**Javaprogram:**

**1.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*1D and 2D Array LC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1. Next Customer ID

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

int Count,Max,Next;

Scanner scanner = new Scanner(System.in);

Count = scanner.nextInt();

int CustomerID[]=new int[Count];

for(int i=0;i<Count;i++)

{

CustomerID[i] = scanner.nextInt();

}

Max = CustomerID[0];

for(int i=1;i<Count;i++)

{

if(Max<CustomerID[i])

{

Max = CustomerID[i];

}

}

Next = Max+1;

System.out.println("Next customer id is

"+Next);

scanner.close();

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Missing Customer

import java.util.Scanner;

public class Main {

public static void main(String[] args)

{

int Count,total;

Scanner scanner = new Scanner(System.in);

Count = scanner.nextInt();

int CustomerID[]=new int[Count];

for(int i=0;i<Count;i++)

{

CustomerID[i] = scanner.nextInt();

}

total = (Count+1)\*(Count+2)/2;

for (int i = 0; i< Count; i++)

total -= CustomerID[i];

System.out.println("Customer id "+total+" is

missing");

scanner.close();

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Sort By Rate

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int Count;

Double temp;

String tempname;

Scanner scanner = new Scanner(System.in);

Count = scanner.nextInt();

String Name[]=new String[Count];

for(int i=0;i<Count;i++)

{

Name[i] = scanner.next();

}

Double Rate[]=new Double[Count];

for(int i=0;i<Count;i++)

{

Rate[i] = scanner.nextDouble();

}

for (int i = 0; i < Count; i++)

{

for (int j = i + 1; j < Count; j++)

{

if (Rate[i] > Rate[j])

{

temp = Rate[i];

Rate[i] = Rate[j];

Rate[j] = temp;

tempname = Name[i];

Name[i] = Name[j];

Name[j] = tempname;

}

}

}

System.out.println("Details are :");

for(int i=0;i<Count;i++)

{

System.out.println(Name[i]+" -

"+Rate[i]);

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Ports

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

int Count,flag,header;

String port\_temp;

Scanner scanner = new Scanner(System.in);

System.out.println("Enter number of ports :");

Count = Integer.parseInt(scanner.nextLine());

System.out.println("Enter port details :");

String port[]=new String[Count];

for(int i=0;i<Count;i++)

{

port[i] = scanner.nextLine();

}

System.out.println("One mode of transportation :");

flag = 0;

header = 0;

for(int i=0;i<Count;i++)

{

port\_temp = port[i];

String Exp[] = port\_temp.split("[|]");

if(Integer.parseInt(Exp

[2])+Integer.parseInt(Exp[3])+Integer.parseInt(Exp[4]) == 1)

{

if(header==0){

System.out.println("Id Name");

header=1;

}

System.out.format("%-5s%s \n",Exp[0],Exp[1]);

flag = 1;

}

}

if(flag == 0) {

System.out.println("No such transportation available");

}

System.out.println("More than one mode of transportation :");

flag = 0;

header = 0;

for(int i=0;i<Count;i++)

{

port\_temp = port[i];

String Exp[] = port\_temp.split("\\|");

if(Integer.parseInt(Exp[2])+Integer.parseInt(Exp[3])+Integer.parseInt(Exp[4]) > 1) {

if(header==0){

System.out.println("Id Name");

header=1;

}

System.out.format("%-5s%s\n",Exp[0],Exp[1]);

flag = 1;

}

}

if(flag == 0) {

System.out.println("No such transportation available");

}

scanner.close();

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Search Array

import java.util.Scanner;

public class Main{

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int Count = Integer.parseInt(scanner.nextLine());

String Name[]=new String[Count];

for(int i=0;i<Count;i++){

Name[i] = scanner.nextLine();

}

System.out.println("Enter the name you want to search");

String Search = scanner.nextLine();

Boolean flag = false;

for(int i=0;i<Count;i++){

if(Name[i].equalsIgnoreCase(Search))

{

System.out.println(Search+" found!");

flag = true;

}

}

if(flag == false) {

System.out.println(Search+" not found");

}

scanner.close();

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Shippment Company

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

int CargoCount,ShipmentCount;

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of cargo");

CargoCount = Integer.parseInt(scanner.nextLine());

System.out.println("Enter the cargo id");

int CargoID[]=new int[CargoCount];

for(int i=0;i<CargoCount;i++){

CargoID[i] = Integer.parseInt(scanner.nextLine());

}

System.out.println("Enter the number of shipment");

ShipmentCount = Integer.parseInt(scanner.nextLine());

System.out.println("Enter the shipment name");

String ShipmentName[]=new String[ShipmentCount];

for(int i=0;i<ShipmentCount;i++){

ShipmentName[i] = scanner.nextLine();

}

System.out.println("Enter the cargo id");

int SearchID = Integer.parseInt(scanner.nextLine());

for(int i=0;i<ShipmentCount;i++){

if(SearchID == CargoID[i]){

System.out.println(ShipmentName[i]);

break;

}

if(i==(ShipmentCount-1)){

System.out.println("Yet to be shipped");

}

}

scanner.close();

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. Port Route

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

int Count;

String temp;

Scanner scanner = new Scanner(System.in);

Count = Integer.parseInt(scanner.nextLine());

String port[][]=new String[Count][Count];

for(int i=0;i<Count;i++)

{

temp = scanner.nextLine();

String arr[] = temp.split("\\s");

for(int j=0;j<Count;j++){

port[i][j]=arr[j];

}

}

System.out.println("Enter two port numbers A and B :");

int row,column;

row = scanner.nextInt();

column = scanner.nextInt();

if(Integer.parseInt(port[row-1][column-1]) == 1){

System.out.println("There is a route");

}

else{

System.out.println("There is no route");

}

scanner.close();

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1.b.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*1D and 2D Array CC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Frequently Purchased Item

import java.util.Arrays;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

int Count;

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of items");

Count = Integer.parseInt(scanner.nextLine());

System.out.println("Enter the item id details");

int itemID[]=new int[Count];

for(int i=0;i<Count;i++)

{

itemID[i] = Integer.parseInt(scanner.nextLine());

}

Arrays.sort(itemID);

for(int i=Count-1;i>0;i--)

{

if(itemID[i] == itemID[i - 1]) {

System.out.println("Frequently purchased item is : "+itemID[i]);

break;

}

if(i==1){

System.out.println("There is no frequently purchased item");

}

}

scanner.close();

}

}

------------------------------------------------------------------------------------------------------------------------------------------

**2.a.\*\*\*\*\*\*\*Access Control LC\*\*\*\*\*\***

1. Display Cargo Details

Main.Java

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Cargo cargoObject = new Cargo();

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the cargo details:");

System.out.println("Enter the name");

cargoObject.setName(scanner.nextLine());

System.out.println("Enter the description");

cargoObject.setdescription(scanner.nextLine());

System.out.println("Enter the length");

cargoObject.setlength(Double.parseDouble(scanner.nextLine()));

System.out.println("Enter the width");

cargoObject.setwidth(Double.parseDouble(scanner.nextLine()));

System.out.println("The cargo details are:");

cargoObject.displayCargoDetails();

scanner.close();

}

}

Cargo.java

public class Cargo {

private String name,description;

private Double width,length;

public String getName() {

return this.name;

}

public String getdescription() {

return this.description;

}

public Double getlength() {

return this.length;

}

public Double getwidth() {

return this.width;

}

public void setName(String name) {

this.name = name;

}

public void setdescription(String description) {

this.description = description;

}

public void setlength(Double length) {

this.length = length;

}

public void setwidth(Double width) {

this.width = width;

}

public void displayCargoDetails(){

System.out.format("%-15s %-15s", "Name :",getName());

System.out.format("%-15s %-15s", "Description :",getdescription());

System.out.format("%-15s %-15s %-15s", "Length :",getlength(),"cm");

System.out.format("%-15s %-15s %-15s", "Width :",getwidth(),"cm");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2. Constructor,StatiC

Main.Java

import java.util.Scanner;

import java.io.\*;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of shipments :");

int c=sc.nextInt();

sc.nextLine();

if(c>0) {

Shipment[] Shpm=new Shipment[c];

for(int i=0;i<c;i++) {

System.out.println("Enter the shipment "+(i+1)+" details");

System.out.println("Enter the shipper name :");

String sN=sc.nextLine();

System.out.println("Enter the mode of transportation :");

String moT=sc.nextLine();

System.out.println("Enter the total weight :");

float tW=sc.nextFloat();

sc.nextLine();

System.out.println("Enter the arrival port :");

String aP=sc.nextLine();

System.out.println("Enter the departure port :");

String dP=sc.nextLine();

Shpm[i]=Shipment.createNewShipment(sN, moT, tW, aP, dP);

}

System.out.println("Shipment details are");

System.out.format(("%-15s%-15s%-25s%-15s%-20s%s%n"),

"Id","Shippername","Mode of transportation","Total weight", "Arrival port","Departure port");

for(int i=0;i<c;i++) {

System.out.println(Shpm[i].toString());

}

}else

System.out.println("Invalid Input");

