Министерство образования Калининградской области

государственное бюджетное учреждение Калининградской области

профессиональная образовательная организация

«Колледж информационных технологий и строительства»

(ГБУ КО ПОО «КИТиС»)

**Отчет по учебной практике**

УП.01 Разработка модулей программного обеспечения для компьютерных систем

по ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

Специальность 09.02.07 «Информационные системы и программирование»

Сроки прохождения практики:

с «02» декабря 2020 г. по «22» декабря 2020 г.

Место практики ГБУ КО ПОО «КИТиС»

|  |  |
| --- | --- |
| Выполнил: | студент 3 курса,  группы ИС 18-2к  <Даниелян Гор Артурович>  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Проверила: | Большакова-Стрекалова Анна Викторовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Калининград, 2020

Содержание

[Введение 4](#_Toc58457762)

[1 Решение задач 6](#_Toc58457763)

[1.1 1 задача 6](#_Toc58457764)

[1.2 Задача 2 9](#_Toc58457765)

[1.3 Задача 3 17](#_Toc58457766)

[1.4 19](#_Toc58457767)

[1.5 Задача 5 22](#_Toc58457768)

# Введение

Учебная практика является этапом теоретического обучения и подготовки студентов к профессиональной деятельности в различных организациях.

Цель учебной практики – приобретение основных навыков практической работы по специальности – "Информационные системы и программирования", применения современных средств вычислительной техники, систематизации и закрепление теоретических знаний, формирование информационной базы.

Целью данной работы является подробный отчет о выполнении задании, данных на практику.

В ходе учебной практики мною были выдвинуты следующие задачи:

- Решение задач по вариантам, учебник «Java - Методы программирования»

## 1 Решение задач

### 1.1 1 задача стр. 50 (с варианта В)

Вариант 10

Условие

Вычислить норму матрицы

Таблица   идентификаторов

|  |  |  |
| --- | --- | --- |
| Наименование | Тип Данных | Назначение |
| max | Integer | Выходная |

Блок Схема

Вычеслить max матрицы

Вычеслить min матрицы

Да дддддддд нет

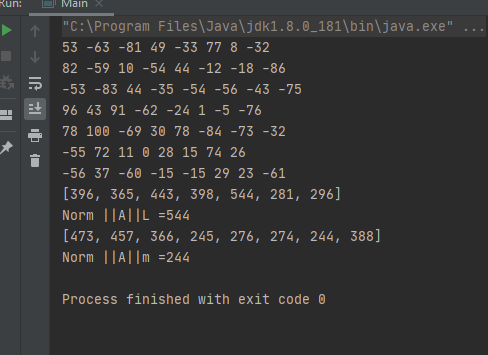
return max

return min;

Код программы

import java.util.Arrays;  
import java.util.Random;  
public class Main {  
 final static Random *rand* = new Random();  
 public static void main(String[] args) {  
  
 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;  
 }  
}

Выполнение



### 1.2 Задача 2 стр. 87 Вариант 10

Условие

Создать классы, спецификации которых приведены ниже. Определить конструкторы и методы setТип(), getТип(), toString(). Определить дополнительно методы в классе, создающем массив объектов. Задать критерий выбора данных и вывести эти данные на консоль. В каждом классе, обладающем информацией, должно быть объявлено несколько конструкторов.

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.

Таблица   идентификаторов

|  |  |  |
| --- | --- | --- |
| Наименование | Тип Данных | Назначение |
| destination  number  timeOfDeparture  seatCountHolder | String  String  String  Integer | Входная  Входная Входная  Выходная |

Блок схема

Вычеслить сколько мест в вагоне

нет

Да

Вычеслить направлене поезда

Свободных мест нету

Количество свободных мест в Разных классах

Null

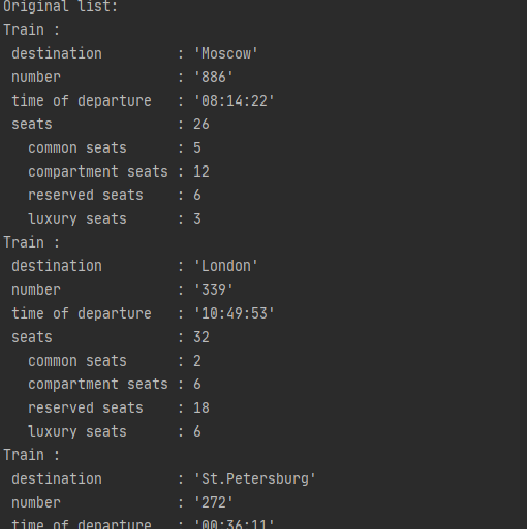
Код программы

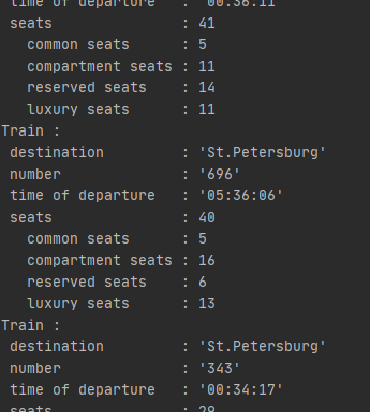
public class Main {  
 public static void main(String[] args) {  
 List<Train> trainsList = Trains.*newRandomTrainList*(10);  
  
