Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «Электронные вычислительные машины»

Отчет о лабораторной работе № 4

«Наследование и полиморфизм»

по дисциплине «Промышленное программирование»

**Выполнили:**

ст. гр. 145

бригада №10

Сеничкин Н.И.

Михеев В.Д.

**Проверил:**

Ст.пр. Бастрычкин А.С.

Асс. Трушина Е.А.

Рязань 2023

**Цель работы:** изучение принципов ООП, приобретение навыков использования наследования и полиморфизма в Java-программах.

**Задание** (вариант 10)**:**

1. Создать объект класса **Пианино**, используя классы **Клавиша, Педаль**. Методы: настроить, играть на пианино, нажимать клавишу.

2) **Мобильная связь**. Определить иерархию тарифов мобильной компании. Создать список тарифов компании. Подсчитать общую численность клиентов. Провести сортировку тарифов на основе размера абонентской платы. Найти тариф в компании, соответствующий заданному диапазону параметров.

**Ход работы:**

1) Код программы для задания 1:

Класс Key:

public class Key {  
 private boolean isPressed;  
  
 public Key() {  
 this.isPressed = false;  
 }  
  
 public boolean getKey(){  
 return this.isPressed;  
 }  
 public void press(int num) {  
 this.isPressed = true;  
 System.*out*.println("Клавиша "+num+" нажата");  
 }  
  
 public void release(int num) {  
 this.isPressed = false;  
 System.*out*.println("Клавиша "+num+" отпущена");  
 }  
}

Класс Pedal:

public class Pedal {  
 private boolean isPressed;  
  
 public Pedal() {  
 this.isPressed = false;  
 }  
 public boolean getPedal(){  
 return this.isPressed;  
 }  
 public void press() {  
 this.isPressed = true;  
 System.*out*.println("Педаль нажата");  
 }  
  
 public void release() {  
 this.isPressed = false;  
 System.*out*.println("Петаль отпущена");  
 }  
}

Класс Piano:

class Piano {  
 private boolean isTuned;  
 private boolean isPlayed;  
 private Key[] keys;  
 private Pedal pedal;  
  
 public Piano(int numberOfKeys) {  
 this.keys = new Key[numberOfKeys];  
 for (int i = 0; i < numberOfKeys; i++) {  
 this.keys[i] = new Key();  
 }  
 this.isPlayed = false;  
 this.isTuned = false;  
 this.pedal = new Pedal();  
 }  
  
 public void display(){  
 System.*out*.println();  
 for (int i = 0; i < this.keys.length; i++) {  
 System.*out*.println("Клавиша "+(i+1) +" "+this.keys[i].getKey());  
 }  
 System.*out*.println("Педаль "+this.pedal.getPedal()+"\n");  
 }  
  
 public void tune() {  
 this.isTuned = true;  
 System.*out*.println("Пианино настроено...");  
 }  
  
 public void startPlay() {  
 this.isPlayed = true;  
 System.*out*.println("Пианино играет...");  
 }  
  
 public void stopPlay(){  
 this.isPlayed = false;  
 System.*out*.println("Пианино завершило играть...");  
 }  
 public void pressKey(int keyNumber) {  
 if (keyNumber >= 0 && keyNumber < this.keys.length) {  
 this.keys[keyNumber].press(keyNumber);  
 } else {  
 System.*out*.println("Неверная клавиша");  
 }  
 }  
 public void releaseKey(int keyNumber){  
 if (keyNumber >= 0 && keyNumber < this.keys.length) {  
 this.keys[keyNumber].release(keyNumber);  
 } else {  
 System.*out*.println("Неверная клавиша");  
 }  
 }  
 public void pressPedal() {  
 this.pedal.press();  
 }  
 public void releasePedal(){  
 this.pedal.release();  
 }  
  
}

Класс Main:

public class Main {  
 public static void main(String[] args) {  
 Piano piano = new Piano(5);  
 piano.tune();  
 piano.startPlay();  
 piano.pressPedal();  
 piano.pressKey(2);  
 piano.pressKey(3);  
 piano.releaseKey(2);  
 piano.pressKey(4);  
 piano.releaseKey(3);  
 piano.releaseKey(4);  
 piano.releasePedal();  
 piano.display();  
 piano.stopPlay();  
 }  
}

На рисунке 1 представлен результат работы программы.