}

}

Shipment.java

import java.text.DecimalFormat;

public class Shipment {

private int id;

private String shipperName;

private String modeOfTransportation;

private float totalWeight;

private String arrivalPort;

private String departurePort;

private static int nextShipmentId = 1000;

public Shipment(int ID,String SN,String MoT,float TW,String AP,String DP) {

this.id=ID;

this.shipperName=SN;

this.modeOfTransportation=MoT;

this.totalWeight=TW;

this.arrivalPort=AP;

this.departurePort=DP;

}

public static Shipment createNewShipment(String SN,String MoT,float TW,String AP,String DP ) {

nextShipmentId++;

int id = nextShipmentId;

Shipment shpmt = new Shipment(id, SN, MoT, TW, AP, DP);

return shpmt;

}

@Override

public String toString() {

DecimalFormat format = new DecimalFormat("0.##");

String weight = format.format(this.totalWeight);

return String.format("%-15d%-15s%-25s%-15s%-20s%s",

this.id,

this.shipperName,

this.modeOfTransportation,

weight+ "Kg",

//String.valueOf(this.totalWeight)+"Kg",

this.arrivalPort,

this.departurePort);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3. Display Port Details

Main.java

public class Main {

public static void main(String[] args) {

int id;

String name;

String portOrigin = "";

String country = "";

java.util.Scanner sc = new java.util.Scanner(System.in);

System.out.println("Enter the number of ports");

int count = Integer.parseInt(sc.nextLine());

Port[] port = new Port[count];

for (int i = 0; i < count; i++) {

System.out.println("Enter the port" + (i + 1) + " details");

id = Integer.parseInt(sc.nextLine());

name = sc.nextLine();

System.out.println("Is the port from same country[Y/N]");

portOrigin = sc.nextLine();

if (portOrigin.equalsIgnoreCase("N")) {

System.out.println("Enter the country");

country = sc.nextLine();

}

else

country = "India";

port[i] = Port.createRecord(id, name, country);

}

System.out.println("The port details are");

System.out.format(("%-15s %-15s %-15s%n"), "Id", "Name", "Country");

for (int i = 0; i < count; i++) {

System.out.println(port[i].toString());

}

sc.close();

}

}

Port.java

public class Port {

private int id;

private String name;

private String country;

public Port(int id,String name){

this.id=id;

this.name=name;

this.country="India";

}

public Port(int id,String name,String country){

this.id = id;

this.name = name;

this.country = country;

}

public int getid() {

return this.id;

}

public String getName() {

return this.name;

}

public String getCountry() {

return this.country;

}

public void setid(int id) {

this.id = id;

}

public void setName(String name) {

this.name = name;

}

public void setcountry(String country) {

this.country = country;

}

@Override

public String toString() {

return String.format(("%-15s %-15s %-15s"), this.id, this.name, this.country);

}

public static Port createRecord(int id, String name, String country) {

Port port = new Port(id, name, country);

return port;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4. Dispaly Agent/Company

AgentCompanyFee.java

public class AgentCompanyFee {

private Integer fees;

private Agent agent;

private Company company;

public AgentCompanyFee(Integer fees, Agent agent, Company company) {

super();

this.fees = fees;

this.agent = agent;

this.company = company;

}

public Integer getFees() {

return fees;

}

public void setFees(Integer fees) {

this.fees = fees;

}

public Agent getAgent() {

return agent;

}

public void setAgent(Agent agent) {

this.agent = agent;

}

public Company getCompany() {

return company;

}

public void setCompany(Company company) {

this.company = company;

}

}

Company.java

public class Company {

private int id;

private String fmcCode;

private String address;

private String companyName;

public Company(int id, String fmcCode, String address, String companyName) {

super();

this.id = id;

this.fmcCode = fmcCode;

this.address = address;

this.companyName = companyName;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getFmcCode() {

return fmcCode;

}

public void setFmcCode(String fmcCode) {

this.fmcCode = fmcCode;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getCompanyName() {

return companyName;

}

public void setCompanyName(String companyName) {

this.companyName = companyName;

}

}

Main.java

import java.io.\*;

public class Main {

Company[] companies = Main.initCompany();

public static Company[] initCompany() {

Company[] companies = new Company[4];

companies[0] = new Company(1,"FM01","India","Titanic");

companies[1] = new Company(2,"FM02","America","Arcadia");

companies[2] = new Company(3,"FM03","England","Umbrella corporation");

companies[3] = new Company(4,"FM04","France","Omnicorp");

return companies;

}

public static void main(String args[])throws Exception{

Main main = new Main();

BufferedReader buf = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of agents :");

Integer n = Integer.parseInt(buf.readLine());

AgentBO agentBo = new AgentBO();

AgentCompanyFee[] agentCompanyFee = new AgentCompanyFee[n];

int companyChoice;

int agentChoice;

for(Integer i=0;i<n;i++){

System.out.println("Select the company where the agent "+(i+1)+" is going to work :");

System.out.println("1. Titanic\n2. Arcadia\n3. Umbrella corporation\n4. Omnicorp\nEnter your choice :");

companyChoice = Integer.parseInt(buf.readLine());

System.out.println("1. New Agent\n2. Existing Agent\nEnter the choice :");

agentChoice = Integer.parseInt(buf.readLine());

switch(agentChoice){

case 1:

System.out.println("Enter the agent details :");

String[] fg=buf.readLine().split(",");

System.out.println("Enter the fee of the agent :");

Integer fee = Integer.parseInt(buf.readLine());

Company[] companies = initCompany();

agentCompanyFee[i]=new AgentCompanyFee(fee,new Agent(companies[companyChoice-1].getFmcCode(),fg[0],fg[2],fg[1]),companies[companyChoice-1]);

agentCompanyFee[i].setFees(fee);

agentCompanyFee[i].setAgent(new Agent(companies[companyChoice-1].getFmcCode(),fg[0],fg[2],fg[1]));

agentCompanyFee[i].setCompany(companies[companyChoice-1]);

break;

case 2:

System.out.println("Enter the name :");

String fgn=buf.readLine();

System.out.println("Enter the fee of the agent :");

Integer nfee = Integer.parseInt(buf.readLine());

Company[] companien = initCompany();

for (int j = 0; j < i; j++) {

if(agentCompanyFee[j].getAgent().getName()!=null && agentCompanyFee[j].getAgent().getName().equalsIgnoreCase(fgn)){

agentCompanyFee[i]=new AgentCompanyFee(nfee,new Agent(companien[companyChoice-1].getFmcCode(),agentCompanyFee[j].getAgent().getIataCode(),agentCompanyFee[j].getAgent().getName(),agentCompanyFee[j].getAgent().getAddress()),companien[companyChoice-1]);

agentCompanyFee[i].setFees(nfee);

agentCompanyFee[i].setAgent(agentCompanyFee[j].getAgent());

agentCompanyFee[i].setCompany(companien[companyChoice-1]);

}

}

break;// fill code here

}

}

System.out.println("Search:\n1. Find agents\n2. Find companies");

int choice = Integer.parseInt(buf.readLine());

// fill code here

switch(choice){

case 2:

System.out.println("Enter the agent name :");

String nam=buf.readLine();

System.out.println("Agent works for");

agentBo.displayCompanyDetailsByAgent(nam, agentCompanyFee);

break;

case 1:

System.out.println("Enter the company name :");

String name=buf.readLine();

System.out.println("Company has");

agentBo.displayAgentDetailsByCompany(name, agentCompanyFee);

break;

}

}

}

Agent BO.java

import java.io.\*;

public class AgentBO {

public void displayAgentDetailsByCompany(String name,AgentCompanyFee[] agentCompanyFee) throws IOException{

for (int i = 0; i < agentCompanyFee.length; i++) {

if(agentCompanyFee[i].getCompany().getCompanyName().equalsIgnoreCase(name)){

System.out.println(agentCompanyFee[i].getAgent().getName()+" "+agentCompanyFee[i].getFees());

}

}

}

public void displayCompanyDetailsByAgent(String name,AgentCompanyFee[] agentCompanyFee) throws IOException{

for (int i = 0; i < agentCompanyFee.length; i++) {

if(agentCompanyFee[i].getAgent().getName().equalsIgnoreCase(name)){

System.out.println(agentCompanyFee[i].getCompany().getCompanyName());

}

}

}

}

Agent.java

public class Agent {

private String fmcCode;

private String iataCode;

private String address;

private String name;

public Agent(String fmcCode, String iataCode, String address, String name) {

super();

this.fmcCode = fmcCode;

this.iataCode = iataCode;

this.address = address;

this.name = name;

}

public String getFmcCode() {

return fmcCode;

}

public void setFmcCode(String fmcCode) {

this.fmcCode = fmcCode;

}

public String getIataCode() {

return iataCode;

}

public void setIataCode(String iataCode) {

this.iataCode = iataCode;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

**1.b.\*\*\*\*\*\*\*\*Access Control CC\*\*\*\*\*\*\***

**Customer Invoice Details**

user.java

public class User {

private String name;

private String state;

private String country;

public User(String name, String state, String country) {

this.name = name;

this.state = state;

this.country = country;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public String toString() {

return String.format("%-15s %-15s %-15s",getName(),getState(),getCountry());

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

INvoice BO.java

public class InvoiceBO {

public void displayUserDetailsByStatus(String status,Invoice[] invoiceArray) {

int length = invoiceArray.length;

for(int i=0;i<length;i++) {

if (invoiceArray[i].getStatus().equalsIgnoreCase(status)) {

System.out.println(invoiceArray[i].getCustomer().toString());

