方法。而每个品牌的组件是接口方法的具体实现。包括 set、get、work 方法

当设计计算机类时：将名称、价格、四个组件作为属性。通过构造方法进行赋初值，访问等操作

实际销售类时，在其中实现三个计算机实例，并实现列表方法

类图

：



三、源代码

CPU 组件：

```
public interface CPU {
    public void work();
    public String getName();
    public double getPrice();
    public int getcoreNum();
}
```

Hard_disk 组件

```
public interface Hard_disk {
public void work();
public String getName();
public double getPrice();
public int getVolume();
```

```
}
```

Mainboard 组件

```
public interface Mainboard {
public void work();
public String getName();
public double getPrice();
public int getSpeed();
}
```

Memory 组件

```
public interface Memory {
public void work();
public String getName();
public double getPrice();
public int getVolume();
}
```

CPU 品牌类

```
public class AMD implements CPU {
    private String Name = "AMD";
    private int coreNum = 8;
    private double price = 1000;
@Override
public void work() {
// TODO Auto-generated method stub
System.out.println("AMD work");
}
@Override
public String getName() {
// TODO Auto-generated method stub
return Name;
}
@Override
public double getPrice() {
// TODO Auto-generated method stub
return price;
}
```

```
@Override
public int getcoreNum() {
// TODO Auto-generated method stub
return coreNum;
}
public void setCoreNum(int coreNum) {
this.coreNum = coreNum;
}
public void setName(String name) {
Name = name;
}
public void setPrice(double price) {
this.price = price;
}
}
```

CPU 品牌类

```
public class Intel implements CPU {
private String Name = "Intel";
    private int coreNum = 4;
    private double price = 600.5;
@Override
public void work() {
// TODO Auto-generated method stub
System.out.println("Intel work");
}
@Override
public String getName() {
// TODO Auto-generated method stub
return Name;
}
@Override
public double getPrice() {
// TODO Auto-generated method stub
return price;
}
@Override
public int getcoreNum() {
```

```
// TODO Auto-generated method stub
return coreNum;
}
public void setCoreNum(int coreNum) {
this.coreNum = coreNum;
}
public void setName(String name) {
Name = name;
}
public void setPrice(double price) {
this.price = price;
}

}
```

Memory 品牌类

```
public class Samsung implements Memory {
    private String Name="Samsung";
    private int volume = 4096;
    private double price = 500.56;
@Override
public void work() {
// TODO Auto-generated method stub
System.out.println("Samsung work");
}
@Override
public String getName() {
// TODO Auto-generated method stub
return Name;
}

@Override
public double getPrice() {
// TODO Auto-generated method stub
return price;
}

@Override
```

```
public int getVolume() {
// TODO Auto-generated method stub
return volume;
}
public void setName(String name) {
Name = name;
}
public void setVolume(int volume) {
this.volume = volume;
}
public void setPrice(double price) {
this.price = price;
}
}
```

Memory 品牌类

```
public class Kingston implements Memory{
private String Name="Kingston";
    private int volume = 2048;
    private double price = 400.52;
@Override
public void work() {
// TODO Auto-generated method stub
System.out.println("Kingston work");
}
@Override
public String getName() {
// TODO Auto-generated method stub
return Name;
}

@Override
public double getPrice() {
// TODO Auto-generated method stub
return price;
}

@Override
```

```
public int getVolume() {
// TODO Auto-generated method stub
return volume;
}
public void setName(String name) {
Name = name;
}
public void setVolume(int volume) {
this.volume = volume;
}
public void setPrice(double price) {
this.price = price;
}
}
```

Hard_disk 类

```
public class Seagate implements Hard_disk {
private String Name="Seagate";
    private int volume = 16;
    private double price = 250.5;
@Override
public void work() {
// TODO Auto-generated method stub
System.out.println("Seagate work");
}
@Override
public String getName() {
// TODO Auto-generated method stub
return Name;
}

@Override
public double getPrice() {
// TODO Auto-generated method stub
return price;
}

@Override
```

```
public int getVolume() {
// TODO Auto-generated method stub
return volume;
}
public void setName(String name) {
Name = name;
}
public void setVolume(int volume) {
this.volume = volume;
}
public void setPrice(double price) {
this.price = price;
}
}
```

Hard_disk 类:

```
public class WestDigitals implements Hard_disk {
private String Name="WestDigitals";
   private int volume = 32;
   private double price = 200.5;
@Override
public void work() {
// TODO Auto-generated method stub
System.out.println("WestDigitals work");
}
@Override
public String getName() {
// TODO Auto-generated method stub
return Name;
}

@Override
public double getPrice() {
// TODO Auto-generated method stub
return price;
}

@Override
```

```
public int getVolume() {
// TODO Auto-generated method stub
return volume;
}
public void setName(String name) {
Name = name;
}
public void setVolume(int volume) {
this.volume = volume;
}
public void setPrice(double price) {
this.price = price;
}
}
```

Mainboard 品牌类：

```
public class Asus implements Mainboard {
private String Name="Asus";
   private int speed = 144;
   private double price = 200.51;
@Override
public void work() {
// TODO Auto-generated method stub
System.out.println("Asus work");
}
@Override
public String getName() {
// TODO Auto-generated method stub
return Name;
}

@Override
public double getPrice() {
// TODO Auto-generated method stub
return price;
}

@Override
```

```
public int getSpeed() {
// TODO Auto-generated method stub
return speed;
}
public void setName(String name) {
Name = name;
}
public void setSpeed(int speed) {
this.speed = speed;
}
public void setPrice(double price) {
this.price = price;
}
}
```

