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

Журавлёв В.А.

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 Computer {

private Motherboard motherboard;

public Computer() {

}

public Computer(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 class GraphicsCard {

private String model;

private String core;

private Integer coreFrequency;

public GraphicsCard() {

}

public GraphicsCard(String model, String core, Integer coreFrequency) {

this.model = model;

this.core = core;

this.coreFrequency = coreFrequency;

}

public String getModel() {

return model;

}

public void setModel(String model) {

this.model = model;

}

public String getCore() {

return core;

}

public void setCore(String core) {

this.core = core;

}

public Integer getCoreFrequency() {

return coreFrequency;

}

public void setCoreFrequency(Integer coreFrequency) {

this.coreFrequency = coreFrequency;

}

@Override

public String toString() {

return "GraphicsCard{" +

"\n\tmodel='" + model + '\'' +

", \n\tcore='" + core + '\'' +

", \n\tcoreFrequency=" + coreFrequency +

"\n}";

}

}

public class CPU {

private String model;

private Double clockFrequency;

private Integer bitness;

public CPU() {

}

public CPU(String model, Double clockFrequency, Integer bitness) {

this.model = model;

this.clockFrequency = clockFrequency;

this.bitness = bitness;

}

public String getModel() {

return model;

}

public void setModel(String model) {

this.model = model;

}

public Double getClockFrequency() {

return clockFrequency;

}

public void setClockFrequency(Double clockFrequency) {

this.clockFrequency = clockFrequency;

}

public Integer getBitness() {

return bitness;

}

public void setBitness(Integer bitness) {

this.bitness = bitness;

}

@Override

public String toString() {

return "CPU{" +

"\n\tmodel='" + model + '\'' +

", \n\tclockFrequency=" + clockFrequency +

", \n\tbitness=" + bitness +

"\n}";

}

}

public class HardDisk {

private String baudRate;

private Double gbCapacity;

private Integer bufferMbSize;

public HardDisk() {

}

public HardDisk(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 Motherboard {

private String socket;

private CPU cpu;

private RAM ram;

private HardDisk hardDisk;

private GraphicsCard graphicsCard;

public Motherboard() {

}

public Motherboard(String socket, CPU cpu, RAM ram, HardDisk 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 HardDisk getHardDisk() {

return hardDisk;

}

public void setHardDisk(HardDisk 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 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}";

}

}

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 + '\'' +

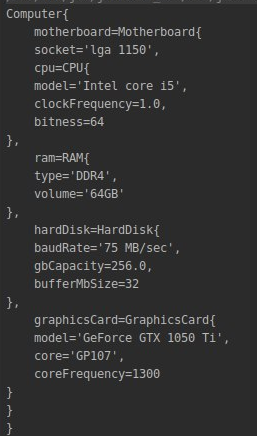
", \n\tvolume='" + volume + '\'' +

"\n}";

}

}

**Результат работы программы:**



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