Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №6

По дисциплине: «Современные системы программирования»

Тема: «Проектирование классов»

Выполнил:

Студент 4 курса

Группы ИИ-16 (2)

Пешко А.С.

Проверил:

Коренкович О. Г.

Брест 2021

**Цель работы:** Получить основные понятия по следующим разделам:  
- объектно-ориентированное программирование;  
- создание объектов и классов из стандартной библиотеки (на примере Java);  
- создание собственных классов.

**Ход работы**

**Задание:**

Комплектующие PC.

1. Определить иерархию комплектующих PC.
2. Создать объекты комплектующих с различными характеристиками.
3. Собрать из комплектующих готовые PC.
4. Найти PC c наибольшей частотой процессора.

**Код программы:**

public class Main {  
  
 public static void main(String[] args) {  
  
 // состовляющие материнской платы  
 RAM ram = new RAM("DDR4", "64GB");  
 CPU cpui7 = new CPU("Intel core i7 920", 2660D, 64);  
 CPU cpui5 = new CPU("Intel core i5 10400", 2900D, 64);  
 HardDrive hardDisk = new HardDrive("250 MB/sec", 256D, 64);  
 GraphicsCard graphicsCard = new GraphicsCard("GeForce RTX 3060 Ti", "GA104", 1665);  
  
 // материнская плата c cpui5  
 Motherboard motherboard1 = new Motherboard("LGA 1200", cpui5, ram, hardDisk, graphicsCard);  
  
 // материнская плата c cpui7  
 Motherboard motherboard2 = new Motherboard("LGA 1200", cpui7, ram, hardDisk, graphicsCard);  
  
 // ноутбук с материнской платой 1  
 Laptop laptop1 = new Laptop(motherboard1);  
  
 // ноутбук с материнской платой 2  
 Laptop laptop2 = new Laptop(motherboard2);  
  
 List<Laptop> laptops = Arrays.asList(laptop1, laptop2);  
  
 Laptop laptopWithMaxCPUFrequency = laptops.stream().max(Laptop::compareCPUFrequency).get();  
  
 System.out.println(laptopWithMaxCPUFrequency);  
 }  
}

public class Laptop {  
  
 private Motherboard motherboard;  
  
 public Laptop() {  
 }  
  
 public Laptop(Motherboard motherboard) {  
 this.motherboard = motherboard;  
 }  
  
 public Motherboard getMotherboard() {  
 return motherboard;  
 }  
  
 public void setMotherboard(Motherboard motherboard) {  
 this.motherboard = motherboard;  
 }  
  
 @Override  
 public String toString() {  
 return "Computer{\n" +  
 "\tmotherboard=" + motherboard +  
 "\n}";  
 }  
  
 public static int compareCPUFrequency (Laptop l1, Laptop l2){  
 if(l1.getMotherboard().getCpu().getFrequencyHz() > l2.getMotherboard().getCpu().getFrequencyHz()) {  
 return 1;  
 }  
 return -1;  
 }  
}

public class Motherboard {  
 private String socket;  
 private CPU cpu;  
 private RAM ram;  
 private HardDrive hardDisk;  
 private GraphicsCard graphicsCard;  
  
 public Motherboard() {  
 }  
  
 public Motherboard(String socket, CPU cpu, RAM ram, HardDrive hardDisk, GraphicsCard graphicsCard) {  
 this.socket = socket;  
 this.cpu = cpu;  
 this.ram = ram;  
 this.hardDisk = hardDisk;  
 this.graphicsCard = graphicsCard;  
 }  
  
 public String getSocket() {  
 return socket;  
 }  
  
 public void setSocket(String socket) {  
 this.socket = socket;  
 }  
  
 public CPU getCpu() {  
 return cpu;  
 }  
  
 public void setCpu(CPU cpu) {  
 this.cpu = cpu;  
 }  
  
 public RAM getRam() {  
 return ram;  
 }  
  
 public void setRam(RAM ram) {  
 this.ram = ram;  
 }  
  
 public HardDrive getHardDisk() {  
 return hardDisk;  
 }  
  
 public void setHardDisk(HardDrive hardDisk) {  
 this.hardDisk = hardDisk;  
 }  
  
 public GraphicsCard getGraphicsCard() {  
 return graphicsCard;  
 }  
  
 public void setGraphicsCard(GraphicsCard graphicsCard) {  
 this.graphicsCard = graphicsCard;  
 }  
  