 System.*out*.println("Original list:");  
 trainsList.forEach(System.*out*::println);  
  
 List<Train> trainsWithGivenDestination = Trains.*findTrainsWithSameDestination*(trainsList, "Moscow");  
 System.*out*.println("Trains for Moscow:");  
 trainsWithGivenDestination.forEach(System.*out*::println);  
  
 List<Train> trainsWithGivenDestinationAndTime = Trains  
 .*findTrainsWithSameDestinationAndAfterTime*(trainsList, "Paris", "09:00:00");  
 System.*out*.println("Trains for Paris and 09:00:");  
 trainsWithGivenDestinationAndTime.forEach(System.*out*::println);  
  
 List<Train> trainsWithGivenDestinationAndSeats = Trains  
 .*findTrainsWithSameDestinationAndHaveCommonSeats*(trainsList, "Ekaterinburg");  
 System.*out*.println("Trains for Ekaterinburg and seats:");  
 trainsWithGivenDestinationAndSeats.forEach(System.*out*::println);  
 }  
}

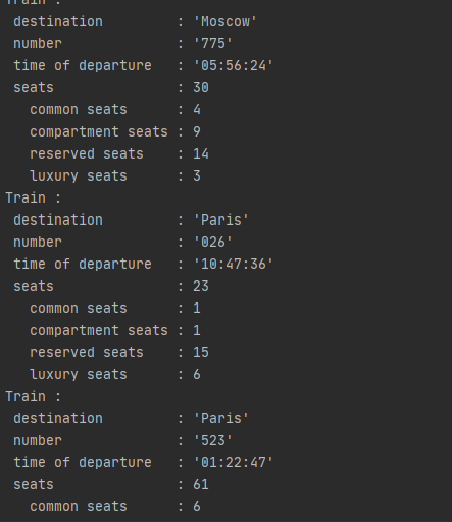
public class Train {  
 private String destination;  
 private String number;  
 private String timeOfDeparture;  
  
 private EnumMap<CarType, Integer> seatCountHolder;  
  
 public enum CarType {  
 *COMMON*, *COMPARTMENT*, *RESERVED\_SEAT*, *LUXURY*;  
 }  
  
 public Train(String destination, String number, String timeOfDeparture,  
 final int commonSeatsCount, final int compartmentSeatsCount,  
 final int reservedSeatCount, final int luxurySeatCount ) {  
 this.destination = destination;  
 this.number = number;  
 this.timeOfDeparture = timeOfDeparture;  
  
 this.seatCountHolder = new EnumMap<CarType, Integer>(CarType.class);  
  
 seatCountHolder.put(CarType.*COMMON*, commonSeatsCount);  
 seatCountHolder.put(CarType.*COMPARTMENT*, compartmentSeatsCount);  
 seatCountHolder.put(CarType.*RESERVED\_SEAT*, reservedSeatCount);  
 seatCountHolder.put(CarType.*LUXURY*, luxurySeatCount);  
 }  
  
 public String getDestination() {  
 return destination;  
 }  
  
 public void setDestination(String destination) {  
 this.destination = destination;  
 }  
  
 public String getNumber() {  
 return number;  
 }  
  
 public void setNumber(String number) {  
 this.number = number;  
 }  
  
 public String getTimeOfDeparture() {  
 return timeOfDeparture;  
 }  
  
 public void setTimeOfDeparture(String timeOfDeparture) {  
 this.timeOfDeparture = timeOfDeparture;  
 }  
  
 public int getSeatCountByCarType(final CarType type) {  
 return seatCountHolder.get(type);  
 }  
  
 public void setSeatCountByCarType(final CarType type, final int newCount) {  
 seatCountHolder.put(type, newCount);  
 }  
  
 public int getTotalSeatCount() {  
 int count = 0;  
  
 for (CarType type : CarType.*values*())  
 count = count + seatCountHolder.get(type);  
  
 return count;  
 }  
  
 @Override  
 public String toString() {  
 return "Train :\n" +  
 " destination : '" + destination + "\'\n" +  
 " number : '" + number + "\'\n" +  
 " time of departure : '" + timeOfDeparture + "\'\n" +  
 " seats : " + getTotalSeatCount() + "\n" +  
 " common seats : " + seatCountHolder.get(CarType.*COMMON*) + "\n" +  
 " compartment seats : " + seatCountHolder.get(CarType.*COMPARTMENT*) + "\n" +  
 " reserved seats : " + seatCountHolder.get(CarType.*RESERVED\_SEAT*) + "\n" +  
 " luxury seats : " + seatCountHolder.get(CarType.*LUXURY*);  
 }  
}