}

}

}

public double getTotalInvoiceValue(String userName,Invoice[] invoiceArray) {

int length = invoiceArray.length;

double total = 0;

for(int i=0;i<length;i++) {

if (invoiceArray[i].getCreatedBy().equalsIgnoreCase(userName)) {

total = total + invoiceArray[i].getTotal();

}

}

return total;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Invoice.java

public class Invoice {

private User customer;

private Double total;

private String status;

private String createdBy;

public Invoice(User customer, Double total, String status, String createdBy) {

super();

this.customer = customer;

this.total = total;

this.status = status;

this.createdBy = createdBy;

}

public User getCustomer() {

return customer;

}

public void setCustomer(User customer) {

this.customer = customer;

}

public Double getTotal() {

return total;

}

public void setTotal(Double total) {

this.total = total;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public String getCreatedBy() {

return createdBy;

}

public void setCreatedBy(String createdBy) {

this.createdBy = createdBy;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.text.DecimalFormat;

import java.text.NumberFormat;

public class Main {

public static void main(String[] args) throws NumberFormatException, IOException {

BufferedReader buf = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of invoices :");

int n = Integer.parseInt(buf.readLine());

Invoice inv[] = new Invoice[n];

for(int i = 0; i <n; i++ ) {

int j = i+1;

System.out.println("Enter the invoice "+ j +" details :");

String Input = buf.readLine();

String arr[] = Input.split(",");

User cust = new User(arr[0], arr[1], arr[2]);

inv[i] = new Invoice(cust, Double.parseDouble(arr[3]), arr[4], arr[5]);

}

System.out.println("Customer details with status cleared Invoices :");

InvoiceBO invoiceBO = new InvoiceBO();

System.out.format("%-15s %-15s %-15s\n","Name","State","Country");

invoiceBO.displayUserDetailsByStatus("Cleared", inv);

System.out.println("Enter the Invoice creator name :");

String name = buf.readLine();

double total = invoiceBO.getTotalInvoiceValue(name, inv);

System.out.println("The total invoice value :");

if (total == 0.0) {

System.out.println("No invoice available");

}

else {

NumberFormat formatter = new DecimalFormat("#0.00");

System.out.println(formatter.format(total));

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3.a.\*\*\*\*\*\*\*\*\*\*Collections LC\*\*\*\*\*\*\*\*\*\*\***

1. List of commodity

Commodity.java

class Commodity{

private String commodityName;

private String quantity;

private String totalvalue;

private String totalWeight;

private String hazardous;

private String hazardous1;

Commodity(){}

Commodity(String commodityName,String quantity,String totalvalue,String totalWeight,String hazardous){

super();

this.commodityName=commodityName;

this.quantity=quantity;

this.totalvalue=totalvalue;

this.totalWeight=totalWeight;

this.hazardous=hazardous;

}

public String getCommodityName() {

return commodityName;

}

public void setCommodityName(String commodityName) {

this.commodityName = commodityName;

}

public String getQuantity() {

return quantity;

}

public void setQuantity(String quantity) {

this.quantity = quantity;

}

public String getTotalvalue() {

return totalvalue;

}

public void setTotalvalue(String totalvalue) {

this.totalvalue = totalvalue;

}

public String getTotalWeight() {

return totalWeight;

}

public void setTotalWeight(String totalWeight) {

this.totalWeight = totalWeight;

}

public String getHazardous() {

return hazardous;

}

public void setHazardous(String hazardous) {

this.hazardous = hazardous;

}

public String toString(){

if(hazardous.equals("True")) {

hazardous1 = "Yes";

}

else {

hazardous1 = "No";

}

String retval =String.format("%-15s%-15s%-15s%-15s%s", commodityName,quantity,totalvalue,totalWeight,hazardous1);

return retval;

}

}

main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of commodities:");

int count = Integer.parseInt(br.readLine());

String input;

String[] commodity;

System.out.println("Enter the commodity details");

Commodity[] object = new Commodity[count];

for(int i = 0;i<count;i++) {

List<Commodity> list = new ArrayList<Commodity>();

input = br.readLine();

commodity = input.split(",");

object[i] = new Commodity(commodity[0], commodity[1], commodity[2], commodity[3], commodity[4]);

list.add(object[i]);

}

System.out.println("Commodity details are:");

System.out.println("Commodity Name Quantity Total Value TotalWeight Hazardous");

for(int i = 0;i<count;i++) {

System.out.println(object[i].toString());

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2. Port Details

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

public class Main {

public static void main(String args[]) throws IOException{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of port details");

int count = Integer.parseInt(br.readLine());

String input;

String[] Port;

System.out.println("Enter the port details");

Port[] object = new Port[count];

List<Port> list = new ArrayList<Port>();

for(int i = 0;i<count;i++) {

input = br.readLine();

Port = input.split(",");

object[i] = new Port(Integer.parseInt(Port[0]), Port[1], Port[2]);

list.add(object[i]);

}

System.out.println("Port Id Name Location ");

for(int i = 0;i<count;i++) {

System.out.println(object[i].toString());

}

System.out.println("Enter the position");

int Position = Integer.parseInt(br.readLine());

System.out.println("Enter port detail to be inserted");

input = br.readLine();

Port = input.split(",");

Port objectadd = new Port(Integer.parseInt(Port[0]), Port[1], Port[2]);

list.add(Position-1,objectadd);

System.out.println("After the insertion of port details");

System.out.println("Port Id Name Location ");

/\*for(int i = 0;i<list.size();i++) {

System.out.println(object[i].toString());

}\*/

Iterator<Port> iterator = list.listIterator();

while (iterator.hasNext()) {

System.out.println(iterator.next().toString());

}

}

}

Port.java

public class Port {

Integer id;

String name;

String city;

Port(){}

Port(Integer id,String name,String city){

this.id = id;

this.name = name;

this.city = city;

}

void setId(Integer id){

this.id = id;

}

Integer getId(){

return id;

}

void setName(String name){

this.name = name;

}

String getName(){

return name;

}

void setCity(String city){

this.city = city;

}

String getCity(){

return city;

}

public String toString(){

return String.format("%-15s %-15s %-15s",getId(),getName(),getCity());

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Comportable Interface - Port Details

Port.java

class Port implements Comparable<Port>{

Integer id;

String name;

String country;

Port(){}

public Port(Integer id, String name, String country) {

this.id = id;

this.name = name;

this.country = country;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

@Override

public int compareTo(Port arg0) {

return this.country.compareTo(arg0.getCountry());

}

}

Main.java

import java.io.\*;

import java.util.\*;

public class Main {

public static void main(String args[]) throws IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

String flag,input;

String []details;

List<Port> List = new ArrayList<Port>();

System.out.println("Enter the port details");

do {

input = br.readLine();

details = input.split(",");

List.add(new Port(Integer.parseInt(details[0]), details[1], details[2]));

System.out.println("Do you want to continue[Yes/No]");

flag = br.readLine();

}while(flag.equals("Yes"));

Collections.sort(List);

System.out.println("Port details in sorted order");

System.out.println("Port Id Name Country ");

for(Port st:List){

System.out.format("%-15s %-15s %-15s\n",st.getId(),st.getName(),st.getCountry());

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4. Collections - Binary Search

ShipmentEntity.java

public class ShipmentEntity implements Comparable<ShipmentEntity>{

Integer id;

String name;

Integer bookingNumber;

public ShipmentEntity(Integer id, String name, Integer bookingNumber) {

this.id = id;

this.name = name;

this.bookingNumber = bookingNumber;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Integer getBookingNumber() {

return bookingNumber;

}

public void setBookingNumber(Integer bookingNumber) {

this.bookingNumber = bookingNumber;

}

@Override

public int compareTo(ShipmentEntity arg0) {

//if(bookingNumber == arg0.getBookingNumber())

// return 0;

//else if(bookingNumber > arg0.getBookingNumber())

// return 1;

//else

// return -1;

return bookingNumber.compareTo(arg0.getBookingNumber());

}

}

Main.java

import java.io.\*;

import java.util.\*;

public class Main {

public static void main(String args[]) throws IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

String flag,input;

String []details;

List<ShipmentEntity> List = new ArrayList<ShipmentEntity>();

System.out.println("Enter the shipment entity details");

do {

input = br.readLine();

details = input.split(",");

List.add(new ShipmentEntity(Integer.parseInt(details[0]), details[1], Integer.parseInt(details[2])));

System.out.println("Do you want to continue [yes/no]");

flag = br.readLine();

}while(flag.equals("yes"));

Collections.sort(List);

System.out.println("After sorting the shipment entity details");

System.out.println("Id Name Number ");

for(ShipmentEntity st:List){

System.out.format("%-15s %-15s %-15s\n",st.getId(),st.getName(),st.getBookingNumber());

}

System.out.println("Enter the shipment entity details to be searched");

input = br.readLine();

details = input.split(",");

Comparator<ShipmentEntity> c = new Comparator<ShipmentEntity>()

{

@Override

public int compare(ShipmentEntity u1, ShipmentEntity u2)

{

return u1.getBookingNumber().compareTo(u2.getBookingNumber());