 @Override  
 public String toString() {  
 return "Motherboard{" +  
 "\n\tsocket='" + socket + '\'' +  
 ", \n\tcpu=" + cpu +  
 ", \n\tram=" + ram +  
 ", \n\thardDisk=" + hardDisk +  
 ", \n\tgraphicsCard=" + graphicsCard +  
 "\n}";  
 }  
}

public class CPU {  
  
 private String model;  
 private Double frequencyHz;  
 private Integer bitness;  
  
 public CPU() {  
 }  
  
 public CPU(String model, Double frequencyHz, Integer bitness) {  
 this.model = model;  
 this.frequencyHz = frequencyHz;  
 this.bitness = bitness;  
 }  
  
 public String getModel() {  
 return model;  
 }  
  
 public void setModel(String model) {  
 this.model = model;  
 }  
  
 public Double getFrequencyHz() {  
 return frequencyHz;  
 }  
  
 public void setFrequencyHz(Double frequencyHz) {  
 this.frequencyHz = frequencyHz;  
 }  
  
 public Integer getBitness() {  
 return bitness;  
 }  
  
 public void setBitness(Integer bitness) {  
 this.bitness = bitness;  
 }  
  
 @Override  
 public String toString() {  
 return "CPU{" +  
 "\n\tmodel='" + model + '\'' +  
 ", \n\tclockFrequency=" + frequencyHz +  
 ", \n\tbitness=" + bitness +  
 "\n}";  
 }  
}

public class HardDrive {  
  
 private String baudRate;  
 private Double gbCapacity;  
 private Integer bufferMbSize;  
  
 public HardDrive() {  
 }  
  
 public HardDrive(String baudRate, Double gbCapacity, Integer bufferMbSize) {  
 this.baudRate = baudRate;  
 this.gbCapacity = gbCapacity;  
 this.bufferMbSize = bufferMbSize;  
 }  
  
 public String getBaudRate() {  
 return baudRate;  
 }  
  
 public void setBaudRate(String baudRate) {  
 this.baudRate = baudRate;  
 }  
  
 public Double getGbCapacity() {  
 return gbCapacity;  
 }  
  
 public void setGbCapacity(Double gbCapacity) {  
 this.gbCapacity = gbCapacity;  
 }  
  
 public Integer getBufferMbSize() {  
 return bufferMbSize;  
 }  
  
 public void setBufferMbSize(Integer bufferMbSize) {  
 this.bufferMbSize = bufferMbSize;  
 }  
  
 @Override  
 public String toString() {  
 return "HardDisk{" +  
 "\n\tbaudRate='" + baudRate + '\'' +  
 ", \n\tgbCapacity=" + gbCapacity +  
 ", \n\tbufferMbSize=" + bufferMbSize +  
 "\n}";  
 }  
}

public class RAM {  
  
 private String type;  
 private String volume;  
  
 public RAM() {  
 }  
  
 public RAM(String type, String volume) {  
 this.type = type;  
 this.volume = volume;  
 }  
  
 public String getType() {  
 return type;  
 }  
  
 public void setType(String type) {  
 this.type = type;  
 }  
  
 public String getVolume() {  
 return volume;  
 }  
  
 public void setVolume(String volume) {  
 this.volume = volume;  
 }  
  
 @Override  
 public String toString() {  
 return "RAM{" +  
 "\n\ttype='" + type + '\'' +  
 ", \n\tvolume='" + volume + '\'' +  
 "\n}";  
 }  
}

**Результат работы программы:**



**Вывод:** получили основные понятия по объектно-ориентированному программированию; создали объекты и классы из стандартной библиотеки (Java); а также собственные классы.