计算机销售系统的设计

3018216095 郭紫珊 软工二班

1 实验要求

1.1 需求描述：

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”。

1.2 实现功能

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

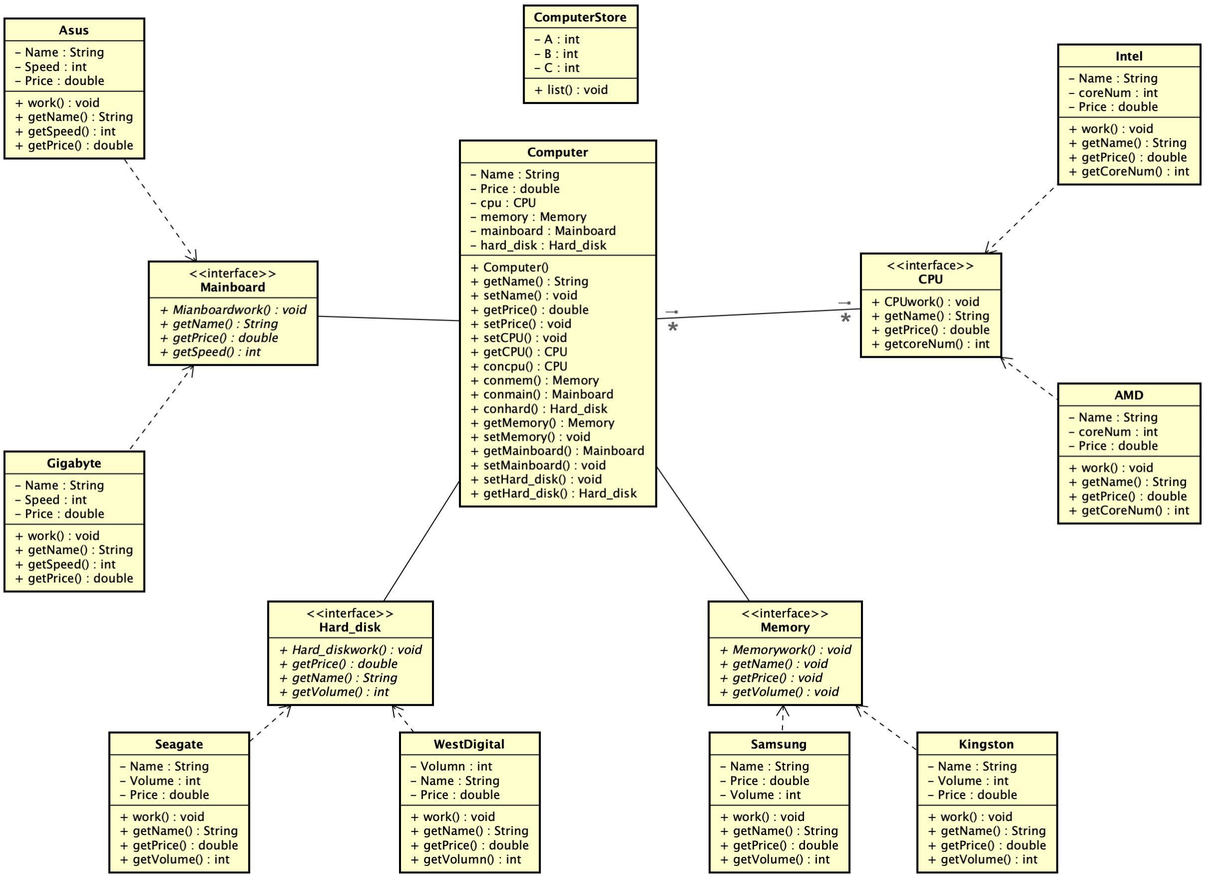
2)设计计算机类（Computer.java），由上述四类组件组装而成，包括计算机的名称、计算机的描述（包括各个组件名）以及总价格等

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。

2 设计思路和类图

类图：



设计思路：

1. 计算机类是由四个组件拼接而成的，一开始我想过把四个组件设计成接口，他们的品牌也是接口，接口继承接口。但是这样，计算机类就会继承八个接口，每个实例都要实现八个接口的方法。题目中要求组件品牌是任选的，所以显然这样的设计是不合理的。

2. 既然接口不合理，那么不如把这几个类变成计算机的属性。根据用户输入的品牌来确定 new 哪个品牌的对象，这样才能实现任意牌子的组件拼在一起。这样一来，这些牌子只能是一个个的类。

3. 而且每个牌子的名字其实都是固定好的，所以 name 这个属性最好在每个品牌类里面来实现，而每一个品牌类里面都有 work（）这个函数，所以各个组件最好设置为接口（他们没有自己的属性），然后将 work（）函数放在这些接口比如 CPU，Memory 等等里面。然后品牌类再去实现这个接口。比如，AMDCPU，IntelCPU 类去实现 CPU 接口，其他的组件类似。