}

};

int index = Collections.binarySearch(List, new ShipmentEntity(Integer.parseInt(details[0]), details[1], Integer.parseInt(details[2])));

//int index = Collections.binarySearch(List,new ShipmentEntity(null, null, Integer.parseInt(details[2])), c);

System.out.println("Shipment entity found in index : " + index);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3.b.\*\*\*\*\*\*Collections CC\*\*\*\*\*\*\*\***

**Sort by account number using Comparable Interface**

main.java

import java.io.\*;

import java.util.\*;

public class Main {

public static void main(String[] args) throws Exception{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int n,i,j;

String shipmentDetails;

String shipmentDetailsarr[];

System.out.println("Enter the number of shipment:");

n = Integer.parseInt(br.readLine());

List<ShipmentEntity> shipmentEntityList = new ArrayList<ShipmentEntity>();

for(i = 0 ; i < n ; i++)

{

System.out.println("Enter the shipment entity "+(i+1)+" details:");

shipmentDetails = br.readLine();

shipmentDetailsarr = shipmentDetails.split(",");

shipmentEntityList.add(new ShipmentEntity(Integer.parseInt(shipmentDetailsarr[0]), shipmentDetailsarr[1], shipmentDetailsarr[2], shipmentDetailsarr[3]));

}

Collections.sort(shipmentEntityList);

System.out.println("Shipment entity details:");

System.out.format("%-15s %-15s %-15s %s\n","Id","Name","Account number","Identificaion number");

for(ShipmentEntity st:shipmentEntityList){

System.out.format("%-15s %-15s %-15s %s\n",st.getId(),st.getName(),st.getAccountNumber(),st.getIdentificationNumber());

}

}

}

ShipmentEntity.java

public class ShipmentEntity implements Comparable<ShipmentEntity> {

private Integer id;

private String name;

private String accountNumber;

private String identificationNumber;

public ShipmentEntity(Integer id, String name, String accountNumber, String identificationNumber) {

this.id = id;

this.name = name;

this.accountNumber = accountNumber;

this.identificationNumber = identificationNumber;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAccountNumber() {

return accountNumber;

}

public void setAccountNumber(String accountNumber) {

this.accountNumber = accountNumber;

}

public String getIdentificationNumber() {

return identificationNumber;

}

public void setIdentificationNumber(String identificationNumber) {

this.identificationNumber = identificationNumber;

}

@Override

public int compareTo(ShipmentEntity o) {

return accountNumber.compareTo(o.getAccountNumber());

}

}

**4.a.\*\*\*\*\*\*\*Exception Handling LC\*\*\*\*\***

1.Arithmetic Exception

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws NumberFormatException, IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the container price :");

try{

int Price = Integer.parseInt(br.readLine());

System.out.println("Enter the number of items in the container :");

int Number = Integer.parseInt(br.readLine());

int Average = Price/Number;

System.out.println("The average price of the item is Rs."+Average);

}catch(Exception e){System.out.println("Exception : java.lang.ArithmeticException");}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2. Number Format Exception

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws NumberFormatException, IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the total number of items");

int Count = Integer.parseInt(br.readLine());

int[] Cost = new int[Count];

for(int i = 0; i < Count; i++) {

try {

System.out.println("Enter the shipping price of the item "+(i+1)+" :");

Cost[i] = Integer.parseInt(br.readLine());

}

catch(Exception e) {

System.out.println("Exception : java.lang.NumberFormatException");

System.out.println("Re-enter the item price :");

Cost[i] = Integer.parseInt(br.readLine());

}

}

int totalcost = 0;

for(int i = 0; i < Count; i++) {

totalcost = Cost[i]+totalcost ;

}

System.out.println("Total cost of the container is "+totalcost);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3. Invalid port Exception

Invalid port exception.java

public class InvalidPortException extends Exception {

InvalidPortException(String s){

super(s);

}

}

Port.java

public class Port {

private int id;

private String country;

private String name;

public Port()

{

}

public Port(int id,String country,String name)

{

this.id=id;

this.country=country;

this.name=name;

}

public int getId()

{

return this.id;

}

public void setId(int id)

{

this.id=id;

}

public String getCountry()

{

return this.country;

}

public void setCountry(String country)

{

this.country=country;

}

public String getName()

{

return this.name;

}

public void setName(String name)

{

this.name=name;

}

}

Shipment.java

public class Shipment {

private int id;

private String name;

private Port arrivalport;

private Port departureport;

Shipment()

{

}

public Shipment(int id,String name,Port arrivalport,Port departureport)

{

this.id=id;

this.name=name;

this.arrivalport=arrivalport;

this.departureport=departureport;

}

public int getId()

{

return this.id;

}

public void setId(int id)

{

this.id=id;

}

public String getName()

{

return this.name;

}

public void setName(String name)

{

this.name=name;

}

public Port getArrivalport()

{

return this.arrivalport;

}

public void setArrivalport(Port arrivalport)

{

this.arrivalport=arrivalport;

}

public Port getDepartureport()

{

return this.departureport;

}

public void setDepartureport(Port departureport)

{

this.departureport=departureport;

}

}

ShipmentBo.java

public class ShipmentBO {

public Boolean Validate(String p1,String p2,Port[] ports) throws InvalidPortException {

try {

if(p1.equalsIgnoreCase(p2)) {

throw new InvalidPortException("The port name is invalid");

}

else {

return true;

}

}

catch(InvalidPortException e) {

System.out.println("InvalidPortException: "+e.getMessage());

return false;

}

}

void displayShipmentDetails(Shipment shipmentObj,Port[] ports,String p1,String p2) {

String p1Name = "";

String p2Name = "";

String p1Country = "";

String p2Country = "";

for(int i=0;i<ports.length;i++) {

if(p1.equalsIgnoreCase(ports[i].getName())) {

p1Country = ports[i].getCountry();

}

if(p2.equalsIgnoreCase(ports[i].getName())) {

p2Country = ports[i].getCountry();

}

if(p1.equalsIgnoreCase(ports[i].getName())) {

p1Name = ports[i].getName();

}

if(p2.equalsIgnoreCase(ports[i].getName())) {

p2Name = ports[i].getName();

}

}

System.out.println("Shipment Details :");

System.out.format("%-20s%-20s%-20s%-20s\n", "Shipment id","Shipment name","Arrival place","Departure place");

System.out.format("%-20s%-20s%-20s%-20s\n", shipmentObj.getId(),shipmentObj.getName(),p1Name+"/"+p1Country,p2Name+"/"+p2Country);

}

}

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String args[])throws IOException, InvalidPortException{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

Port[] port = new Port[4];

port[0]=new Port(1,"India","Chennai");

port[1]=new Port(2,"America","California");

port[2]=new Port(3,"England","London");

port[3]=new Port(4,"Australia","Melbourne");

Shipment shipment=new Shipment();

System.out.println("Enter the Shipment Id ");

shipment.setId(Integer.parseInt(br.readLine()));

System.out.println("Enter the Shipment Name ");

shipment.setName(br.readLine());

System.out.println("Available ports are");

System.out.format("%-15s %-15s %s","ID","Country","PortName");

for(int i=0;i<port.length;i++)

{

System.out.format("\n%-15s %-15s %s",port[i].getId(),port[i].getCountry(),port[i].getName());

}

System.out.println("\nEnter the arrival port name");

String Arrivalport = br.readLine();

System.out.println("Enter the departure port name");

String Departureport = br.readLine();

ShipmentBO obj = new ShipmentBO();

if(obj.Validate(Arrivalport, Departureport, port)) {

obj.displayShipmentDetails(shipment, port, Arrivalport, Departureport);

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4.b.\*\*\*\*\*\*\*Exception Handling CC\*\*\*\*\***

**Contact Validation**

Contact BO.java

import java.util.regex.Pattern;

import java.util.regex.Matcher;

public class ContactBO {

public ContactBO()

{

}

private Pattern regexPattern;

private Matcher regMatcher;

public boolean validateContactDetails(Contact co) throws InvalidContactException

{

Pattern regexPattern = Pattern.compile("^.\*@.\*\\..\*$");

Matcher regMatcher = regexPattern.matcher(co.getEmail());

try {

if(co.getName().equals("") | co.getSname().equals("") | co.getAdd().equals("") | co.getCity().equals("") | co.getState().equals("") | co.getPh().equals("") | co.getEmail().equals("") ) {

throw new InvalidContactException("Make sure you entered all the fields");

}

else if(co.getPh().length() != 10) {

throw new InvalidContactException("Invalid Phone Number");

}

else if(regMatcher.matches()!=true) {

throw new InvalidContactException("Invalid email");

}

else {

return true;

}

}

catch(InvalidContactException e) {

System.out.println("InvalidContactException: "+e.getMessage());

return false;

}

}

}

Contact.java

public class Contact

{

public String name,streetname,address,city,state,phoneNumber,email;

public Contact(String name, String streetname, String add, String city, String state, String ph, String email)

{

this.name=name;

this.streetname=streetname;

this.address=add;

this.city=city;

this.state=state;

this.phoneNumber=ph;

this.email=email;

}

public void setName(String name)

{

this.name=name;

}

public String getName()

{

return name;

}

public void setSname(String streetname)