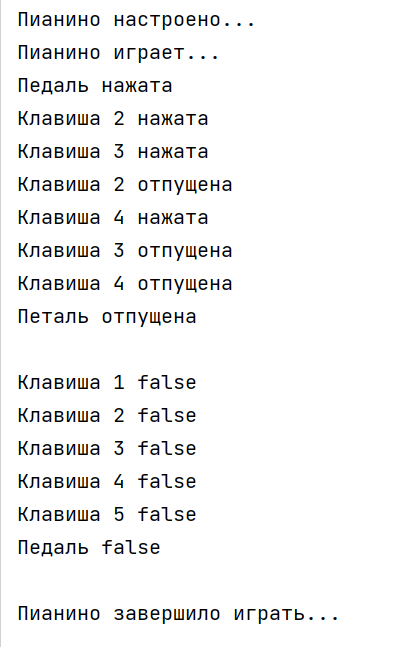


Рисунок 1 – Результат работы программы задания 1

2) Код программы для задания 2

Класс Person:

import java.util.Scanner;  
  
class Person {  
 private String name;  
 Person(){}  
 Person(String name){  
 this.name = name;  
 }  
 public Person setName(Person person){  
 Scanner in = new Scanner(System.in);  
 System.out.print("Введите имя: ");  
 person.name = in.next();  
 return person;  
 }  
 public String getName(){  
 return this.name;  
 }  
 public void display(){  
 System.out.println("Имя: "+name);  
 }  
}

Класс Rate:

import java.util.Scanner;  
import java.util.Collections;  
public class Rate {  
 int id;  
 private String rate;  
 private int price;  
 private int traffic;  
 private int minute;  
 private int sms;  
 Rate(){};  
 Rate(int id, String rate, int price, int traffic, int minute, int sms){  
 this.id = id;  
 this.rate = rate;  
 this.price = price;  
 this.traffic = traffic;  
 this.minute = minute;  
 this.sms = sms;  
 };  
 Rate(Rate tariff){  
 this.id = tariff.id;  
 this.rate = tariff.rate;  
 this.price = tariff.price;  
 this.traffic = tariff.traffic;  
 this.minute = tariff.minute;  
 this.sms = tariff.sms;  
 }  
 public void display(){  
 System.*out*.println( "id: "+this.id+"\n"+  
 "Тариф: "+this.rate+"\n"+  
 "Абоненская плата: "+this.price+"\n"+  
 "Пакет интернета: "+this.traffic+"\n"+  
 "Пакет минут: "+this.minute+"\n"+  
 "Пакет сообщений: "+this.sms+"\n");  
 };  
 public static void displayList(Rate[] tariff){  
 for (int i = 0; i < tariff.length; i++) {  
 System.*out*.println("id: " + tariff[i].id + "\n" +  
 "Тариф: " + tariff[i].rate + "\n" +  
 "Абоненская плата: " + tariff[i].price + "\n" +  
 "Пакет интернета: " + tariff[i].traffic + "\n" +  
 "Пакет минут: " + tariff[i].minute + "\n" +  
 "Пакет сообщений: " + tariff[i].sms + "\n");  
 }  
 };  
  
 public static Rate[] getRateStandart(){  
 Rate[] result;  
 int[] id = new int[] {1, 2, 3};  
 String[] rate = new String[] {"Стандарт", "Безлимит", "Тарифище"};  
 int[] price = new int[] {200, 500, 350};  
 int[] traffic = new int[] {5, 50, 30};  
 int[] minute = new int[] {100, 1000, 500};  
 int[] sms = new int[] {50, 500, 250};  
 result = new Rate[id.length];  
 for (int i = 0; i < id.length; i++) {  
 result[i] = new Rate(id[i], rate[i], price[i], traffic[i], minute[i], sms[i]);  
 }  
 return result;  
 }  
 public static void displayAllStandartRate(){  
 Rate[] rate;  
 rate = *getRateStandart*();  
 for (int i = 0; i < rate.length; i++) {  
 rate[i].display();  
 }  
 }  
 public static Rate setRateForClient(String name) {  
 Scanner in = new Scanner(System.*in*);  
 Rate[] tariff;  
 tariff = *getRateStandart*();  
 System.*out*.print("Введите id тарифа для пользователя "+name+": ");  
 int id = in.nextInt();  
 Rate result = new Rate(tariff[id-1]);  
 return result;  
 };  
  
 public static Rate[] sortRateByPrice(Rate[] tariff){  
 Rate t;  
 for (int i = 0; i < tariff.length-1; i++) {  
 for (int j = i; j < tariff.length; j++) {  
 if (tariff[i].price > tariff[j].price){  
 t = tariff[i];  
 tariff[i] = tariff[j];  
 tariff[j] = t;  
 }  
  