4. 在计算机的构造函数里面，由顾客输入品牌来创造计算机实例：

```
 public Computer(String name,String cpu,String memory,String harddisk,String mainboard) {
        this.name=name;
        this.cpu=Wcpu(cpu);
        this.mainboard=Wmainboard(mainboard);
        this.memory=Wmemory(memory);
        this.harddisk=Wharddisk(harddisk);
      this.price=this.cpu.getPrice()+this.memory.getPrice()+this.harddisk.getPrice()+this.mainboard.getPrice();
 }
```

其中的 WCPU（），WMemory（）方法是根据名字判断 new 什么类的，具体如下：

```
public CPU Wcpu(String brand) {
        if(brand.equals("AMD")) {
                return new AMDCPU();
        }else if(brand.equals("Intel")) {
                return new IntelCPU();
        }else {
                return null;
        }
 }
```

5. 计算机销售类可以直接包含计算机类的三个实例，然后再写一个静态方法，打印出计算机实例的每个信息。这里由于要调用 getName（），getCoreNum（）等方法，所以也要把他们写在组件接口里。比如：

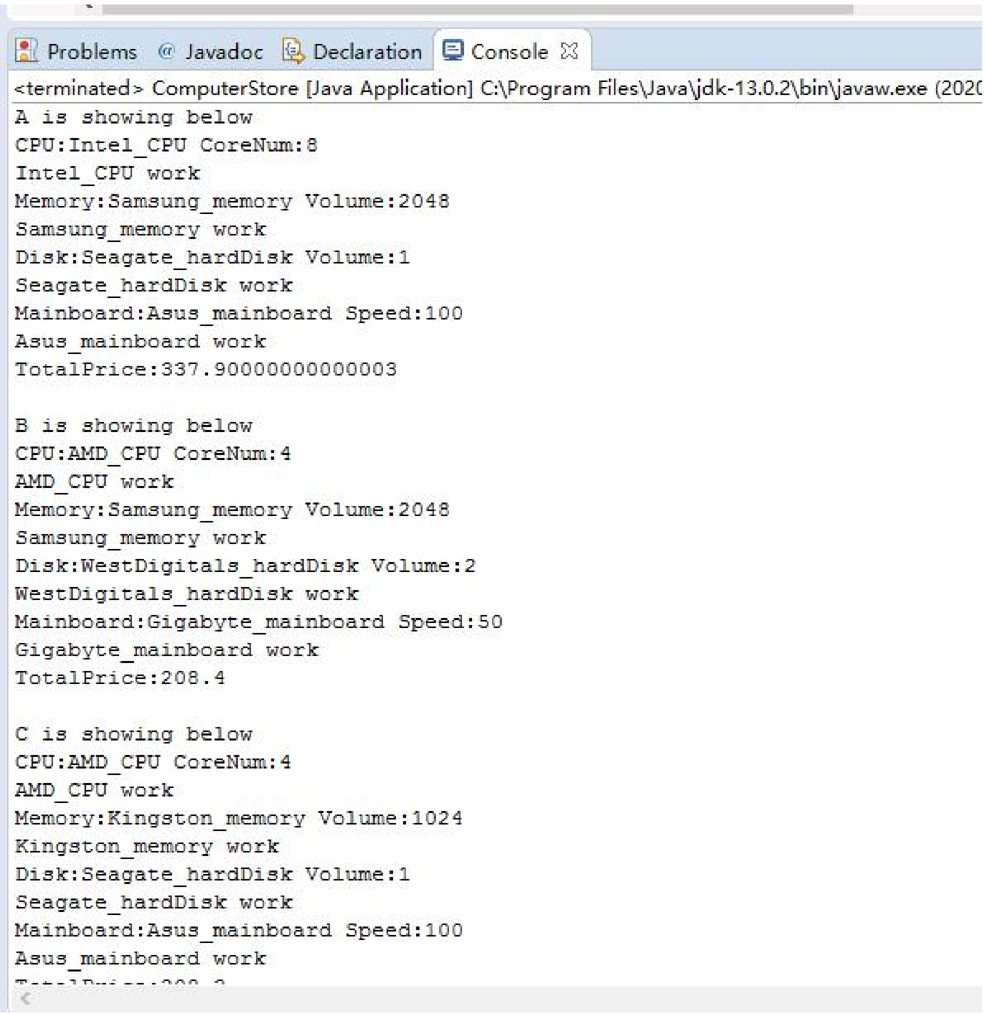
```
public interface CPU {
        public void CPUWork();
        public double getPrice();
        public String getName();
        public int getCoreNum();
}
```