{

this.streetname=streetname;

}

public String getSname()

{

return streetname;

}

public void setAdd(String add)

{

this.address=add;

}

public String getAdd()

{

return address;

}

public void setCity(String city)

{

this.city=city;

}

public String getCity()

{

return city;

}

public void setState(String state)

{

this.state=state;

}

public String getState()

{

return state;

}

public void setPh(String ph)

{

this.phoneNumber=ph;

}

public String getPh()

{

return phoneNumber;

}

public void setEmail(String email)

{

this.email=email;

}

public String getEmail()

{

return email;

}

}

InvalidContactException.java

public class InvalidContactException extends Exception {

InvalidContactException(String s){

super(s);

}

}

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.\*;

public class Main {

public static void main(String arg[])throws IOException, InvalidContactException

{

String name,streetname,address,city,state,phoneNumber,email;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("CONTACT DETAILS");

System.out.println("Enter the name :");

name = br.readLine();

System.out.println("Enter the street name :");

streetname = br.readLine();

System.out.println("Enter the address :");

address = br.readLine();

System.out.println("Enter the city :");

city = br.readLine();

System.out.println("Enter the state :");

state = br.readLine();

System.out.println("Enter the phone number :");

phoneNumber = br.readLine();

System.out.println("Enter the email id :");

email = br.readLine();

Contact obj = new Contact(name, streetname, address, city, state, phoneNumber, email);

ContactBO val = new ContactBO();

if(val.validateContactDetails(obj)) {

System.out.println("Contact is valid");

}

}

}

------------------------------------------------------------------------------------------------------------------------------------------

**4.c.\*\*\*\*\*Exception LC\*\*\*\***

Contact Validation

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.\*;

public class Main {

public static void main(String arg[])throws IOException, InvalidContactException

{

String name,streetname,address,city,state,phoneNumber,email;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("CONTACT DETAILS");

System.out.println("Enter the name :");

name = br.readLine();

System.out.println("Enter the street name :");

streetname = br.readLine();

System.out.println("Enter the address :");

address = br.readLine();

System.out.println("Enter the city :");

city = br.readLine();

System.out.println("Enter the state :");

state = br.readLine();

System.out.println("Enter the phone number :");

phoneNumber = br.readLine();

System.out.println("Enter the email id :");

email = br.readLine();

Contact obj = new Contact(name, streetname, address, city, state, phoneNumber, email);

ContactBO val = new ContactBO();

if(val.validateContactDetails(obj)) {

System.out.println("Contact is valid");

}

}

}

Contact BO.java

import java.util.regex.Pattern;

import java.util.regex.Matcher;

public class ContactBO {

public ContactBO()

{

}

private Pattern regexPattern;

private Matcher regMatcher;

public boolean validateContactDetails(Contact co) throws InvalidContactException

{

Pattern regexPattern = Pattern.compile("^.\*@.\*\\..\*$");

Matcher regMatcher = regexPattern.matcher(co.getEmail());

try {

if(co.getName().equals("") | co.getSname().equals("") | co.getAdd().equals("") | co.getCity().equals("") | co.getState().equals("") | co.getPh().equals("") | co.getEmail().equals("") ) {

throw new InvalidContactException("Make sure you entered all the fields");

}

else if(co.getPh().length() != 10) {

throw new InvalidContactException("Invalid Phone Number");

}

else if(regMatcher.matches()!=true) {

throw new InvalidContactException("Invalid email");

}

else {

return true;

}

}

catch(InvalidContactException e) {

System.out.println("InvalidContactException: "+e.getMessage());

return false;

}

}

}

Contact.java

public class Contact

{

public String name,streetname,address,city,state,phoneNumber,email;

public Contact(String name, String streetname, String add, String city, String state, String ph, String email)

{

this.name=name;

this.streetname=streetname;

this.address=add;

this.city=city;

this.state=state;

this.phoneNumber=ph;

this.email=email;

}

public void setName(String name)

{

this.name=name;

}

public String getName()

{

return name;

}

public void setSname(String streetname)

{

this.streetname=streetname;

}

public String getSname()

{

return streetname;

}

public void setAdd(String add)

{

this.address=add;

}

public String getAdd()

{

return address;

}

public void setCity(String city)

{

this.city=city;

}

public String getCity()

{

return city;

}

public void setState(String state)

{

this.state=state;

}

public String getState()

{

return state;

}

public void setPh(String ph)

{

this.phoneNumber=ph;

}

public String getPh()

{

return phoneNumber;

}

public void setEmail(String email)

{

this.email=email;

}

public String getEmail()

{

return email;

}

}

Invalid Contact exception.java

public class InvalidContactException extends Exception {

InvalidContactException(String s){

super(s);

}

}

**5.a.\*\*\*\*\*\*\*\*\*OOPS Concept LC\*\*\*\*\*\*\*\***

**1. Display User Details**

Main.java

import java.io.IOException;

public class Main {

public static void main(String[] args) throws IOException {

java.util.Scanner sc = new java.util.Scanner(System.in);

System.out.println("Enter the total number of users");

int count = Integer.parseInt(sc.nextLine());

String details;

User Customer[] = new User[count];

System.out.println("Enter user details");

for(int i=0;i<count;i++){

details = sc.nextLine();

String Arr[] = details.split(",");

Customer[i] = new User();

Customer[i].setUserName(Arr[0]);

Customer[i].setFirstName(Arr[1]);

Customer[i].setLastName(Arr[2]);

Customer[i].setContact(Arr[3]);

}

System.out.println("1)Search user by user name");

System.out.println("2)Search user by first name and last name");

System.out.println("Enter your option");

int option = sc.nextInt();

if(option == 1){

User obj = new User();

User res = new User();

System.out.println("Enter the user name to search");

String username = sc.next();

res = obj.findUser(Customer, username);

if(res != null){

System.out.println("User details :");

System.out.println("Username :"+res.getUserName());

System.out.println("FirstName :"+res.getFirstName());

System.out.println("LastName :"+res.getLastName());

System.out.println("Contact :"+res.getContact());

}

else{

System.out.println("User not found");

}

}

else if(option == 2){

User obj = new User();

User res = new User();

System.out.println("Enter the first name to search");

String firstname = sc.next();

System.out.println("Enter the last name to search");

String lasttname = sc.next();

res = obj.findUser(Customer, firstname,lasttname );

if(res != null){

System.out.println("User details :");

System.out.println("Username :"+res.getUserName());

System.out.println("FirstName :"+res.getFirstName());

System.out.println("LastName :"+res.getLastName());

System.out.println("Contact :"+res.getContact());

}

else{

System.out.println("User not found");

}

}

sc.close();

}

}

User.java

public class User {

private String userName;

private String firstName;

private String lastName;

private String contact;

public User(String userName, String firstName, String lastName, String contact) {

this.userName = userName;

this.firstName = firstName;

this.lastName = lastName;

this.contact = contact;

}

User() {}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getContact() {

return contact;

}

public void setContact(String contact) {

this.contact = contact;

}

public User findUser(User userArray[],String userName){

User obj = new User();

for(int i=0;i<userArray.length;i++){

if(userArray[i].getUserName().equals(userName)){

obj = userArray[i];

break;

}

else{

obj = null;

}

}

return obj;

}

public User findUser(User userArray[],String firstName,String lastName){

User obj = new User();

for(int i=0;i<userArray.length;i++){

if(userArray[i].getFirstName().equals(firstName)&&userArray[i].getLastName().equals(lastName)){

obj = userArray[i];

break;

}

else{

obj = null;

}

}

return obj;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Display Container Details

Container.java

public class Container extends Cargo{

private String containerNumber;

Container(){}

public Container(String containerNumber) {

this.containerNumber = containerNumber;

}

public String getContainerNumber() {

return containerNumber;

}

public void setContainerNumber(String containerNumber) {

this.containerNumber = containerNumber;

}

public void displayContainerDetails() {

System.out.format("%-20s %-15s %-15s %-15s %s\n", containerNumber, length, width, height, weight);

}

}

Cargo.java

public class Cargo {

protected Float length;

protected Float width;

protected Float height;

protected Double weight;

Cargo(){}

public Cargo(Float length, Float width, Float height, Double weight) {

this.length = length;

this.height = height;

this.width = width;

this.weight = weight;

}

public Float getLength() {

return length;

}

public void setLength(Float length) {

this.length = length;

}

public Float getHeight() {

return height;

}

public void setHeight(Float height) {

this.height = height;

}

public Float getWidth() {

return width;

}

public void setWidth(Float width) {

this.width = width;

}

public Double getWeight() {

return weight;

}

public void setWeight(Double weight) {

this.weight = weight;

}

}

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader buff = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of containers :");

int n = Integer.parseInt(buff.readLine());

Container[] container = new Container[n];

String details;

for(int i=0;i<n;i++) {

container[i] = new Container();

System.out.println("Enter the container "+(i+1)+" details :");

details = buff.readLine();

String arr[] = details.split(",");

container[i].setContainerNumber(arr[0]);

container[i].setLength(Float.parseFloat(arr[1]));

container[i].setWidth(Float.parseFloat(arr[2]));

container[i].setHeight(Float.parseFloat(arr[3]));

container[i].setWeight(Double.parseDouble(arr[4]));

