Первый этап заданий берется из учебника «Java - Методы программирования». Выполняются задачи на языке Java в программе IntelliJ IDEA.

Первая задача находится на странице 50 (с главы B). Вариант номер 10.

Условие задачи: Вычислить норму матрицы.

Код задачи:

import java.util.Arrays;  
import java.util.Random;  
public class Main {  
final static Random rand = new Random();  
public static void main(String[] args) {  
// TODO Auto-generated method stub  
int [][]arr = createArray(7,8,-100,100);  
toConsole(arr);  
System.out.println("Norm ||A||L ="+maxRows(sumRows(arr)));  
System.out.println("Norm ||A||m ="+maxCols(sumCols(arr)));  
}  
public static int[][] createArray(int N,int M,int a,int b)  
{  
int [][]arr = new int[N][M];  
for(int i = 0;i<N;i++)  
{  
for(int j = 0;j<M;j++)  
{  
arr[i][j] = rand.nextInt((b - a) + 1) + a;  
}  
}  
//System.out.println(Arrays.deepToString(arr));  
return arr;  
  
}  
  
public static int[] sumRows(int[][]arr)  
{  
int []array = new int[arr.length];  
for(int i = 0;i<arr.length;i++)  
{  
for(int j = 0;j<arr[0].length;j++)  
{  
array[i]+=Math.abs(arr[i][j]);  
}  
}  
System.out.println(Arrays.toString(array));  
return array;  
  
}  
  
public static int[] sumCols(int [][]arr)  
{  
int[]array = new int[arr[0].length];  
for(int j = 0;j<arr[0].length;j++)  
{  
for(int i =0;i<arr.length;i++)  
{  
array[j]+=Math.abs(arr[i][j]);  
}  
}  
System.out.println(Arrays.toString(array));  
return array;  
  
}  
  
public static void toConsole(int [][]arr)  
{  
for (int[] ints : arr) {  
for (int anInt : ints) {  
System.out.print(anInt + " ");  
}  
System.out.println();  
}  
}  
  
public static int maxRows(int []arr)  
{  
int max = 0;  
for(int val:arr) {  
if(val > max)max = val;  
}  
return max;  
}  
  
public static int maxCols(int []arr)  
{  
int min = arr[0];  
for(int val:arr) {  
if(val < min) min=val;  
}  
return min;  
}  
}

Результат работы программы представлен ниже (Рисунок 1).

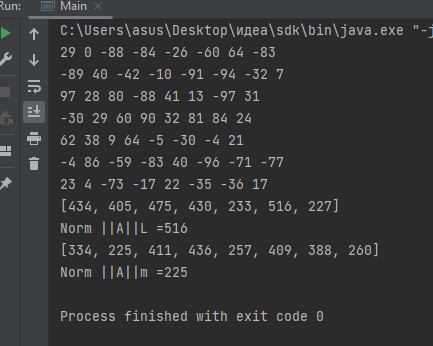


Рисунок 1 – Результат выполнения Задачи 1

Вторая задача на странице 89 вариант 10.

Условия задачи: Train: Пункт назначения, Номер поезда, Время отправления, Число мест (общих, купе, плацкарт, люкс). Создать массив объектов. Вывести:

a) список поездов, следующих до заданного пункта назначения;

b) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа;

c) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.

Код задачи:

import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Arrays;  
import java.util.Comparator;  
import java.util.Date;  
  
public class Train implements Comparable<Train> {  
private String lastStation;  
private String trainNumber;  
private Date departureTime;  
private static SimpleDateFormat dt = new SimpleDateFormat("HH:mm");  
  
public Train(String lastStation, String trainNumber, Date departureTime) {  
this.lastStation = lastStation;  
this.trainNumber = trainNumber;  
this.departureTime = departureTime;  
}  
  
public String getLastStation() {  
return lastStation;  
}  
  
public String getTrainNumber() {  
return trainNumber;  
}  
  
public Date getDepartureTime() {  
return departureTime;  
}  
  
public void setLastStation(String lastStation) {  
this.lastStation = lastStation;  
}  
  
public void setTrainNumber(String trainNumber) {  
this.trainNumber = trainNumber;  
}  
  
public void setDepartureTime(Date departureTime) {  
this.departureTime = departureTime;  
}  
private static void SortByNumber(Train[] trains){  
Arrays.sort(trains);  
}  
  