再在 main 函数里面对三个实例执行静态方法，注意把三个实例也设置为 static。

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

以上就是我的设计思路。代码见附录。

3 运行结果



```
<terminated> ComputerStore [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (202
A is showing below
CPU:Intel_CPU CoreNum:8
Intel_CPU work
Memory:Samsung_memory Volume:2048
Samsung_memory work
Disk:Seagate_hardDisk Volume:1
Seagate_hardDisk work
Mainboard:Asus_mainboard Speed:100
Asus_mainboard work
TotalPrice:337.90000000000003

B is showing below
CPU:AMD_CPU CoreNum:4
AMD_CPU work
Memory:Samsung_memory Volume:2048
Samsung_memory work
Disk:WestDigitals_hardDisk Volume:2
WestDigitals_hardDisk work
Mainboard:Gigabyte_mainboard Speed:50
Gigabyte_mainboard work
TotalPrice:208.4

C is showing below
CPU:AMD_CPU CoreNum:4
AMD_CPU work
Memory:Kingston_memory Volume:1024
Kingston_memory work
Disk:Seagate_hardDisk Volume:1
Seagate_hardDisk work
Mainboard:Asus_mainboard Speed:100
Asus_mainboard work
TotalPrice:208.3
```

附录：

CPU.java:

```
public interface CPU {
    public void CPUWork();
    public double getPrice();
    public String getName();
    public int getCoreNum();
}
```

Mainboard.java

```
public interface mainboard {
    public void mainboardWork();
    public double getPrice();
    public String getName();
    public int getspeed();
}
```

Memory.java

```
public interface memory {
    public void memoryWork();
    public double getPrice();
    public String getName();
    public int getvolume();
}
```

HardDisk.java

```
public interface HardDisk {
    public void hardDiskWork();
    public double getPrice();
    public String getName();
    public int getvolume();
}
```

AMDCPU.java

```
public class AMDCPU implements CPU{
    private String Name="AMD_CPU";
    private int coreNum=4;
    private double price=100.5;
    public void CPUWork() {
        System.out.println("AMD_CPU work");
    };

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

IntelCPU.java

```
public class IntelCPU implements CPU{
    private String Name="Intel_CPU";
    private int coreNum=8;
    private double price=211.3;
    public void CPUWork() {
        System.out.println("Intel_CPU work");
    };

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

KingstonMemory.java

```
public class KingstonMemory implements memory{
    private String Name="Kingston_memory";
    private int volume=1024;
    private double price=35.8;
    public void memoryWork() {
        System.out.println("Kingston_memory work");
    };

    public String getName(){
        return Name;
    }
    public void setName(String Name) {
        this.Name=Name;
    }
    public int getvolume(){
        return volume;
    }
    public void setvolume(int volume){
        this.volume=volume;
    }
    public double getPrice(){
        return price;
    }
    public void setPrice(double price){
        this.price=price;
    }
}
```

SamsungMemory.java

```
public class SamsungMemory implements memory{
    private String Name="Samsung_memory";
    private int volume=2048;
    private double price=54.6;
    public void memoryWork() {
        System.out.println("Samsung_memory work");
    };

    public String getName(){
        return Name;
    }
    public void setName(String Name) {
```

```
		this.Name=Name;
	}
	public int getvolume(){
		return volume;
	}
	public void setvolume(int volume){
		this.volume=volume;
	}
	public double getPrice(){
		return price;
	}
	public void setPrice(double price){
		this.price=price;
	}
}
```

AsusMainboard.java

```
public class AsusMainboard implements mainboard{
	private String Name="Asus_mainboard";
	private int speed=100;
	private double price=54.8;
	public void mainboardWork() {
		System.out.println("Asus_mainboard work");
	};

	public String getName(){
		return Name;
	}
	public void setName(String Name) {
		this.Name=Name;
	}
	public int getspeed(){
		return speed;
	}
	public void setspeed(int speed){
		this.speed=speed;
	}
	public double getPrice(){
		return price;
	}
	public void setPrice(double price){
		this.price=price;
	}
}
```

GigabyteMainboard.java

```
public class GigabyteMainboard implements mainboard{
        private String Name="Gigabyte_mainboard";
        private int speed=50;
        private double price=25.4;
        public void mainboardWork() {
                System.out.println("Gigabyte_mainboard work");
        };

        public String getName(){
                return Name;
        }
        public void setName(String Name) {
                this.Name=Name;
        }
        public int getspeed(){
                return speed;
        }
        public void setspeed(int speed){
                this.speed=speed;
        }
        public double getPrice(){
                return price;
        }
        public void setPrice(double price){
                this.price=price;
        }
}
```