}

System.out.format("Container details are\n%-20s %-15s %-15s %-15s %s\n","Container Number","Length","Width","Height","Weight");

for(int i=0;i<n;i++) {

container[i].displayContainerDetails();

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3. Abstract Class - ShipmentEntity

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws NumberFormatException, IOException {

ShipmentEntity shipmentEntity = null;

BufferedReader buf = new BufferedReader(new InputStreamReader(System.in));

System.out.println("1. Customer Shipment\n2. Company Shipment\nEnter your choice :");

Integer choice = Integer.parseInt(buf.readLine());

if(choice <= 2){

System.out.println("Enter the name :");

String name = buf.readLine();

System.out.println("Enter the weight :");

Double weight = Double.parseDouble(buf.readLine());

System.out.println("Enter the quantity :");

Integer quantity = Integer.parseInt(buf.readLine());

System.out.println("Enter the shipment transfer cost :");

Double transferCost = Double.parseDouble(buf.readLine());

System.out.println("Enter the maximum shipment capacity :");

Double maxShipmentCapacity = Double.parseDouble(buf.readLine());

switch(choice) {

case 1:System.out.println("Enter the referal fee :");

Double referalFee = Double.parseDouble(buf.readLine());

shipmentEntity = new CustomerShipment(name, weight, quantity, transferCost, maxShipmentCapacity, referalFee);

shipmentEntity.calculateCost();

shipmentEntity.operatingCapacity();

break;

case 2:System.out.println("Enter the tax percentage:");

int tax = Integer.parseInt(buf.readLine());

shipmentEntity = new CompanyShipment(name, weight, quantity, transferCost, maxShipmentCapacity, tax);

shipmentEntity.calculateCost();

shipmentEntity.operatingCapacity();

break;

}

}

else{

System.out.println("Invalid Input");

}

}

}

CompanyShipment.java

public class CompanyShipment extends ShipmentEntity{

private Integer tax;

public CompanyShipment(String name, Double weight, Integer quantity, Double transferCost, Double maxShipmentCapacity,Integer tax) {

super(name, weight, quantity, transferCost , maxShipmentCapacity);

this.tax = tax;

}

public int getTax() {

return tax;

}

public void setTax(int tax) {

this.tax = tax;

}

public void calculateCost(){

Double taxamount = (weight \* quantity \* transferCost) \* tax /100;

Double cost = weight \* quantity \* transferCost + taxamount;

System.out.println("Cost for the shipment is "+ cost);

}

public void operatingCapacity(){

if( maxShipmentCapacity < weight \* quantity) {

System.out.println("The shipment is not within the shipping capacity of the company");

}

else {

System.out.println("The shipment is within the shipping capacity of the company");

}

}

}

Customershipment.java

public class CustomerShipment extends ShipmentEntity{

private Double referalFee;

public CustomerShipment(String name, Double weight, Integer quantity, Double transferCost,Double maxShipmentCapacity, Double referalFee) {

super(name, weight, quantity, transferCost, maxShipmentCapacity);

this.referalFee = referalFee;

}

public Double getReferalFee() {

return referalFee;

}

public void setReferalFee(Double referalFee) {

this.referalFee = referalFee;

}

public void calculateCost(){

Double cost = weight \* quantity \* transferCost + referalFee;

System.out.println("Cost for the shipment is "+ cost);

}

public void operatingCapacity(){

if( maxShipmentCapacity < weight \* quantity) {

System.out.println("The shipment is not within the shipping capacity of the agent");

}

else {

System.out.println("The shipment is within the shipping capacity of the agent");

}

}

}

ShipmentEntity.java

public abstract class ShipmentEntity {

protected String name;

protected Double weight;

protected Integer quantity;

protected Double transferCost;

protected Double maxShipmentCapacity;

public ShipmentEntity(String name, Double weight, Integer quantity, Double transferCost,Double maxShipmentCapacity) {

this.name = name;

this.weight = weight;

this.quantity = quantity;

this.transferCost = transferCost;

this.maxShipmentCapacity = maxShipmentCapacity;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Double getWeight() {

return weight;

}

public void setWeight(Double weight) {

this.weight = weight;

}

public Integer getQuantity() {

return quantity;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

public Double getTransferCost() {

return transferCost;

}

public void setTransferCost(Double transferCost) {

this.transferCost = transferCost;

}

public Double getMaxShipmentCapacity() {

return maxShipmentCapacity;

}

public void setMaxShipmentCapacity(Double maxShipmentCapacity) {

this.maxShipmentCapacity = maxShipmentCapacity;

}

abstract void calculateCost();

abstract void operatingCapacity();

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4. Display Shipment Details

Shipment.java

public class Shipment {

int id;

String name;

String arrivalPort;

String departurePort;

ShipmentStatus status;

public Shipment() {}

public Shipment(int id,String name, String arrivalPort,String departurePort,ShipmentStatus status) {

this.id = id;

this.name = name;

this.arrivalPort = arrivalPort;

this.departurePort = departurePort;

this.status = status;

}

public int getid() {

return id;

}

public void setid(int id) {

this.id = id;

}

public String getname() {

return name;

}

public void setname(String name) {

this.name = name;

}

public String getarrivalPort () {

return arrivalPort;

}

public void setarrivalPort(String arrivalPort) {

this.arrivalPort = arrivalPort;

}

public String getdeparturePort () {

return departurePort;

}

public void setdeparturePort(String departurePort) {

this.departurePort = departurePort;

}

public ShipmentStatus getstatus() {

return status;

}

public void setstatus(ShipmentStatus status) {

this.status = status;

}

}

IExport Data.java

public interface IExportData {

void convertToJSON(Shipment[] shipmentObj);

void convertToCSV(Shipment[] shipmentObj);

}

ShipmentBO.java

public class ShipmentBO implements IExportData {

public void convertToJSON(Shipment[] shipmentObj) {

for(int i = 0; i < shipmentObj.length; i++) {

System.out.println("[");

System.out.println("{");

System.out.println("\"id\":"+shipmentObj[i].getid());

System.out.println("\"name\":\""+shipmentObj[i].getname()+"\"");

System.out.println("\"arrivalName\":\""+shipmentObj[i].getarrivalPort()+"\"");

System.out.println("\"departureName\":\""+shipmentObj[i].getdeparturePort()+"\"");

System.out.println("\"shipmentCode\":\""+shipmentObj[i].getstatus().getcode()+"\"");

System.out.println("\"shipmentName\":\""+shipmentObj[i].getstatus().getname()+"\"");

System.out.println("}");

System.out.println("]");

}

}

public void convertToCSV(Shipment[] shipmentObj) {

for(int i = 0; i < shipmentObj.length; i++) {

System.out.println(shipmentObj[i].getid()+","+shipmentObj[i].getname()+","+shipmentObj[i].getarrivalPort()+","+shipmentObj[i].getdeparturePort()+","+shipmentObj[i].getstatus().getcode()+","+shipmentObj[i].getstatus().getname());

}

}

}

Shipmentstatus.java

public class ShipmentStatus {

String code;

String name;

public ShipmentStatus() {}

public ShipmentStatus(String code,String name) {

this.code = code;

this.name = name;

}

public String getname() {

return name;

}

public void setname(String name) {

this.name = name;

}

public String getcode() {

return code;

}

public void setcode(String code) {

this.code = code;

}

}

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader buf=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of shipment");

int count = Integer.parseInt(buf.readLine());

String details;

Shipment[] sobj = new Shipment[count];

for(int i = 0; i< count; i++ ) {

System.out.println("Enter Shipment "+(i+1)+" details");

details = buf.readLine();

String Arr[] = details.split(",");

ShipmentStatus ss = new ShipmentStatus(Arr[4], Arr[5]);

sobj[i] = new Shipment(Integer.parseInt(Arr[0]), Arr[1], Arr[2], Arr[3], ss);

}

System.out.println("Enter the type to display the data (JSON/CSV)");

String Choice = buf.readLine();

ShipmentBO bo = new ShipmentBO();

if(Choice.equalsIgnoreCase("JSON")) {

System.out.println("JSON Data:");

bo.convertToJSON(sobj);

}

else if(Choice.equalsIgnoreCase("CSV")) {

System.out.println("CSV Data:");

bo.convertToCSV(sobj);

}

buf.close();

}

}

------------------------------------------------------------------------------------------------------------------------------------------

**5.b.\*\*\*\*\*\*\*\*OOPS Concept CC\*\*\*\*\*\*\***

TransGlobal Shipping Company- Single Inheritance concept

1. Single Inheritance

ShipmentEntity.java

public class ShipmentEntity {

private String name;

private String accountNumber;

private String address;

public ShipmentEntity(String name, String accountNumber, String address) {

this.name = name;

this.accountNumber = accountNumber;

this.address = address;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAccountNumber() {

return accountNumber;

}

public void setAccountNumber(String accountNumber) {

this.accountNumber = accountNumber;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

}

Customer.java

public class Customer extends ShipmentEntity {

private String identificationNumber;

public Customer(String name, String accountNumber, String address,String identificationNumber) {

super(name, accountNumber, address);

this.identificationNumber = identificationNumber;

}

public String getIdentificationNumber() {

return identificationNumber;