Mainboard 品牌类：

```
public class Gigabyte implements Mainboard {
private String Name="Gigabyte";
   private int speed = 288;
   private double price = 400.11;
@Override
public void work() {
// TODO Auto-generated method stub
System.out.println("Gigabyte work");
}
@Override
public String getName() {
// TODO Auto-generated method stub
return Name;
}

@Override
public double getPrice() {
// TODO Auto-generated method stub
return price;
}

@Override
```

```
public int getSpeed() {
// TODO Auto-generated method stub
return speed;
}
public void setName(String name) {
Name = name;
}
public void setSpeed(int speed) {
this.speed = speed;
}
public void setPrice(double price) {
this.price = price;
}
}
```

Computer 类：

```
public class Computer {
   private String Name;
   private double Price;
   private CPU cpu;
   private Memory memory;
   private Mainboard mainboard;
   private Hard_disk hard_disk;
   public Computer(String name,String cpu,String memory,String hard_disk,String
mainboard) {
      this.Name=name;
      this.cpu=conCPU(cpu);
      this.memory=conMem(memory);
      this.mainboard=conMain(mainboard);
      this.hard_disk=conHar(hard_disk);
      this.Price=this.cpu.getPrice()+this.memory.getPrice()+this.hard_disk.getPrice()+
this.mainboard.getPrice();
   }
public String getName() {
return Name;
}
public void setName(String name) {
Name = name;
```

```
}
public double getPrice() {
return Price;
}
public void setPrice(double price) {
}
public CPU getCPU() {
return cpu;
}
public void setCPU(String name) {
if(name.equals("AMD")) {
   this.cpu = new AMD();
}
   else if(name.equals("Intel")) {
this.cpu = new Intel();
}
   else {
   this.cpu = null;
}
}
public Memory getMemory() {
return memory;
}
public void setMemory(String name) {
if(name.equals("Samsung")) {
   this.memory = new Samsung();
}
   else if(name.equals("Kingston")) {
this.memory = new Kingston();
}
   else {
   this.memory = null;
}
}
public Mainboard getMainboard() {
return mainboard;
}
public void setMainboard(String name) {
if(name.equals("Asus")) {
```

```
    this.mainboard = new Asus();
}
    else if(name.equals("Gigabyte")) {
this.mainboard = new Gigabyte();
}
    else {
    this.mainboard = null;
}
}
public Hard_disk getHard_disk() {
return hard_disk;
}
public void setHard_disk(String name) {
if(name.equals("Seagate")) {
    this.hard_disk = new Seagate();
}
    else if(name.equals("WestDigitals")) {
this.hard_disk = new WestDigitals();
}
    else {
    this.hard_disk = null;
}
}
public CPU conCPU(String name){
    if(name.equals("AMD")) {
    return new AMD();
}
    else if(name.equals("Intel")) {
return new Intel();
}
    else {
return null;
}
}
public Memory conMem(String name){
    if(name.equals("Samsung")) {
    return new Samsung();
}
    else if(name.equals("Kingston")) {
```

```
return new Kingston();
}
   else {
return null;
}
}
public Hard_disk conHar(String name){
   if(name.equals("Seagate")) {
   return new Seagate();
}
   else if(name.equals("WestDigitals")) {
return new WestDigitals();
}
   else {
return null;
}
}
public Mainboard conMain(String name){
   if(name.equals("Asus")) {
   return new Asus();
}
   else if(name.equals("Gigabyte")) {
return new Gigabyte();
}
   else {
return null;
}
}
}
```

ComputerStore 类：

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

```
    public static void list(Computer com) {
    System.out.println(com.getName()+" : ");
    System.out.println("Price is :"+ com.getPrice());
        System.out.println("CPU "+com.getCPU().getName()+" CoreNum
"+com.getCPU().getcoreNum()+" Price "+com.getCPU().getPrice());
        com.getCPU().work();
        System.out.println("Memory "+com.getMemory().getName()+" Volume
"+com.getMemory().getVolume()+" Price "+com.getMemory().getPrice());
        com.getMemory().work();
        System.out.println("Hard_disk "+com.getHard_disk().getName()+" Volume
"+com.getHard_disk().getVolume()+" Price "+com.getHard_disk().getPrice());
        com.getHard_disk().work();
        System.out.println("Mainboard "+com.getMainboard().getName()+" Speed
"+com.getMainboard().getSpeed()+" Price "+com.getMainboard().getPrice());
        com.getMainboard().work();
        System.out.println("\n");
    }
    public static void main(String[] args) {
    list(A);
    list(B);
    list(C);
    }
}
```

四、实验结果

打印计算机

```
<terminated> ComputerStore [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home/bin/java (2020年3月13日 下午10:56:31)
A :
Price is :2151.17
CPU AMD CoreNum 8 Price 1000.0
AMD work
Memory Samsung Volume 4096 Price 500.56
Samsung work
Hard_disk Seagate Volume 16 Price 250.5
Seagate work
Mainboard Gigabyte Speed 288 Price 400.11
Gigabyte work


B :
Price is :1402.03
CPU Intel CoreNum 4 Price 600.5
Intel work
Memory Kingston Volume 2048 Price 400.52
Kingston work
Hard_disk WestDigitals Volume 32 Price 200.5
WestDigitals work
Mainboard Asus Speed 144 Price 200.51
Asus work


C :
Price is :1701.67
CPU Intel CoreNum 4 Price 600.5
Intel work
Memory Samsung Volume 4096 Price 500.56
Samsung work
Hard_disk WestDigitals Volume 32 Price 200.5
WestDigitals work
Mainboard Gigabyte Speed 288 Price 400.11
Gigabyte work
```