private static void SortByStation(Train[] trains){  
Arrays.sort(trains, Train.TrainStationComparator);  
}  
  
public String toString(){  
StringBuilder builder = new StringBuilder();  
builder.append("Train {");  
builder.append("To Station: ");  
builder.append(lastStation);  
builder.append(", Number: ");  
builder.append(trainNumber);  
builder.append(", time departure: ");  
builder.append(dt.format(departureTime));  
builder.append("}");  
return builder.toString();  
}  
public static String printQuestionInformation(Train[] trains, String trainNumber){  
for (Train train : trains)  
if (train.trainNumber.equalsIgnoreCase(trainNumber))  
return train.toString();  
return trainNumber;  
}  
  
public static Comparator<Train> TrainStationComparator = (t1, t2) -> {  
String station1 = t1.getLastStation().toUpperCase();  
String station2 = t2.getLastStation().toUpperCase();  
int result = station1.compareTo(station2);  
if (result == 0)  
return t1.departureTime.compareTo(t2.departureTime);  
return result;  
};  
  
@Override  
public int compareTo(Train o) {  
String compareNumber = ((Train)o).getTrainNumber();  
return this.trainNumber.compareToIgnoreCase(compareNumber);  
}  
  
public static void main(String[] args) throws ParseException {  
Train [] trains = new Train[5];  
trains[0] = new Train("Vienna", "001C", dt.parse("20:48"));  
trains[1] = new Train("Moscow", "030Y", dt.parse("12:00"));  
trains[2] = new Train("Minsk", "025B", dt.parse("01:44"));  
trains[3] = new Train("Warsaw", "013M", dt.parse("09:56"));  
trains[4] = new Train("Vienna", "296C", dt.parse("16:44"));  
  
SortByNumber(trains);  
for (Train train : trains)  
System.out.println(train.trainNumber);  
System.out.println();  
SortByStation(trains);  
for (Train train : trains)  
System.out.println(train.lastStation + " " + train.trainNumber + " " + dt.format(train.departureTime));  
  
String userQuestion = "001C";  
System.out.printf("%nTrain № %s ", userQuestion);  
System.out.println();  
System.out.println(printQuestionInformation(trains, userQuestion));  
  
}  
}

Результат выполнения представлен ниже (Рисунок 2).

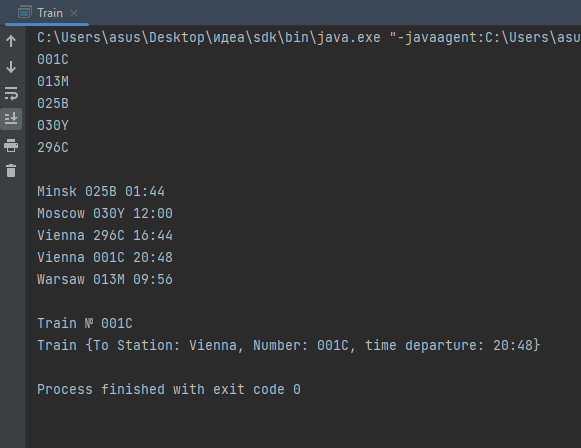


Рисунок 2 – Результат выполнения Задачи 2

Вторая задача на странице 125 вариант 10.

Условия задачи: создать объект класса Щенок, используя классы Животное, Собака. Методы: вывести на консоль имя, подать голос, прыгать, бегать, кусать.

Код задачи:

package zadanie3;  
public class Animal {  
protected String name;  
  
public Animal (String name) {  
this.name = name;  
}  
  
public String getName() {  
return "Бобик";  
}  
}  
package zadanie3;  
public class Dog extends Animal {  
public Dog(String name) {  
super (name);  
}  
  
public String votesCast(){  
return "Подать голос";  
}  
  
public String jump() {  
return "Прыгать";  
}  
  
public String run() {  
return "Бегать";  
}  
  
public String bite() {  
return "Кусать";  
}  
}  
package zadanie3;  
public class Puppy extends Dog {  
public Puppy(String name) {  
super(name);  
}  
  
}   
package zadanie3;  
public class PuppyRunner {  
public static void main (String[] args) {  
Puppy puppy = new Puppy("Ralf");  
  
System.out.println("Name: "+ puppy.getName());  
System.out.println("Votes Cast: " + puppy.votesCast());  
System.out.println("Jump: " + puppy.jump());  
System.out.println("Run: "+ puppy.run());  
System.out.println("Bite: "+ puppy.bite());  
}  
}

Результат работы представлен ниже (Рисунок 3).

