**JAVA编程进阶上机报告**

****

第一次上机作业

**学 院\_\_\_\_智能与计算\_\_\_\_\_**

**专 业\_\_\_\_\_软件工程\_\_\_\_\_\_**

**姓 名\_\_\_\_\_\_ \_刘昕\_\_\_\_\_**

**学 号\_\_\_\_\_3018216232\_\_\_\_**

**年 级\_\_\_\_\_\_\_2018\_\_ \_**

**班 级\_\_\_\_\_\_ 五班\_\_\_\_\_\_\_\_**

一，实验要求

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. **实现功能：**

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，UML类图（在附件中）

三，源代码

package java\_19202;

public abstract class CPU {

private String Name;

private String coreNum;

private int price;

public String getName() {

return Name;

}

public void setName(String name) {

Name = name;

}

public String getCoreNum() {

return coreNum;

}

public void setCoreNum(String coreNum) {

this.coreNum = coreNum;

}

public int getPrice() {

return price;

}

public void setPrice(int price) {

this.price = price;

}

}

package java\_19202;

public class AMDCPU extends CPU {

AMDCPU(){

this.setName("AMDCPU");

this.setCoreNum("8th");

this.setPrice(100);

}

}

package java\_19202;

public class intelCPU extends CPU {

intelCPU(){

this.setName("intel");

this.setCoreNum("8th");

this.setPrice(200);

}

}

package java\_19202;

public abstract class ROM {

private String Name;

private String volume;

private int price;

public String getName() {

return Name;

}

public void setName(String name) {

Name = name;

}

public String getVolume() {

return volume;

}

public void setVolume(String volume) {

this.volume = volume;

}

public int getPrice() {

return price;

}

public void setPrice(int price) {

this.price = price;

}

}

package java\_19202;

public class KingstonROM extends ROM{

KingstonROM(){

this.setName(" Kingston");

this.setVolume("A");

this.setPrice(500);

}

}

package java\_19202;

public class SamsungROM extends ROM{

SamsungROM(){

this.setName("SamsungROM");

this.setVolume("B");

this.setPrice(100);

}

}

package java\_19202;

public abstract class Disk {

private String Name;

private String volume;

private int price;

public String getName() {

return Name;

}

public void setName(String name) {

Name = name;

}

public String getVolume() {

return volume;

}

public void setVolume(String volume) {

this.volume = volume;

}

public int getPrice() {

return price;

}

public void setPrice(int price) {

this.price = price;

}

}

package java\_19202;

public class WestDigitalsDisk extends Disk{

WestDigitalsDisk(){

this.setName("WestDigitals");

this.setVolume("D");

this.setPrice(120);

}

}

package java\_19202;

public class SeagateDisk extends Disk {

SeagateDisk(){

this.setName("Seagate");

this.setVolume("C");

this.setPrice(166);

}

}

package java\_19202;

public abstract class Board {

private String Name;

private String speed;

private int price;

public String getSpeed() {

return speed;

}

public void setSpeed(String speed) {

this.speed = speed;

}

public String getName() {

return Name;

}

public void setName(String name) {

Name = name;

}

public int getPrice() {

return price;

}

public void setPrice(int price) {

this.price = price;

}

} package java\_19202;

public class AsusBoard extends Board{

AsusBoard(){

this.setName("Asus");

this.setSpeed("1000");

this.setPrice(300);

}

}

package java\_19202;

public class GigabyteBoard extends Board{

GigabyteBoard(){

this.setName("Gigabyte");

this.setSpeed("2000");

this.setPrice(300);

}

}

package java\_19202;

public class Computer implements show {

private CPU cpu;

private ROM rom;

private Disk disk;

private Board board;

Computer(CPU cpu, ROM rom,Disk disk,Board board)

{

this.cpu = cpu;

this.rom = rom;

this.disk = disk;

this.board = board;

}

public CPU getCpu() {

return cpu;

}

public void setCpu(CPU cpu) {

this.cpu = cpu;

}

public ROM getRom() {

return rom;

}

public void setRom(ROM rom) {

this.rom = rom;

}

public Disk getDisk() {

return disk;

}

public void setDisk(Disk disk) {

this.disk = disk;

}

public Board getBoard() {

return board;

}

public void setBoard(Board board) {

this.board = board;

}

@Override

public void declare() {

System.out.print(this.getCpu().getName());

System.out.println(this.getCpu().getCoreNum());

System.out.print(this.getRom().getName());

System.out.println(this.getRom().getVolume());

System.out.print(this.getDisk().getName());

System.out.println(this.getDisk().getVolume());

System.out.print(this.getBoard().getName());

System.out.println(this.getBoard().getSpeed());

}

@Override

public void cost() {

System.out.println(this.getBoard().getPrice()+this.getCpu().getPrice()+this.getDisk().getPrice()+this.getRom().getPrice());

}

}

package java\_19201;

public interface show {

public void declare();

public void cost();

}

package java\_19202;

public class ComputerStore {

public static void main(String arg0[]) {

CPU cpu1 = new intelCPU();

CPU cpu2 = new AMDCPU();

ROM rom1 = new SamsungROM();

ROM rom2 = new KingstonROM();

Disk disk1 = new SeagateDisk();

Disk disk2 = new WestDigitalsDisk();

Board board = new AsusBoard();

Computer c1 = new Computer(cpu1,rom1,disk1,board);

Computer c2 = new Computer(cpu2,rom2,disk2,board);

Computer c3 = new Computer(cpu1,rom2,disk1,board);

c1.cost();

c2.cost();

c3.cost();

c1.declare();

c2.declare();

c3.declare();

四，运行结果

766

1020

1166

Intel 8th

SamsungROM B

Seagate C

Asus 1000

AMDCPU 8th

Kingston A

WestDigitals D

Asus 1000

Intel 8th

Kingston A

Seagate C

Asus 1000