}

public void setIdentificationNumber(String identificationNumber) {

this.identificationNumber = identificationNumber;

}

void displayCustomerDetails()

{

System.out.println("customer = '{");

System.out.println("\"name\" : \""+getName()+"\",");

System.out.println("\"account-number\" : \""+getAccountNumber()+"\",");

System.out.println("\"address\" : \""+getAddress()+"\",");

System.out.println("\"identification-number\" : \""+getIdentificationNumber()+"\"");

System.out.println("}';");

}

}

Agent.java

public class Agent extends ShipmentEntity {

private String fmc;

private String iata;

public Agent(String name, String accountNumber, String address,String fmc, String iata ) {

super(name, accountNumber, address);

this.fmc = fmc;

this.iata = iata;

}

public String getFmc() {

return fmc;

}

public void setFmc(String fmc) {

this.fmc = fmc;

}

public String getIata() {

return iata;

}

public void setIata(String iata) {

this.iata = iata;

}

void displayAgentDetails()

{

System.out.println("agent = '{");

System.out.println("\"name\" : \""+getName()+"\",");

System.out.println("\"account-number\" : \""+getAccountNumber()+"\",");

System.out.println("\"address\" : \""+getAddress()+"\",");

System.out.println("\"fmc\" : \""+getFmc()+"\",");

System.out.println("\"iata\" : \""+getIata()+"\"");

System.out.println("}';");

}

}

Main.java

import java.io.\*;

public class Main {

public static void main(String[] args) throws Exception {

int choice;

String name,accountNumber,address,fmc,iata,identificationNumber;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("1. Agent\n2. Customer\nEnter the choice :");

choice = Integer.parseInt(br.readLine());

System.out.println("Enter the name :");

name = br.readLine();

System.out.println("Enter the account number :");

accountNumber = br.readLine();

System.out.println("Enter the address :");

address = br.readLine();

switch(choice)

{

case 1:

System.out.println("Enter the fmc :");

fmc = br.readLine();

System.out.println("Enter the iata code :");

iata = br.readLine();

System.out.println("JSON Format :");

Agent agentobj = new Agent(name, accountNumber, address, fmc, iata);

agentobj.displayAgentDetails();

break;

case 2:

System.out.println("Enter the identification number :");

identificationNumber = br.readLine();

System.out.println("JSON Format :");

Customer Custobj = new Customer(name, accountNumber, address, identificationNumber);

Custobj.displayCustomerDetails();

break;

}

}

}

------------------------------------------------------------------------------------------------------------------------------------------

**6.a.\*\*\*\*\*\*Streams and writers LC\*\*\*\***

1. User Details

Main.java

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String[] args) throws IOException {

//Creating the file object

File dir = new File(".");

File fin = new File(dir.getCanonicalPath() + File.separator + "input.txt");

//Construct InputStreamReader with parameter as file

FileInputStream fis = new FileInputStream(fin);

//Construct BufferedReader from InputStreamReader

BufferedReader br = new BufferedReader(new InputStreamReader(fis));

//Create an ArrayList

List<User> UserList = new ArrayList<User>();

//Reading the data from the sheet

String line = null;

while ((line = br.readLine()) != null) {

UserList.add(new User(Long.parseLong(line.substring(0, 5)),line.substring(5, 15),line.substring(15, 25),line.substring(25, 35),line.substring(35, 43),line.substring(43, 53)));

}

//Closing the BufferedReader

br.close();

System.out.println("id First Name Last Name Username Password Mobile Number");

for(User st:UserList){

System.out.println(st.toString());

}

}

}

User.java

public class User {

private Long id;

private String firstName;

private String lastName;

private String username;

private String password;

private String mobileNumber;

public User(Long id,String firstName,String lastName,String username,String password,String mobileNumber) {

this.id = id;

this.firstName = firstName;

this.lastName = lastName;

this.username = username;

this.password = password;

this.mobileNumber = mobileNumber;

}

@Override

public String toString() {

return String.format("%-10s %-15s %-15s %-15s %-15s %s",id,firstName,lastName,username,password,mobileNumber);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Commodities Details

ShipmentBO.java

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

public class ShipmentBO {

public List<Commodity> readCommodity(String fileName) throws IOException {

//Creating the file object

File dir = new File(".");

File fin = new File(dir.getCanonicalPath() + File.separator + fileName);

//Construct InputStreamReader with parameter as file

FileInputStream fis = new FileInputStream(fin);

//Construct BufferedReader from InputStreamReader

BufferedReader br = new BufferedReader(new InputStreamReader(fis));

//Create an ArrayList

List<Commodity> CommodityList = new ArrayList<Commodity>();

//Reading the data from the sheet

String line = null;

String Data[];

Boolean flag;

while ((line = br.readLine()) != null) {

Data = line.split(",");

if(Data[3].equals("1"))

flag = true;

else

flag = false;

CommodityList.add(new Commodity(Long.parseLong(Data[0]), Double.parseDouble(Data[1]), Integer.parseInt(Data[2]),flag , Long.parseLong(Data[4])));

}

//Closing the BufferedReader

br.close();

int totalquantity = 0;

double totalweight = 0;

for(Commodity st:CommodityList){

totalquantity = st.quantity + totalquantity;

totalweight = st.weight + totalweight;

}

System.out.println("Total Weight: "+totalweight);

System.out.println("Total Quantity: "+totalquantity);

return CommodityList;

}

void checkCommodity(List<Commodity> commodityList) {

boolean flag = false;

for(Commodity st:commodityList){

if(st.hazardous == true)

flag = true;

}

if(flag == true)

System.out.println("Hazardous Commodity Found!");

}

}

Commodity.java

public class Commodity {

Long commodityId;

Double weight;

Integer quantity;

Boolean hazardous;

Long containerId;

public Commodity(Long commodityId,Double weight, Integer quantity,Boolean hazardous,Long containerId) {

this.commodityId = commodityId;

this.weight = weight;

this.quantity = quantity;

this.hazardous = hazardous;

this.containerId = containerId;

}

}

Main.java

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String[] args) throws IOException {

String fileName = "input.txt";

ShipmentBO BOobj = new ShipmentBO();

List<Commodity> CommodityList = new ArrayList<Commodity>();

CommodityList = BOobj.readCommodity(fileName);

BOobj.checkCommodity(CommodityList);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Display User Details

Role.java

public class Role {

Long roleId;

String name;

public Role(Long roleId,String name) {

this.roleId = roleId;

this.name = name;

}

}

user.java

public class User {

Long id;

String firstName;

String lastName;

String username;

String password;

String mobileNumber;

Role role;

public User(Long id,String firstName,String lastName,String username,String password,String mobileNumber,Role role) {

this.id = id;

this.firstName = firstName;

this.lastName = lastName;

this.username = username;

this.password = password;

this.mobileNumber = mobileNumber;

this.role = role;

}

}

UserBO.java

import java.io.FileWriter;

import java.io.IOException;

import java.util.List;

public class UserBO {

public void saveAllUser(List<User> userList,String fileName) throws IOException {

FileWriter fw = new FileWriter(fileName);

for(User st:userList){ fw.write(st.id+","+st.firstName+","+st.lastName+","+st.username+","+st.password+","+st.mobileNumber+","+st.role.roleId+","+st.role.name+"\n");

}

fw.close();

}

}

Main.java

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class Main {

public static void main(String[] args) throws IOException {

Scanner scanner = new Scanner(System.in);

int Count;

String fileName = "output.txt";

System.out.println("Enter the number of users:");

Count = Integer.parseInt(scanner.nextLine());

Long id;

String firstName;

String lastName;

String username;

String password;

String mobileNumber;

String rolename;

Long roleId = (long) 100;

List<User> list = new ArrayList<User>();

for(int i = 0;i<Count;i++) {

System.out.println("Enter the user details:");

System.out.println("Enter the User ID");

id = Long.parseLong(scanner.nextLine());

System.out.println("Enter the first name");

firstName = scanner.nextLine();

System.out.println("Enter the last name");

lastName = scanner.nextLine();

System.out.println("Enter the username");

username = scanner.nextLine();

System.out.println("Enter the password");

password = scanner.nextLine();

System.out.println("Enter the mobile number");

mobileNumber = scanner.nextLine();

System.out.println("Enter the role name");

rolename = scanner.nextLine();

list.add(new User(id,firstName,lastName,username,password,mobileNumber,new Role(roleId++, rolename)));

}

UserBO obj = new UserBO();

obj.saveAllUser(list, fileName);

scanner.close();

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6.a.\*\*\*\*\*\*Streams and writers CC\*\*\*\***

Display Rejected Payments

Invoice.java

public class Invoice {

String invoice\_number, customer\_name,cheque\_number,cheque\_date,bank\_name,amount,status;

public Invoice(String invoice\_number,String customer\_name,String cheque\_number,String cheque\_date,String bank\_name,String amount,String status) {

this.invoice\_number = invoice\_number;

this.customer\_name = customer\_name;

this.cheque\_number = cheque\_number;

this.cheque\_date = cheque\_date;

this.bank\_name = bank\_name;

this.amount = amount;

this.status = status;

}

}

InvoiceBO.java

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

public class InvoiceBO {

void findAllRejectedPayments(String inputfilename,String outputfile) throws IOException {

//Creating the file object

File dir = new File(".");

