DoSelect practice sol

1 => List of Operations;

```
import java.util.*;
class ArrayListOps {
      public ArrayList<Integer>makeArrayListInt(int n){
          ArrayList<Integer> list=new ArrayList<>();

          for(int i=0; i<n; i++)
          {
                        list.add(0)    ;
                                 }

          return list;
}
public ArrayList<Integer>reverseList(ArrayList<Integer> list){

          Collections.reverse(list);

          return list;
}
public ArrayList<Integer>changeList(ArrayList<Integer> list, int m, int n){

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

                    if(list.get(i) == m){

                              list.set(i, n);
                    }
          }
          return list;
}
}
public class Source{
          public static void main(String[] args) {
          }
}
```

----

2 => Mobile Shop;

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

class Mobile{
        // Write your code here..
          HashMap<String, ArrayList<String>> mobiles = new HashMap<>();
          public String    addMobile(String company, String model){
                    if(!mobiles.containsKey(company)){
                              ArrayList<String> list = new ArrayList<>();
                              list.add(model);
                              mobiles.put(company, list);
                              }else{
                                        ArrayList<String> list = mobiles.get(company);
                                        list.add(model);
                                        }
                                        return    "model successfully added";
                                        }
                                        public ArrayList<String>getModels(String company){
                                                  ArrayList<String> s=mobiles.get(company);
                                                  return s;
          }
          public String buyMobile(String company, String model){
                    if(mobiles.containsKey(company)){
                              ArrayList<String> p=mobiles.get(company);
                              if(p.contains(model)){
                                        p.remove(model);
                                        return "mobile sold successfully";
                              }
                    }
                    return    "item not available";
          }
}

public class Source {
          public static void main(String args[] ) throws Exception {
                    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
          }
}
                              mobiles.put(company, list);

                    }
                    else{

                              ArrayList<String> list = mobiles.get(company);
                    list.add(model);

            }

            return    "model successfully added";
        }
        public ArrayList<String>getModels(String company){
            ArrayList<String> s=mobiles.get(company);



            return s;
        }
        public String buyMobile(String company, String model){
            if(mobiles.containsKey(company)){

                ArrayList<String> p=mobiles.get(company);

                if(p.contains(model)){

                    p.remove(model);

                    return "mobile sold successfully";
                }
            }

            return    "item not available";
        }
}

public class Source {
        public static void main(String args[] ) throws Exception {
            /* Enter your code here. Read input from STDIN. Print output to STDOUT */
        }
}
```

----

3 => Handling Stuff

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;
class Activity{
    String string1;
    String string2;
    String operator;
    public Activity(String string1,String string2,String operator){
        this.string1=string1;
        this.string2= string2;
        this.operator= operator;
    }
}
class Source{
    public static String handleException(Activity a){
        try{
            if(a.string1==null||a.string2==null){
                throw new NullPointerException("Null values found");
            }
            if(!("+".equals(a.operator)||"-".equals(a.operator))){
                throw new Exception("Default Exception: Operator is not valid");
            }
            return "No Exception Found";
        }
        catch(NullPointerException e){
            return e.getMessage();
        }
        catch(Exception e){
            return e.getMessage();
        }
    }
    public static String doOperation (Activity a ){
        switch(a.operator){
            case "+":
            return a.string1 + a.string2;
            case "-":
            return a.string1.replace(a.string2,"");
            default:
            return "Invalid operator";
        }
    }
}
    class Main {
    public static void main(String[] args){
            Activity activity1=new Activity("hello","world","+");
        Activity activity2=new Activity("helloworld","","-");
        Activity activity3=new Activity("hello","world","*");

        System.out.println("handleException(activity1):"+Source.handleException(activity1));
        System.out.println("doOperation(activity1):"+Source.doOperation(activity1));
        System.out.println("handleException(activity2):"+Source.handleException(activity2));
        System.out.println("doOperation(activity2):"+Source.doOperation(activity2));
        System.out.println("handleException(activity3):"+Source.handleException(activity3));
        System.out.println("doOperation(activity3):"+Source.doOperation(activity3));


    }
}
```

----

4 => Job Agency

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

class CompanyJobRepository {
        static String getJobPrediction(int age, String highestQualification) throws
        NotEligibleException
        {
                if(age<19)
                  throw new NotEligibleException("You are underage for any job");
                else if(age>=21&&highestQualification.equals("B.E"))
                {
                        return "We have openings for junior developer";
                }
                else
                if(age>=26&&(highestQualification.equals("M.S")||highestQualification.equals("PhD")))
                {
                        return "We have openings for senior developer";
                }
                else

        if(age>=19&&(!(highestQualification.equals("B.E")||highestQualification.equals("M.S")||highestQualific
ation.equals ("PhD"))))
                {
                        throw new NotEligibleException("We do not have any job that matches your
qualifications");
                }
                return "Sorry we have no openings for now";
        }
}

public class Source {
        public String searchForJob(int age, String highestQualification)throws NotEligibleException
        {
                String s= "";
                        if(age>=200||age<=0)
                        {
                                throw new NotEligibleException("The age entered is not typical for a human being");
                        }
                        else{
                                s=CompanyJobRepository.getJobPrediction(age,highestQualification);
                        }
                  return s;
        }
        public static void main(String args[] )    {
                /* Enter your code here. Read input from STDIN. Print output to STDOUT */
        }
}
class NotEligibleException extends Exception{
        NotEligibleException(String msg){
                super(msg);
        }
}
```

5=> Employee Verification Code