 }  
 }  
 return tariff;  
 };  
  
 public static Rate[] getTariffBySetting(Rate[] rate){  
 Scanner in = new Scanner(System.*in*);  
 Rate[] result;  
 System.*out*.print("Введите максимальную стоимость абонентской платы: ");  
 int price = in.nextInt();  
 System.*out*.print("Введите минмальный пакет интернета: ");  
 int traffic = in.nextInt();  
 System.*out*.print("Введите минмальный пакет минут: ");  
 int minute = in.nextInt();  
 System.*out*.print("Введите минмальный пакет смс: ");  
 int sms = in.nextInt();  
 int count = 0;  
 for (int i = 0; i < rate.length; i++) {  
 if (rate[i].price <= price && rate[i].traffic >= traffic &&  
 rate[i].minute >= minute && rate[i].sms >= sms){  
 count+=1;  
 }  
 }  
 result = new Rate[count];  
 int index = 0;  
 for (int i = 0; i < rate.length; i++) {  
 if (rate[i].price <= price && rate[i].traffic >= traffic &&  
 rate[i].minute >= minute && rate[i].sms >= sms){  
 result[index] = new Rate(rate[i]);  
 index++;  
 }  
 }  
  
 return result;  
 }  
}

Класс MobileCommunication:

import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
  
class MobileCommunication{  
 int countClient;  
 Person[] client;  
 Rate[] rate;  
 *//List<Integer> trafic; // Можно передлать, чтобы сюда сразу вставлялась вся инфа о тарифе. Поиск по id будет, вместо вставки самого id* MobileCommunication(){}  
 public List<Integer> setTrafic(List<Integer> trafic, String name){  
 Scanner in = new Scanner(System.in);  
 System.out.print("Введите id тарифа для "+name+": ");  
 trafic.add(in.nextInt());  
 return trafic;  
 }  
 public MobileCommunication setClient(MobileCommunication mobCom){  
 Scanner in = new Scanner(System.in);  
 System.out.print("Введите количество клиентов: ");  
 mobCom.countClient = in.nextInt();  
 mobCom.client = new Person[countClient];  
 for (int i = 0; i < mobCom.countClient; i++) {  
 mobCom.client[i] = new Person();  
 mobCom.client[i].setName(mobCom.client[i]);  
 }  
 mobCom.rate = new Rate[countClient];  
 System.*out*.println("Доступные тарифы:");  
 Rate.*displayAllStandartRate*();  
 for (int i = 0; i < mobCom.countClient; i++) {  
 mobCom.rate[i] = new Rate();  
 mobCom.rate[i] = Rate.*setRateForClient*(mobCom.client[i].getName());  
 }  
 return mobCom;  
 };  
  
 public void display(MobileCommunication mobCom){  
 for (int i = 0; i < mobCom.countClient; i++) {  
 mobCom.client[i].display();  
 mobCom.rate[i].display();  
 }  
 }  
 public int getCountClient(){  
 return this.client.length;  
 }  
}

Класс Main:

import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 *// Заполнение клиентской базы* MobileCommunication mobCom = new MobileCommunication();  
 mobCom.setClient(mobCom);  
 mobCom.display(mobCom);  
 int countClient = mobCom.countClient;  
 System.*out*.println("Количесво клиентов: "+countClient);  
  
 *// Сортировка тарифов по размеру абонентской платы* System.*out*.println("\n\nОтсортированные тарифы по размеру абонентской платы:");  
 Rate[] tariff;  
 tariff = Rate.*getRateStandart*();  
 tariff = Rate.*sortRateByPrice*(tariff);  
 Rate.*displayList*(tariff);  
  
 *// Тарифы, удоавлетворяющий параметрам* Rate[] result = Rate.*getTariffBySetting*(tariff);  
 System.*out*.println("Тарифы, удовлетворяющие требованиям:");  
 Rate.*displayList*(result);  
 }  
}

