



第一次上机作业

学 院: 智能与计算学部

专 业: 软件工程

姓 名: 王申宇

学 号: 3018216142

年 级: 2018

班 级: 3

## 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。”

需要自己设计出一个计算机，该计算机包括四大组成部分：CPU、内存、主板、硬盘，每一部分有自己的属性，每个部分又有自己的品牌。

## 二、源代码

com.shenyu.factory包

Factory.java

```
package com.shenyu.factory;

import com.shenyu.itemFactory.ComputerFactory;

public abstract class Factory {
    public static Factory getFactory(String factoryClassName) {
        Factory factory = new ComputerFactory();
        return factory;
    }

    public abstract CPU creatCPU(String name, int price, int coreNum, String brand);
    public abstract HD creatHD(String name, int price, int volume, String brand);
    public abstract Mainboard creatMainboard(String name, int price, int speed, String brand);
    public abstract Memory creatMemory(String name, int price, int volume, String brand);
    public abstract Computer creatComputer(String name);
}
```

ComputerItem.java

```
package com.shenyu.factory;

import com.shenyu.itemFactory.AMD;

public abstract class ComputerItem {

    protected String name; // 每个零件都要有的 name
    protected int price; // 每个零件都要有的 price
    protected String type; // 每个零件都要有一个 类型

    public ComputerItem(String name, int price, String type) {
        this.name = name;
        this.price = price;
        this.type = type;
    }
    // 每个零件都要工作
    public void work() {
        System.out.println(type + " work");
    }
}
```

Computer.java

```
package com.shenyu.factory;

import java.util.ArrayList;

public abstract class Computer {

    protected String name;
    protected String describe = "";
    protected int price = 0;

    protected ArrayList<ComputerItem> items = new ArrayList<>();

    public Computer(String name) {
        this.name = name;
    }

    public void add(ComputerItem item) {
        items.add(item);
        this.price += item.price;
        this.describe += item.type + ": " + item.getClass().getSimpleName() + " " +
item.name + "\r\n";
    }

    public abstract void work();

    public String getName() {
        return this.name;
    }

    public String getDescribe() {
        return describe;
    }

    public int getPrice() {
        return price;
    }
}
```

CPU.java

```
package com.shenyu.factory;
// 抽象表示 CPU 的类
public abstract class CPU extends ComputerItem {

    protected int coreNum;

    public CPU(String name, int price, int coreNum) {
        super(name, price, "CPU");
        this.coreNum = coreNum;
    }
}
```

HD.java

```
package com.shenyu.factory;

public abstract class HD extends ComputerItem {

    protected int volume;

    public HD(String name, int price, int volume) {
        super(name, price, "HD");
        this.volume = volume;
    }
}
```

Mainboard.java

```
package com.shenyu.factory;

public abstract class Mainboard extends ComputerItem {

    protected int speed;

    public Mainboard(String name, int price, int speed) {
        super(name, price, "Mainboard");
        this.speed = speed;
    }
}
```

Memory.java

```

package com.shenyu.factory;

public abstract class Memory extends ComputerItem {

    protected int volume;

    public Memory(String name, int price, int volume) {
        super(name, price, "Memory");
        this.volume = volume;
    }
}

```

com.shenyu.itemFactory包

ComputerFactory.java

```

package com.shenyu.itemFactory;

import com.shenyu.factory.*;

public class ComputerFactory extends Factory {

    @Override
    public CPU creatCPU(String name, int price, int coreNum, String brand) {
        if(brand.equals("Intel")) {
            return new Intel(name, price, coreNum);
        }else if(brand == "AMD") {
            return new AMD(name, price, coreNum);
        }
        return null;
    }

    @Override
    public HD creatHD(String name, int price, int volume, String brand) {
        if(brand.equals("Seagate")) {
            return new Seagate(name, price, volume);
        }else if(brand == "WestDigitals") {
            return new WestDigitals(name, price, volume);
        }
        return null;
    }

    @Override
    public Mainboard creatMainboard(String name, int price, int speed, String brand) {
        if(brand.equals("Asus")) {
            return new Asus(name, price, speed);
        }else if(brand == "Gigabyte") {
            return new Gigabyte(name, price, speed);
        }
    }
}

```

```

    }
    return null;
}

@Override
public Memory creatMemory(String name, int price, int volume, String brand) {
    if(brand.equals("Samsung")) {
        return new Samsung(name, price, volume);
    }else if(brand == "Kingston") {
        return new Kingston(name, price, volume);
    }
    return null;
}

@Override
public Computer creatComputer(String name) {
    return new TrueComputer(name);
}
}

```

AMD.java

```

package com.shenyu.itemFactory;

import com.shenyu.factory.CPU;

public class AMD extends CPU {
    public AMD(String name, int coreNum, int price) {
        super(name, price, coreNum);
    }
}

```

Asus.java

```

package com.shenyu.itemFactory;

import com.shenyu.factory.Mainboard;

public class Asus extends Mainboard{

    public Asus(String name, int price, int speed) {
        super(name, price, speed);
    }

}

```

Gigabyte.java

```
package com.shenyu.itemFactory;

import com.shenyu.factory.Mainboard;

public class Gigabyte extends Mainboard{

    public Gigabyte(String name, int price, int speed) {
        super(name, price, speed);
    }
}
```

Intel.java

```
package com.shenyu.itemFactory;

import com.shenyu.factory.*;

public class Intel extends CPU{

    public Intel(String name, int coreNum, int price) {
        super(name, price, coreNum);
    }

}
```

Kingston.java

```
package com.shenyu.itemFactory;

import com.shenyu.factory.Memory;

public class Kingston extends Memory {

    public Kingston(String name, int price, int volume) {
        super(name, price, volume);
    }

}
```