File fin = new File(dir.getCanonicalPath() + File.separator + "input.txt");

//Construct InputStreamReader with parameter as file

FileInputStream fis = new FileInputStream(fin);

//Construct BufferedReader from InputStreamReader

BufferedReader br = new BufferedReader(new InputStreamReader(fis));

//Create an ArrayList

List<Invoice> InvoiceList = new ArrayList<Invoice>();

//Reading the data from the sheet

String line = null;

String [] Data;

while ((line = br.readLine()) != null) {

Data = line.split(",");

InvoiceList.add(new Invoice(Data[0],Data[1],Data[2],Data[3],Data[4],Data[5],Data[6]));

}

//Closing the BufferedReader

br.close();

FileWriter fw = new FileWriter(outputfile);

for(Invoice st:InvoiceList){

if(st.status.equalsIgnoreCase("Rejected")) { fw.write(st.invoice\_number+","+st.cheque\_number+","+st.customer\_name+","+st.amount+","+st.status+"\n");

}

}

fw.close();

}

}

Main.java

import java.io.IOException;

public class Main {

public static void main(String[] args) throws IOException {

InvoiceBO obj = new InvoiceBO();

obj.findAllRejectedPayments("input.txt", "output.txt");

}

}

------------------------------------------------------------------------------------------------------------------------------------------

**7.a.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*String LC\*\*\*\*\*\*\*\*\*\*\***

1. Find virus from words using contains

Main.java

public class Main {

public static void main(String args[]){

String text,virus;

java.util.Scanner sc = new java.util.Scanner(System.in);

System.out.println("Enter the file content:");

text = sc.nextLine();

System.out.println("Enter the virus keyword:");

virus = sc.nextLine();

if(text.contains(virus.trim())) {

System.out.println("Virus "+virus.trim()+" is present");

}

else {

System.out.println("Virus "+virus.trim()+" is not present");

}

sc.close();

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. String equals and equalsIgnoreCase

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the document 1 content:");

String doc1 = br.readLine();

System.out.println("Enter the document 2 content:");

String doc2 = br.readLine();

if(doc1.equals(doc2)) {

System.out.println("Green");

}

else if(doc1.equalsIgnoreCase(doc2)) {

System.out.println("Blue");

}

else if(doc1.replace(" ", "").equals(doc2.replace(" ", ""))) {

System.out.println("Orange");

}

else {

System.out.println("Red");

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3. Email Validation II

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the email id");

String emailid = br.readLine();

int length = emailid.length();

String substr = emailid.substring(length-3,length);

if(substr.contains("com") | substr.contains("biz") | substr.contains("net") | substr.contains("org")) {

System.out.println("Valid domain");

}

else {

System.out.println("Not a valid domain");

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4. Username

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the username");

String name = br.readLine();

System.out.println("Formatted username");

System.out.println(name.toLowerCase());

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5. Validating booking id

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the generated booking id");

String bookingId = br.readLine();

Pattern regexPattern = Pattern.compile("^\\d+$");

Matcher regMatcher = regexPattern.matcher(bookingId);

if(regMatcher.matches()==true) {

System.out.println("Generated booking id is valid");

}

else {

System.out.println("Generated booking id is not valid");

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6. Invoice Code

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the invoice code");

String invoicecode = br.readLine();

invoicecode = invoicecode.replace("CU", "CUR");

invoicecode = invoicecode.replace("AT", "AGT");

invoicecode = invoicecode.replace("CY", "CMY");

int length = invoicecode.length();

StringBuilder builder = new StringBuilder();

builder.append(invoicecode);

int a = 3;

for(int i = length;i<8;i++) {

builder.insert(a, 0);

a++;

}

System.out.println("Formated Code :");

System.out.println(builder);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7.b.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*String CC\*\*\*\*\*\*\*\*\*\*\***

**Invoice number validation**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the invoice code");

String invoicecode = br.readLine();

Pattern regexPattern = Pattern.compile("(CU)(\\d+){3}(AT)(\\d+){2}(S)(\\d+){1}(E)(\\d+){2}");

Matcher regMatcher = regexPattern.matcher(invoicecode);

if(regMatcher.matches()==true) {

System.out.println(invoicecode+" is valid");

}

else {

System.out.println(invoicecode+" is not valid");

}

}

}

------------------------------------------------------------------------------------------------------------------------------------------

**8.a.\*\*\*\*\*\*\*Threads LC\*\*\*\***

**Threads - i**

Cargo.java

public class Cargo {

private Long id;

private Integer length;

private Integer width;

private Integer weight;

private String cargoType;

private String storageType; //Dry Storage or Cold Storage

public static String DRY\_STORAGE = "DRY";

public static String COLD\_STORAGE = "COLD";

public Cargo(String value) {

String val[] = value.split(",");

id = Long.parseLong(val[0]);

length = Integer.parseInt(val[1]);

width = Integer.parseInt(val[2]);

weight = Integer.parseInt(val[3]);

cargoType = val[4];

storageType = val[5];

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Integer getLength() {

return length;

}

public void setLength(Integer length) {

this.length = length;

}

public Integer getWidth() {

return width;

}

public void setWidth(Integer width) {

this.width = width;

}

public Integer getWeight() {

return weight;

}

public void setWeight(Integer weight) {

this.weight = weight;

}

public String getCargoType() {

return cargoType;

}

public void setCargoType(String cargoType) {

this.cargoType = cargoType;

}

public String getStorageType() {

return storageType;

}

public void setStorageType(String storageType) {

this.storageType = storageType;

}

}

ShippingCostThread.java

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

public class ShippingCostThread extends Thread {

List<Cargo> cargoList = new ArrayList<Cargo>();

List<Double> priceList = new ArrayList<Double>();

public void run() {

double price,weight;

Iterator<Cargo> iterator = cargoList.iterator();

while (iterator.hasNext()) {

Cargo obj;

obj = iterator.next();

weight = obj.getWeight();

if(obj.getStorageType().equalsIgnoreCase(Cargo.COLD\_STORAGE)) {

price = weight\*1.85;

}else{

price = weight\*0.90;

}

priceList.add(price);

}

Main.displayPrice(priceList);

}

public List<Cargo> getCargoList() {

return cargoList;

}

public void setCargoList(List<Cargo> cargoList) {

this.cargoList = cargoList;

}

public List<Double> getPriceList() {

return priceList;

}

public void setPriceList(List<Double> priceList) {

this.priceList = priceList;

}

}

main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String args[]) throws IOException, InterruptedException {

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of Cargo:");

int numberOfCargo = Integer.parseInt(reader.readLine());

System.out.println("Enter cargo details (id,length,width,weight,cargo type,storage type):");

List<Cargo> cargoList = new ArrayList<Cargo>();

for(int i=0;i<numberOfCargo;i++) {

cargoList.add(new Cargo(reader.readLine()));

}

ShippingCostThread obj = new ShippingCostThread();

obj.setCargoList(cargoList);

obj.start();

}

public static void displayPrice(List<Double> list) {

System.out.println("Price List:");

for(Double st:list){

System.out.println(st); }

}

}

------------------------------------------------------------------------------------------------------------------------------------------**8.b.\*\*\*\*\*\*\*Threads CC\*\*\*\***

Threads - iv

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String args[]) throws IOException, InterruptedException {

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter number of notification:");

int numberOfMsg = Integer.parseInt(reader.readLine());

List<String> notification = new ArrayList<String>();

for(int i=0;i<numberOfMsg;i++) {

String line = reader.readLine();

notification.add(line);

}

System.out.println("Enter number of threads:");

int numberOfThreads = Integer.parseInt(reader.readLine());

Thread[] threads = new Thread[numberOfThreads];

if(numberOfThreads==3)

{

NotificationManager l=NotificationManager.getInstance();

for (int i=0;i<notification.size()-1;i++)

{

String val[]=notification.get(i).split(",");

int code=Integer.parseInt(val[1]);

l.sendMessage(val[0],code);

}

} else

{

NotificationManager l=NotificationManager.getInstance();

for (String b:notification)

{

String val[]=b.split(",");

int code=Integer.parseInt(val[1]);

l.sendMessage(val[0],code);

}

}

}

}

NotificationThread.java

import java.util.Iterator;

import java.util.List;

public class NotificationThread

{

List<String> notification;

public void run() {}

public void setNotification(List<String> notification) {

this.notification = notification;

}

}

NotificationManager.java

import java.util.HashMap;

import java.util.Map;

public class NotificationManager {

static NotificationManager instance;

static Map<Integer,String> messages = new HashMap<Integer,String>();

static {

messages.put(100,"Arrived at destination");

messages.put(101,"Attempted delivery");

messages.put(102,"Awaiting unloading at consignee");

messages.put(103,"Delayed en route to destination");

messages.put(104,"Delivered part short");

messages.put(105,"En route to destination");

messages.put(106,"En route to Hawaii");

messages.put(107,"Late but no possible delay notification sent");

messages.put(108,"Out for delivery");

}

private NotificationManager() {}

public static NotificationManager getInstance() {

if(instance == null) {

instance = new NotificationManager();

}

return instance;

}

public void sendMessage(String number,Integer code) {

System.out.println("Sending SMS TO "+number+" MSG:"+messages.get(code));

}}