```
import java.util.*;
import java.util.function.*;
import java.util.stream.Stream;
import java.util.stream.Collectors;
class Employee {
       String name;
       int salary;
       Employee(String name, int salary){
              this.name=name;
              this.salary=salary;
       }
       public String getName(){
              return name;
       }
       public void setName(String name){
              this.name=name;
       }
       public int getSalary(){
```

```
		return salary;
	}
	public void setSalary(int salary){
		this.salary=salary;
	}

	@Override
	public String toString() {
		StringBuilder sb = new StringBuilder("<");
		sb.append("name: ");
		sb.append(name);
		sb.append(" salary: ");
		sb.append("" + salary+">");
		return sb.toString();

	}
}

class EmployeeInfo{
	enum SortMethod {
		BYNAME, BYSALARY;
	}
	public List<Employee> sort(List<Employee> emps, final SortMethod method){
		if(method.name().equals("BYNAME")){
			List<Employee> result= emps.stream()
			.sorted(Comparator.comparing(Employee::getName))
			.collect(Collectors.toList());
			return result;
		}
		else{
			List<Employee>result= emps.stream()
			.sorted(Comparator.comparing(Employee::getSalary)
			.thenComparing(Employee::getName))
			.collect(Collectors.toList());
		return result;
		}
	}
	public boolean isCharacterPresentInAllNames(Collection<Employee> entities, String character){
		boolean bool=entities.stream()
		.allMatch((s)->s.getName().contains(character));
		return bool;
	}
}
```

----

6 => Hiring On

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;
import java.util.stream.Collectors;
class Candidate {
        private int age;
        private String name;
        private int id;
        private String gender;
        private int yearOfJoining;
        private double salary;
        private String department;
        public Candidate(int id, String name, int age, String gender, String department, int yearOfJoining,
double salary) {
                this.age = age;
                this.name = name;
                this.id = id;
                this.gender = gender;
                this.yearOfJoining = yearOfJoining;
                this.salary = salary;
                this.department = department;
                }
      // Getter and setter methods for the private data members
        public int getAge() {
                return age;
                }
                public void setAge(int age) {
                        this.age = age;
                        }
                        public String getName() {
                                return name;
                                }
                                public void setName(String name) {
                                        this.name = name;
                                        }
                                        public int getId() {
                                                return id;
                                                }
                                                public void setId(int id) {
                                                        this.id = id;
                                                        }
                                                        public String getGender() {
                                                                return gender;
                                                                }
                                                                public void setGender(String gender) {
```

```
                                                            this.gender = gender;
                                                            }
                                                            public int getYearOfJoining() {
                                                                    return yearOfJoining;
                                                                    }
                                                                    public void setYearOfJoining(int
yearOfJoining) {
                                                                            this.yearOfJoining = yearOfJoining;
                                                                            }
                                                                            public double getSalary() {
                                                                                    return salary;
                                                                                    }
                                                                                    public void setSalary(double
salary) {
                                                                                            this.salary = salary;
                                                                                            }
                                                                                            public String
getDepartment() {
                                                                                                    return department;
                                                                                                    }
                                                                                                    public void
setDepartment(String department){

        this.department = department;
                                                                                                            }
@Override
public String toString() {
        return "Employee [id=" + id + ", name=" + name + ", age=" + age + ", gender=" + gender + ",
department=" + department + ", yearOfJoining=" + yearOfJoining + ", salary=" + salary + "]";
        }
        }
        class Implementation {
                public static Map<String, Long> getCount(List<Candidate> list) {
                        return list.stream()
                        .collect(Collectors.groupingBy(Candidate::getGender, Collectors.counting()));
                        }
                        public static Map<String, Double> getAverageAge(List<Candidate> list) {
                                return list.stream()
                                .collect(Collectors.groupingBy(Candidate::getGender,
Collectors.averagingInt(Candidate::getAge)));
                                }
                                public static Map<String, Long> countCandidatesDepartmentWise(List<Candidate>
list) {
                                        return list.stream()
                                        .collect(Collectors.groupingBy(Candidate::getDepartment,
Collectors.counting()));
                                        }
                                        public static Optional<Candidate>
getYoungestCandidateDetails(List<Candidate> list){
						return list.stream()
						.filter(candidate -> candidate.getGender().equals("Male") &&
candidate.getDepartment().equals("Product Development"))
						.min(Comparator.comparingInt(Candidate::getAge));
						}
						}
class Source {
	public static void main(String[] args) {
		List<Candidate> list = new ArrayList<>();
		list.add(new Candidate(111, "Jiya Brein", 32, "Female", "HR", 2011, 25000.0));
		list.add(new Candidate(144, "Scarlet Jhonson", 28, "Male", "Product Development", 2014,
32500.0));
		// Get count of male and female employees
		Map<String, Long> countMap = Implementation.getCount(list);
		System.out.println(countMap);
		// Get average age of male and female employees
		Map<String, Double> avgAgeMap = Implementation.getAverageAge(list);
		System.out.println(avgAgeMap);
		// Count employees in each department
		Map<String, Long> deptCountMap = Implementation.countCandidatesDepartmentWise(list);
		System.out.println(deptCountMap);
		// Get details of the youngest male employee in Product Development
		Optional<Candidate> youngestCandidate = Implementation.getYoungestCandidateDetails(list);
		System.out.println(youngestCandidate);
		}
}
```