На рисунке 2-5 представлен результат работы программы задания 2

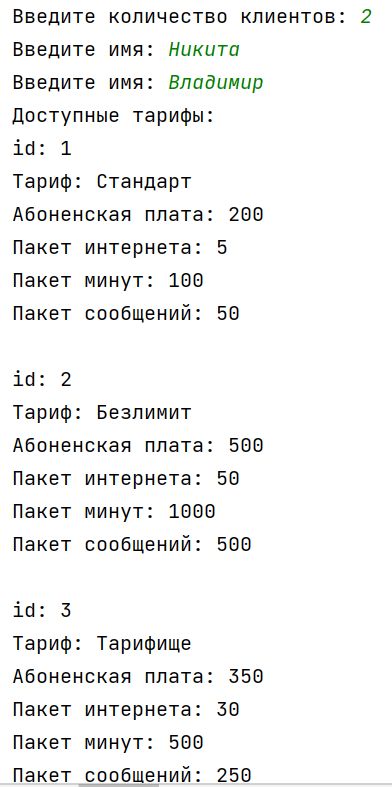


Рисунок 2 – Ввод клиентов и вывод доступных тарифов

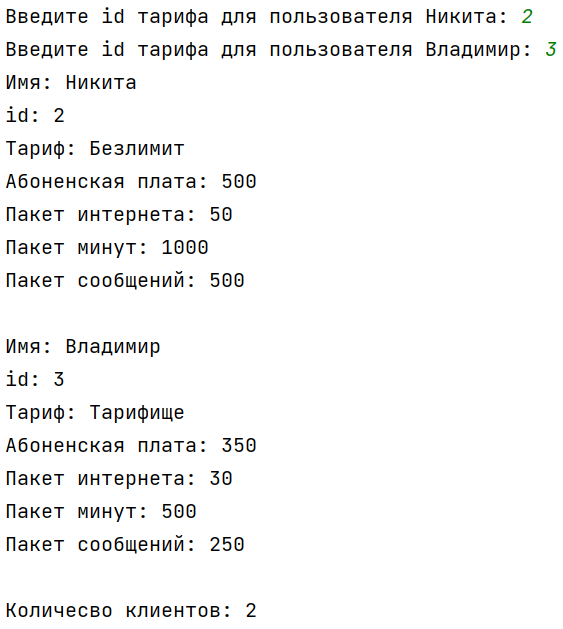
****

Рисунок 3 – Ввод тарифов клиентов и их вывод

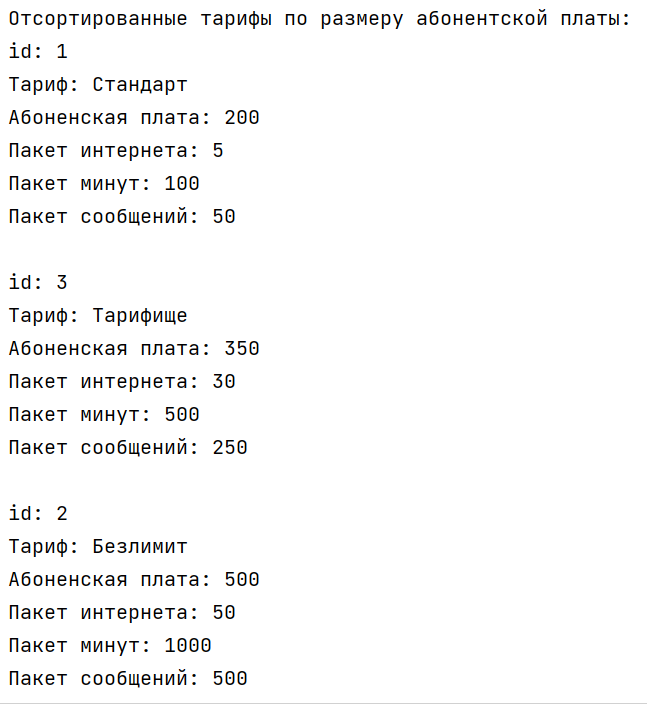
****

Рисунок 4 – Вывод отсортированных тарифов по размеру абонентской платы

****

Рисунок 5 – Вывод тарифа удовлетворяющего требования

**Заключение**

В ходе лабораторной работы были изучены принципы ООП, приобретены навыки работы с наследованием и полиморфизмом Java-программах, а также написаны 2 программы с их использованием.