Samsung.java

```
package com.shenyu.itemFactory;

import com.shenyu.factory.Memory;

public class Samsung extends Memory{

    public Samsung(String name, int price, int volume) {
        super(name, price, volume);
    }

}
```

WestDigitals.java

```
package com.shenyu.itemFactory;

import com.shenyu.factory.HD;

public class WestDigitals extends HD {

    public WestDigitals(String name, int price, int volume) {
        super(name, price, volume);
    }

}
```

Seagate.java

```
package com.shenyu.itemFactory;

import com.shenyu.factory.HD;

public class Seagate extends HD {

    public Seagate(String name, int price, int volume) {
        super(name, price, volume);
    }

}
```

TrueComputer.java

```
package com.shenyu.itemFactory;

import java.util.Iterator;

import com.shenyu.factory.*;
```



```

public class TrueComputer extends Computer{

    public TrueComputer(String name) {
        super(name);
    }

    @Override
    public void work() {
        System.out.println("计算机开始工作: ");
        Iterator<ComputerItem> it = items.iterator();
        while(it.hasNext()) {
            ComputerItem item = (ComputerItem)it.next();
            item.work();
        }
    }

}

```

com.shenyu.store包

ComputerStore.java

```

package com.shenyu.store;

import com.shenyu.factory.*;
import com.shenyu.factory.Computer;
import com.shenyu.factory.HD;
import com.shenyu.factory.Memory;
import com.shenyu.itemFactory.Intel;
import com.shenyu.itemFactory.Samsung;
import com.shenyu.itemFactory.Seagate;

public class ComputerStore {

    private Computer computer1;
    private Computer computer2;
    private Computer computer3;

    public ComputerStore(Computer computer1, Computer computer2, Computer
computer3) {
        this.computer1 = computer1;
        this.computer2 = computer2;
        this.computer3 = computer3;
    }

    public void printGoods() {
        System.out.println("本店包含商品为: ");
        System.out.println(computer1.getName());
    }
}

```

```

        System.out.println(computer2.getName());
        System.out.println(computer3.getName());
    }

    public void printPrice() {
        System.out.println("本店商品价格为：");
        System.out.println("" + computer1.getName() + "的价格：" +
computer1.getPrice());
        System.out.println("" + computer2.getName() + "的价格：" +
computer2.getPrice());
        System.out.println("" + computer3.getName() + "的价格：" +
computer3.getPrice());
    }

    public void printDescribe() {
        System.out.println("商品描述为：");
        System.out.println("" + computer1.getName() + "的描述:");
        System.out.println(computer1.getDescribe());
        System.out.println("" + computer2.getName() + "的描述:");
        System.out.println(computer2.getDescribe());
        System.out.println("" + computer3.getName() + "的描述:");
        System.out.println(computer3.getDescribe());
    }

    public void work() {
        System.out.println("" + computer1.getName() + "开始工作:");
        computer1.work();

        System.out.println("" + computer2.getName() + "开始工作:");
        computer2.work();

        System.out.println("" + computer3.getName() + "开始工作:");
        computer3.work();
    }

    public static void main(String[] args) {
        Factory factory = Factory.getFactory("ComputerFactory");

        CPU cpu1 = factory.creatCPU("cpu_name1", 4, 2999, "Intel");
        Memory memory1 = factory.creatMemory("memory_name1", 123, 2000, "Samsung");
        HD hd1 = factory.creatHD("hd_name1", 300, 16, "Seagate");
        Mainboard mainboard1 = factory.creatMainboard("mainboard_name1", 123, 333,
"Asus");

        Computer computer1 = factory.creatComputer("暗影精灵");
        computer1.add(cpu1);
        computer1.add(memory1);
        computer1.add(hd1);
        computer1.add(mainboard1);
    }

```

```

    CPU cpu2 = factory.creatCPU("cpu_name2", 1000, 4000, "AMD");
    Memory memory2 = factory.creatMemory("memory_name2", 400, 2000,
"Kingston");
    HD hd2 = factory.creatHD("hd_name2", 300, 16, "Seagate");
    Mainboard mainboard2 = factory.creatMainboard("mainboard_name2", 129, 633,
"Asus");

    Computer computer2 = factory.creatComputer("神州");
    computer2.add(cpu2);
    computer2.add(memory2);
    computer2.add(hd2);
    computer2.add(mainboard2);

    CPU cpu3 = factory.creatCPU("cpu_name3", 3000, 4000, "AMD");
    Memory memory3 = factory.creatMemory("memory_name3", 1000, 2000,
"Kingston");
    HD hd3 = factory.creatHD("hd_name3", 1000, 16, "WestDigitals");
    Mainboard mainboard3 = factory.creatMainboard("mainboard_name3", 1000,
2000, "Gigabyte");

    Computer computer3 = factory.creatComputer("Mac");
    computer3.add(cpu3);
    computer3.add(memory3);
    computer3.add(hd3);
    computer3.add(mainboard3);

    ComputerStore store = new ComputerStore(computer1, computer2, computer3);

    store.printGoods();
    System.out.println("-----");
    store.work();
    System.out.println("-----");
    store.printPrice();
    System.out.println("-----");
    store.printDescribe();
}

}

```

### 三、运行结果

本店包含商品为：

暗影精灵

神州

Mac

-----  
暗影精灵开始工作：

计算机开始工作：

CPU work!

Memory work!

HD work!

Mainboard work!

神州开始工作：

计算机开始工作：

CPU work!

Memory work!

HD work!

Mainboard work!

Mac开始工作：

计算机开始工作：

CPU work!