----

7 => Email operations

```
import java.util.stream.Collectors;
class Email{
// Implement Email Class according to the specifiaction.
Header header;
String body;
String greetings;
public Email(Header header, String body, String greetings){
	this.header = header;
```

```
        this.body = body;
        this.greetings = greetings;
        }
}
class Header{
// Implemet the Header Class according to the specifiaction.
String from;
String to;
public Header(String from, String to){
        this.from = from;
        this.to = to;
}
}



class EmailOperations{
        public int emailVerify(Email e){
                int ans = 0;
                boolean check1 = Pattern.matches("^[a-zA-Z_]+@[a-zA-Z.]+$", e.header.from);
                boolean check2 = Pattern.matches("^[a-zA-Z_]+@[a-zA-Z.]+$", e.header.to);
                if(check1 && check2){
                        ans = 2;
                }
                else if(check1 || check2){
                        ans = 1;
                }
                return ans;
                }
                public String bodyEncryption(Email e){
                        String ans = "";
                        for(int i=0;i<e.body.length();i++){
                                if(e.body.charAt(i) == ' '){
                                        ans += e.body.charAt(i);
                                        }else{
                                                switch(e.body.charAt(i)){
                                                        case 'X': ans += 'A'; break;
                                                        case 'x': ans += 'a'; break;
                                                        case 'Y': ans += 'B'; break;
                                                        case 'y': ans += 'b'; break;
                                                        case 'Z': ans += 'C'; break;
                                                        case 'z': ans += 'c'; break;
                                                        default:ans += (char)(e.body.charAt(i) + 3);
                                                        }
                                                        }
                                                        }
                                                        return ans;
                                                        }
                                                        public String greetingMessage(Email e){
```

```
                                        String ans = "";
                                        String temp = e.header.from;
                                        for(int i=0;i<temp.length();i++){
                                                if(temp.charAt(i) == '@'){
                                                        break;
                                                        }else{
                                                                ans += temp.charAt(i);
                                                                }
                                                                }
                                                                return e.greetings + " " + ans;
                                                                                }
                                                                        }
```

----

8 => Validating Usersy

```
import java.util.*;
import java.lang.*;
import java.util.regex.*;
class TransactionParty {
    String seller;
    String buyer;
    public TransactionParty(String seller, String buyer){
        this.seller = seller;
        this.buyer = buyer;
        }
}

class Receipt{
    TransactionParty transactionParty;
    String productsQR;
    public Receipt(TransactionParty transactionParty, String productsQR){
        this.transactionParty = transactionParty;
        this.productsQR = productsQR;
        }
}

class GenerateReceipt{
      public int    verifyParty(Receipt r){
          boolean check1 = Pattern.matches("[A-Za-z][A-Za-z\\s'-]*[A-Za-z]", r.transactionParty.seller);
```

```
     boolean check2 = Pattern.matches("[A-Za-z][A-Za-z\\s'-]*[A-Za-z]", r.transactionParty.buyer);
     System.out.println(check1 + " " + check2);
     if(check1 && check2){
       return 2;
       }
       else if(check1 || check2){
         return 1;
         }
         return 0;
         }
         public String calcGST(Receipt r){
           int gst = 0;
           int add = 0;
           String word = "";
           String str = r.productsQR;
           for(int i=0;i<str.length();i++){
             if(str.charAt(i) == ','){
               add = Integer.parseInt(word);
               word = "";
               }
               else if(str.charAt(i) == '@'){
                 gst += (add * Integer.parseInt(word));
                 word = "";
                 }
                 else{
                   word += str.charAt(i);
                   }
                   }
                   gst += (add * Integer.parseInt(word));
                   gst *= 0.12;
                   return Integer.toString(gst);
                   }
                   }
class Source{
  public static void main(String[] args){

  }
}
```