SeagateHardDisk.java

```
public class SeagateHardDisk implements HardDisk{
        private String Name="Seagate_hardDisk";
        private int volume=1;
        private double price=17.2;
        public void hardDiskWork() {
                System.out.println("Seagate_hardDisk work");
        };

        public String getName(){
                return Name;
        }
        public void setName(String Name) {
                this.Name=Name;
```

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

WestDigitalsHardDisk.java

```
public class WestDigitalsHardDisk implements HardDisk{
    private String Name="WestDigitals_hardDisk";
    private int volume=2;
    private double price=27.9;
    public void hardDiskWork() {
        System.out.println("WestDigitals_hardDisk work");
    };

    public String getName(){
        return Name;
    }
    public void setName(String Name) {
        this.Name=Name;
    }
    public int getvolume(){
        return volume;
    }
    public void setvolume(int volume){
        this.volume=volume;
    }
    public double getPrice(){
        return price;
    }
    public void setPrice(double price){
        this.price=price;
    }
}
```

Computer.java

```
public class Computer {
  private CPU cpu;
  private memory memory;
  private HardDisk harddisk;
  private mainboard mainboard;
  private String name;
  private double price;

  public Computer(String name,String cpu,String memory,String harddisk,String mainboard) {
      this.name=name;
      this.cpu=Wcpu(cpu);
      this.mainboard=Wmainboard(mainboard);
      this.memory=Wmemory(memory);
      this.harddisk=Wharddisk(harddisk);

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

  public CPU Wcpu(String brand) {
      if(brand.equals("AMD")) {
          return new AMDCPU();
      }else if(brand.equals("Intel")) {
          return new IntelCPU();
      }else {
          return null;
      }
  }
  public memory Wmemory(String brand) {
      if(brand.equals("Samsung")) {
          return new SamsungMemory();
      }else if(brand.equals("Kingston")) {
          return new KingstonMemory();
      }else {
          return null;
      }
  }
  public HardDisk Wharddisk(String brand) {
      if(brand.equals("Seagate")) {
          return new SeagateHardDisk();
      }else if(brand.equals("WestDigitals")) {
          return new WestDigitalsHardDisk();
      }else {
          return null;
      }
  }
  public mainboard Wmainboard(String brand) {
      if(brand.equals("Asus")) {
          return new AsusMainboard();
      }else if(brand.equals("Gigabyte")) {
          return new GigabyteMainboard();
      }else {
          return null;
      }
  }

  public CPU getCPU() {
      return this.cpu;
  }
  public void setCPU(String brand) {
      if(brand.equals("AMD")) {
          this.cpu = new AMDCPU();
      }else if(brand.equals("Intel")) {
          this.cpu = new IntelCPU();
      }else {
          System.out.println("InValid!");
      }
  }
  public memory getMemory() {
      return this.memory;
  }
  public void setMemory(String brand) {
      if(brand.equals("Samsung")) {
          this.memory = new SamsungMemory();
      }else if(brand.equals("Kingston")) {
          this.memory =    new KingstonMemory();
      }else {
          System.out.println("InValid!");
      }
  }
  public mainboard getMainboard() {
      return this.mainboard;
  }
  public void setMainboard(String brand) {
      if(brand.equals("Asus")) {
          this.mainboard = new AsusMainboard();
      }else if(brand.equals("Gigabyte")) {
```

```
			this.mainboard = new GigabyteMainboard();
		}else {
			System.out.println("InValid!");
		}
	}
	public HardDisk getHardDisk() {
		return this.harddisk;
	}
	public void setHardDisk(String brand) {
		if(brand.equals("Seagate")) {
			this.harddisk = new SeagateHardDisk();
		}else if(brand.equals("WestDigitals")) {
			this.harddisk = new WestDigitalsHardDisk();
		}else {
			System.out.println("InValid!");
		}
	}

	public double getPrice() {
		return this.price;
	}
	public void setPrice(double price) {
		this.price=price;
	}
	public String getName() {
		return this.name;
	}
	public void setName(String name) {
		this.name=name;
	}
}
```

ComputerStore.java

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

	public static void Show(Computer computer) {
		System.out.println(computer.getName()+" is showing below");
		System.out.println("CPU"+":"+computer.getCPU().getName()+"
CoreNum:"+computer.getCPU().getCoreNum());
        computer.getCPU().CPUWork();
        System.out.println("Memory:"+computer.getMemory().getName()+"
Volume:"+computer.getMemory().getvolume());
        computer.getMemory().memoryWork();
        System.out.println("Disk"+":"+computer.getHardDisk().getName()+"
Volume:"+computer.getHardDisk().getvolume());
        computer.getHardDisk().hardDiskWork();
        System.out.println("Mainboard"+":"+computer.getMainboard().getName()+"
Speed:"+computer.getMainboard().getspeed());
        computer.getMainboard().mainboardWork();
        System.out.println("TotalPrice:"+computer.getPrice()+"\n");
    }

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