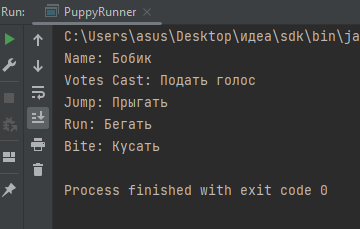


Рисунок 3 – Результат выполнения Задачи 3

Четвертая задача представлена на 145 странице вариант 10.

Условия задачи: создать класс Mobile с внутренним классом, с помощью объектов которого можно хранить информацию о моделях телефонов и их свойствах.

Код задачи:

package zadanie4;  
import java.util.Scanner;  
public class Mobile {  
public static Scanner ob=new Scanner(System.in);  
public info info;  
public String mobile;  
public Mobile(){  
System.out.print("Введите модель телефона: ");  
setMobile(ob.next());  
info=new info();  
}  
public String getMobile(){  
return mobile;  
}  
public void setMobile(String mobile){  
this.mobile=mobile;  
}  
public class info{  
int pr,st,sq;  
public info(){  
System.out.print("Оперативнная память (Гб):");  
setPr(ob.nextInt());  
System.out.print("Встроенная память (ГБ):");  
setSt(ob.nextInt());  
System.out.print("Камера (Мп):");  
setSq(ob.nextInt());  
}  
public int getPr(){  
return pr;  
}  
public void setPr(int pr){  
this.pr=pr;  
}  
public int getSt(){  
return st;  
}  
public void setSt(int st){  
this.st=st;  
}  
public int getSq(){  
return sq;  
}  
public void setSq(int sq){  
this.sq=sq;  
}  
public void print(){  
System.out.print("Модель телефона " +getMobile()+ ". Характеристики: Оперативная память (Гб) - " +getPr()+ "; Встроенная память (Гб) - " +getSt()+ "; Камера (Мп) - " +getSq());  
}  
}  
static public void main(String[]args){  
Mobile ct=new Mobile();  
ct.info.print();  
}  
}

Результат работы программы представлен ниже (Рисунок 4).

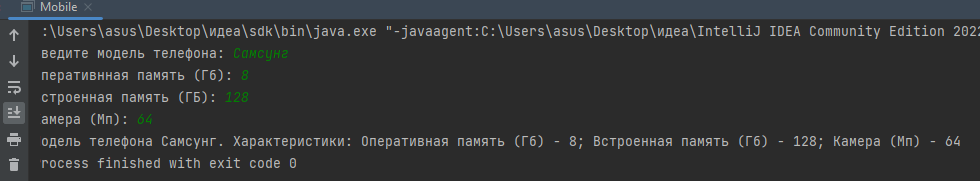


Рисунок 4 – Результат выполнения Задачи 4

Пятая задача представлена на странице 194, вариант 10.

Условие задачи: в тексте найти и напечатать n символов (и их количество), встречающихся наиболее часто.

Код задачи:

package zadanie5;  
  
import java.util.Scanner;  
import java.util.Map;  
import java.util.HashMap;  
import java.util.Collections;  
  
public class Main {  
public static void main(String[] args) {  
Scanner s = new Scanner(System.in);  
String line = s.nextLine();  
  
Map<Character, Integer> occurrences = new HashMap<>();  
  
for(char c : line.toCharArray()) {  
if(Character.isLetter(c)) {  
if(!occurrences.containsKey(c))  
occurrences.put(c, 1);  
else  
occurrences.put(c, occurrences.get(c) + 1);  
}  
}  
  
int maxValueInMap=(Collections.max(occurrences.values()));  
for (Map.Entry<Character, Integer> entry : occurrences.entrySet()) {  
if (entry.getValue() == maxValueInMap) {  
System.out.println("Наиболее часто встречающаяся буква: " + entry.getKey());  
break;  
}  
}  
  
}  
}

Результат выполнения представлен ниже (Рисунок 5).

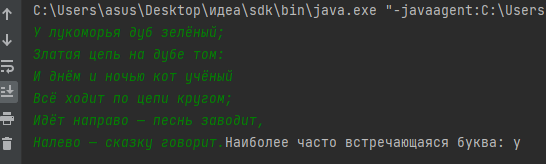


Рисунок 5 – Результат выполнения Задачи 5