**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 2**

по дисциплине:

«Визуальные средства разработки программных приложений»

на тему:

«Наследование в *Java*»

Выполнил: студент гр. ИТП-31

Коркуц С. И.

Принял: ассистент

Гуменников Е.Д.

Гомель 2020

**Цель работы:** изучить механизм наследования *Java*.

**Задание:** Вариант 7. Условия задания на рисунках 1 – 2.

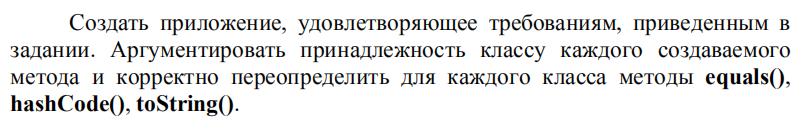


Рисунок 1 – Условие задания

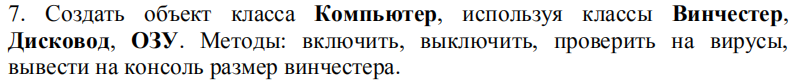


Рисунок 2 – Вариант задания

Структура созданных классов на рисунке 3.

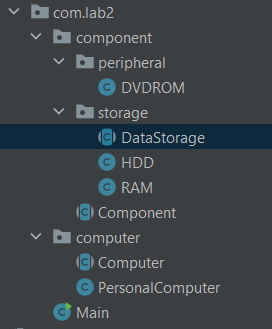


Рисунок 3 – Структура созданных классов

Изначально были созданы абстрактные классы для сущностей комплектующего и компьютера. Далее были созданы конкретные реализации этих классов. Компьютер выступает оболочкой для всех входящих в него компонентов. Были написаны методы для включения/выключения компьютера. Добавлены методы проверки на вирусы и получения размера диска.

В классе Main было написано заполнение базы тестовыми значениями, далее были написаны методы, решающие данные задания.

Результат выполнения задания представлен на рисунке 4.

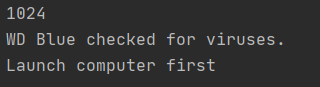


Рисунок 4 – Результат выполнения задания

**Вывод:** Были написаны абстрактные классы и их реализации соответственно поставленному заданию. Были написаны аксессоры и переопределения наследуемых методов.

**ПРИЛОЖЕНИЕ А**

Листинг созданных классов

package com.lab2.component;  
  
public abstract class Component {  
 protected String name;  
 protected String description;  
 protected double price;  
  
 public Component(String name, String description, double price) {  
 this.name = name;  
 this.description = description;  
 this.price = price;  
 }  
  
 @Override  
 public String toString() {  
 return String.format("Component\n%s\n%s\nPrice: %.2f", name, description, price);  
 }  
}

package com.lab2.component.storage;  
  
public class RAM extends DataStorage {  
 public enum RAMType {  
 DDR1, DDR2, DDR3, DDR4  
 }  
  
 RAMType type;  
  
 public RAM(String name, String description, double price, int sizeInGb, RAMType type) {  
 super(name, description, price, sizeInGb);  
 this.type = type;  
 }  
  
 @Override  
 public String toString() {  
 return String.format("RAM\n%s\n%s\nPrice: %.2f\nSize: %d\nType: %s", name, description, price, size, type.toString());  
 }  
}

package com.lab2.component.storage;  
  
public class HDD extends DataStorage {  
 int HDDSpeed;  
  
 public HDD(String name, String description, double price, int sizeInGb, int HDDSpeed) {  
 super(name, description, price, sizeInGb);  
 this.HDDSpeed = HDDSpeed;  
 }  
  
 @Override  
 public String toString() {  
 return String.format("RAM\n%s\n%s\nPrice: %.2f\nSize: %d\nSpeed: %s", name, description, price, size, HDDSpeed);  
 }  
}

package com.lab2.component.storage;  
  
import com.lab2.component.Component;  
  
public abstract class DataStorage extends Component {  
 int size;  
  
 public int getSize() {  
 return size;  
 }  
  
 public DataStorage(String name, String description, double price, int sizeInGb) {  
 super(name, description, price);  
 this.size = sizeInGb;  
 }  
  
 public void checkForViruses() {  
 System.out.println(name + " checked for viruses.");  
 }  
  
 @Override  
 public String toString() {  
 return String.format("DataStorage\n%s\n%s\nPrice: %.2f\nSize in GB: %d", name, description, price, size);  
 }  
}

package com.lab2.component.peripheral;  
  
import com.lab2.component.Component;  
  
public class DVDROM extends Component {  
 public DVDROM(String name, String description, double price) {  
 super(name, description, price);  
 }  
  
 @Override  
 public String toString() {  
 return String.format("DVDROM\n%s\n%s\nPrice: %.2f", name, description, price);  
 }  
}

package com.lab2.computer;  
  
import com.lab2.component.Component;  
  
import java.util.List;  
  
public abstract class Computer {  
 boolean isLaunched = false;  
 List<Component> components;  
  
 public Computer(List<Component> components) {  
 this.components = components;  
 }  
  
 public void launch() {  
 isLaunched = true;  
 }  
  
 public void switchOff() {  
 isLaunched = false;  
 }  
}

package com.lab2.computer;  
  
import com.lab2.component.Component;  
import com.lab2.component.storage.DataStorage;  
import com.lab2.component.storage.HDD;  
  
import java.util.List;  
  
public class PersonalComputer extends Computer {  
 public PersonalComputer(List<Component> components) {  
 super(components);  
 }  
  
 public void virusCheck() throws Exception {  
 if(isLaunched) {  
 for (Component component:  
 components) {  
 if(component instanceof HDD) {  
 ((HDD) component).checkForViruses();  
 }  
 }  
 }  
 else {  
 throw new Exception("Launch computer first");  
 }  
 }  
  
 public void getDisksSizes() throws Exception{  
 if(isLaunched) {  
 for (Component component:  
 components) {  
 if(component instanceof HDD) {  
 System.out.println(((HDD) component).getSize());  
 }  
 }  
 }  
 else {  
 throw new Exception("Launch computer first");  
 }  
 }  
}

package com.lab2;  
  
import com.lab2.component.Component;  
import com.lab2.component.peripheral.DVDROM;  
import com.lab2.component.storage.HDD;  
import com.lab2.component.storage.RAM;  
import com.lab2.computer.Computer;  
import com.lab2.computer.PersonalComputer;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Main {  
  
 public static void main(String[] args) {  
 List<Component> components = new ArrayList<>();  
 components.add(new DVDROM("DVDM11", "Simple dvd rom", 30));  
 components.add(new RAM("Kingston", "Powerfull ram", 200, 16, RAM.RAMType.DDR4));  
 components.add(new HDD("WD Blue", "Simple hdd", 100, 1024, 7200));  
 PersonalComputer computer = new PersonalComputer(components);  
 computer.launch();  
 try {  
 computer.getDisksSizes();  
 }  
 catch(Exception exception) {  
 System.out.println(exception.getMessage());  
 }  
 try {  
 computer.virusCheck();  
 }  
 catch(Exception exception) {  
 System.out.println(exception.getMessage());  
 }  
 computer.switchOff();  
 try {  
 computer.getDisksSizes();  
 }  
 catch(Exception exception) {  
 System.out.println(exception.getMessage());  
 }  
 }  
}