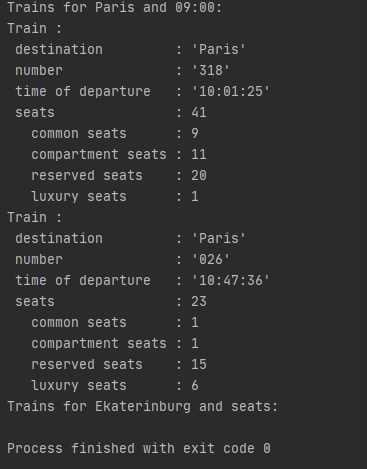
public class Trains {  
 private static final String[] *CITIES* = {"Moscow", "St.Petersburg", "Ekaterinburg", "Paris", "London"};  
  
 private static final ThreadLocalRandom *RNG* = ThreadLocalRandom.*current*();  
  
  
 public static List<Train> findTrainsWithSameDestination(final Collection<Train> trains, final String destination) {  
 List<Train> result = new ArrayList<>(trains);  
 return result.stream()  
 .filter(train -> Objects.*equals*(train.getDestination(), destination))  
 .collect(Collectors.*toList*());  
 }  
  
 public static List<Train> findTrainsWithSameDestinationAndAfterTime(final Collection<Train> trains,  
 final String destination,  
 final String givenTime) {  
 List<Train> result = new ArrayList<>(trains);  
 return result.stream()  
 .filter(train -> Objects.*equals*(train.getDestination(), destination))  
 .filter(train -> DatatypeConverter.*parseTime*(train.getTimeOfDeparture()).after(  
 DatatypeConverter.*parseTime*(givenTime)))  
 .collect(Collectors.*toList*());  
 }  
  
 public static List<Train> findTrainsWithSameDestinationAndHaveCommonSeats(final Collection<Train> trains,  
 final String destination) {  
 List<Train> result = new ArrayList<>(trains);  
 return result.stream()  
 .filter(train -> Objects.*equals*(train.getDestination(), destination))  
 .filter(train -> train.getSeatCountByCarType(Train.CarType.*COMMON*) > 0)  
 .collect(Collectors.*toList*());  
 }  
  
 public static List<Train> newRandomTrainList(final int trainCount) {  
 List<Train> trains = new ArrayList<>(trainCount);  
  
 for (int i = 0; i < trainCount; i++)  
 trains.add(*newRandomTrain*());  
  
 return trains;  
 }  
  
 public static Train newRandomTrain() {  
 final String city = *randomCity*();  
 final String number = *randomNumber*();  
 final String time = *randomTime*();  
  
 Train train = new Train(city, number, time, 0, 0, 0, 0);  
 for (Train.CarType type : Train.CarType.*values*())  
 train.setSeatCountByCarType(type, *RNG*.nextInt(21));  
 return train;  
 }  
  
 private static String randomCity() {  
 return *CITIES*[*RNG*.nextInt(*CITIES*.length)];  
 }  
  
 private static String randomTime() {  
 return String.*format*("%02d:%02d:%02d", *RNG*.nextInt(13), *RNG*.nextInt(60), *RNG*.nextInt(60));  
 }  
  
 private static String randomNumber() {  
 return String.*format*("%03d", *RNG*.nextInt(1000));  
 }  
}

Выполнение









### 1.3 Задача 3 стр.124

Вариант 10

Условие

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

Таблица   идентификаторов

|  |  |  |
| --- | --- | --- |
| Наименование | Тип Данных | Назначение |
| Dog puppy=new Dog("Sirko"); | String | Входная |

Блок схема

Задать имя собаки

Обучать собаку к различным командам

Да

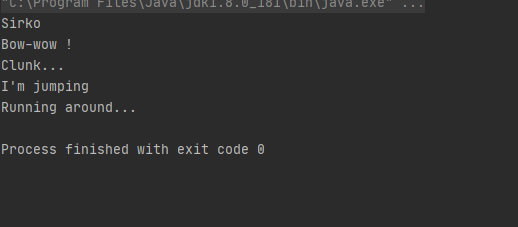
Собака обучена

подать голос, прыгать, бегать, кусать

Код программы

public class Blinov4A10\_Puppy {  
 public static void main(String[] args) {  
  
 Dog puppy=new Dog("Sirko");  
 puppy.showName();  
 puppy.makeVoice();  
 puppy.makeBite();  
 puppy.makeJump();  
 puppy.makeRun();  
}  
}  
  
class Animal {  
 private final String name;  
  
 public Animal(String name) {  
 this.name=name;  
 }  
  
 public void showName() {  
 System.*out*.println(name);  
 }  
}  
  
class Dog extends Animal {  
  
 public Dog(String name) {  
 super(name);  
 }  
  
 public void makeJump() {  
 System.*out*.println("I'm jumping");  
 }  
  
 public void makeVoice() {  
 System.*out*.println("Bow-wow !");  
 }  
  
 public void makeRun() {  
 System.*out*.println("Running around...");  
 }  
  
 public void makeBite() {  
 System.*out*.println("Clunk...");  
 }  
}

Выполнение



### 1.4 Задача 4 стр.145

Вариант 10

Условие

Создать класс Mobile с внутренним классом, с помощью объектов которого можно хранить информацию о моделях телефонов и их свойствах.

Таблица   идентификаторов

|  |  |  |
| --- | --- | --- |
| Наименование | Тип Данных | Назначение |
| brand  model | String  String | Входная  Выходная |

Блок схема

Создать метод model

Создать метод brand

Занести данные

данные есть null

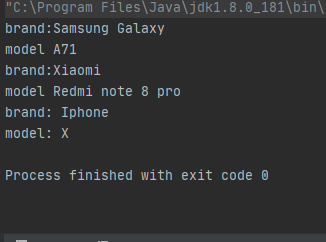
Показ данных на консоли

Ошибка введите данные

Код программы

public class Mobile {  
  
 public static <ModelProperties> void main(String[] args) {  
  
 String brand;  
 String model;  
 ModelProperties modelProperties;  
  
 brand:"Samsung Galaxy".toString();  
 System.*out*.println("brand:Samsung Galaxy");  
 model:"A71".toString();  
 System.*out*.println("model A71");  
  
 brand:"Xiaomi".toString();  
 System.*out*.println("brand:Xiaomi");  
  
 model:"model:Redmi note 8 pro".toString();  
  
 System.*out*.println("model Redmi note 8 pro");  
  
 brand:"Iphome".toString();  
  
 model:"X".toString();  
  
 System.*out*.println("brand: Iphone");  
 System.*out*.println("model: X");  
  
  
 }  
  
 private class ModelProperties {  
  
 private String model;  
 private long accumCapacity;  
 private double screenSize;  
  
 public void setModel(String X) {  
 this.model = model;  
  
  
  
 }  
  
 }}

Выполнение



### 1.5 Задача 5 стр.194

Вариант 10

Условие

В тексте найти и напечатать n символов (и их количество), встречающихся наиболее часто.

Таблица Индетификаторов

|  |  |  |
| --- | --- | --- |
| Наименование | Тип Данных | Назначение |
| k=1  str = sc.nextLine(); | Integer  String | Входная  Выходная |

Блок-Схема

Создать метод Scanner

Примнить строку

Расчитать буквы

for (int i = 0; i < str.length(); i++) { Character c = str.charAt(i);

Код программы

import java.util.HashMap;  
import java.util.Iterator;  
import java.util.Map;  
import java.util.Map.Entry;  
import java.util.Scanner;  
import java.util.Set;  
// Практика день 5 Даниелян Гор Артурович ИС 18-2К стр 194 вариант 10  
public class Main {  
  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Введите строку: ");  
 String str = sc.nextLine();  
 sc.close();  
  
 Map<Character, Integer> map = new HashMap<Character, Integer>();  
 for (int i = 0; i < str.length(); i++) {  
 Character c = str.charAt(i);  
 int k = 1;  
 if (map.containsKey(c)) {  
 k = map.get(c) + 1;  
 }  
 map.put(c, k);  
 }  
  
 Set<Entry<Character, Integer>> setvalue = map.entrySet();  
 Iterator<Entry<Character, Integer>> i = setvalue.iterator();  
 Entry<Character, Integer> max = null;  
 while (i.hasNext()) {  
 Entry<Character, Integer> me = i.next();  
 if (max == null) {  
 max = me;  
 } else if (me.getValue() > max.getValue()) {  
 max = me;  
 }  
 }  
  
 System.*out*.println("Cимвол в строке, встречающийся наибольшее число раз : \'" + max.getKey() + "\'.");  
 } }

Выполнение

