

Lab Sheet 09

Answer

```
public class List {  
  
    public int maxSize;  
    public int position;  
    public Customer[] entry;  
  
    public List(int x){  
        maxSize=x;  
        position=-1;  
        entry=new Customer[maxSize];  
    }  
    public boolean isEmpty(){  
        return(position==-1);  
    }  
    public boolean isFull(){  
        return(position==maxSize-1);  
    }  
    public int listSize(){  
        return(position+1);  
    }  
    public void insertLast(Customer x){  
        if (isFull()) {  
            System.out.println("List is Already Full.");  
        }else{  
            entry[++position]=x;  
        }  
    }  
    public void insertList(int p, Customer x){  
        if (isFull()) {  
            System.out.println("List is Already Full.");  
        }else if (p<0||p>listSize()) {  
            System.out.println("Position is out of the List.");  
        }else{  
            for (int i = listSize(); i > p; i--) {  
                entry[i]=entry[i-1];  
            }  
            entry[p]=x;  
            position++;  
        }  
    }  
}
```

```
}  
public Customer deleteList(int p){  
    if (isEmpty()) {  
        System.out.println("List is Already Empty.");  
    }else if (p<0||p>listSize()) {  
        System.out.println("Position is out of the List.");  
    }else{  
        Customer y=entry[p];  
        for (int i = p; i < listSize()-1; i++) {  
            entry[i]=entry[i+1];  
            position--;  
            return y;  
        }  
    }  
    return null;  
}  
  
public Customer retrieveList(int p){  
    if (isEmpty()) {  
        System.out.println("List is Already Full.");  
    }else if (p<0||p>listSize()) {  
        System.out.println("Position is out of the List.");  
    }else{  
        return entry[p];  
    }  
    return null;  
}  
  
public void replaceList(int p,Customer x){  
    if (isEmpty()) {  
        System.out.println("List is Already Full.");  
    }else if (p<0||p>listSize()) {  
        System.out.println("Position is out of the List.");  
    }else{  
        entry[p]=x;  
    }  
}  
  
public void traverseList(){  
    for (int i = 0; i < listSize(); i++) {  
        System.out.println(entry[i]);  
    }  
}  
  
public double interestAmount(double x){  
    double amount;
```

```
        for (int i = 0; i < listSize(); i++) {
            if (retrieveList(i).accountBalance<250000) {
                amount=x*2.5/100;
                return amount;
            }else if (retrieveList(i).accountBalance>250000 &&
retrieveList(i).accountBalance<500000) {
                amount=x*5.0/100;
                return amount;
            }else if (retrieveList(i).accountBalance>500000 &&
retrieveList(i).accountBalance<750000) {
                amount=x*7.5/100;
                return amount;
            }else if (retrieveList(i).accountBalance>750000 &&
retrieveList(i).accountBalance<1000000) {
                amount=x*8.0/100;
                return amount;
            }else if (retrieveList(i).accountBalance>1000000 &&
retrieveList(i).accountBalance<1500000) {
                amount=x*9.5/100;
                return amount;
            }
        }

        return 0;
    }

    public void printinterestAmount(){
        for (int i = 0; i < listSize(); i++) {
            System.out.println(entry[i].accountNumber+"\t\t"+entry[i].customerName+"\t\t"+interestAmount(entry[i].accountBalance));
        }
    }

    public void totalAccountBalance(){
        double total;
        for (int i = 0; i < listSize(); i++) {
            total=entry[i].accountBalance+interestAmount(entry[i].accountBalance)
;
            System.out.println(entry[i].accountNumber+"\t\t"+entry[i].customerName+"\t\t"+total);
        }
    }

    public void quickSort(int low,int high){
        if (low<high) {
            int x=partition(low,high);
            quickSort(low, x-1);
            quickSort(x+1, high);
        }
    }
}
```

```
    }  
}  
private int partition(int low, int high) {  
    double pivot = entry[high].accountBalance +  
interestAmount(entry[high].accountBalance);  
    int i = low - 1;  
    for (int j = low; j < high; j++) {  
        if (entry[j].accountBalance + interestAmount(entry[j].accountBalance)  
<= pivot) {  
            i++;  
            Customer temp = entry[i];  
            entry[i] = entry[j];  
            entry[j] = temp;  
        }  
    }  
    Customer temp = entry[i + 1];  
    entry[i + 1] = entry[high];  
    entry[high] = temp;  
    return i + 1;  
}  
public void displaycustomerWithHightBalance(){  
    for (int i = 0; i < listSize(); i++) {  
        double  
totalBalance=entry[i].accountBalance+interestAmount(entry[i].accountBalance);  
        if (totalBalance>1000000) {  
            System.out.println(entry[i]);  
        }  
    }  
}  
public double calculateTotalSavingsBalance(String x) {  
    double totalSavings = 0;  
    for (int i = 0; i < listSize(); i++) {  
        if (x.equalsIgnoreCase(entry[i].accountType)) {  
            double totalBalance = entry[i].accountBalance +  
interestAmount(entry[i].accountBalance);  
            totalSavings += totalBalance;  
            System.out.println(entry[i]);  
        }  
    }  
    return totalSavings;  
}  
public void calculateSavingspercentage(){  
    double totalsavings=calculateTotalSavingsBalance("savings");  
    double totalBalance=0;
```

```
        for (int i = 0; i < listSize(); i++) {
            totalBalance+=entry[i].accountBalance+interestAmount(entry[i].account
Balance);
        }
        System.out.println("\nTotal savings Balance: "+totalsavings);
        System.out.println("Total Balance: "+totalBalance);
        System.out.println("\nPercentage of Savings in Total Balance:
"+(totalsavings/totalBalance)*100+" %");
    }
}
```

```
public class Customer {
    public String accountNumber;
    public String customerName;
    public String accountType;
    public double accountBalance;

    public Customer(String accountNumber, String customerName, String
accountType, double accountBalance) {
        this.accountNumber = accountNumber;
        this.customerName = customerName;
        this.accountType = accountType;
        this.accountBalance = accountBalance;
    }
    public String toString(){
        return(accountNumber+"\t\t"+customerName+"\t"+accountType+"\t\t"+accountB
alance);
    }
}
```

```
public class MainPromt {

    public static void main(String[] args) {
        Customer c1=new Customer("1001", "Kamal Dissanayake", "savings", 500000);
        Customer c2=new Customer("1002", "Namal Perera\t", "current", 975000);
        Customer c3=new Customer("1003", "Sithumm Udovita\t", "current", 100000);
        Customer c4=new Customer("1004", "Manel Dias\t", "savings", 1250000);
        Customer c5=new Customer("1005", "Chethiya Munasinghe", "savings",
950000);
        Customer c6=new Customer("1006", "Sanju Perera\t", "current", 1500000);
        Customer c7=new Customer("1007", "Lahiru Karunarathna", "savings",
600000);
        Customer c8=new Customer("1008", "Tharanga Prasad\t", "savings", 400000);
        Customer c9=new Customer("1009", "Shashi Dayarathna", "savings", 250000);
        Customer c10=new Customer("1010", "Anju Senanayake\t", "current",
1100000);

        List list=new List(10);
        list.insertLast(c1);
        list.insertLast(c2);
        list.insertLast(c3);
        list.insertLast(c4);
        list.insertLast(c5);
        list.insertLast(c6);
        list.insertLast(c7);
        list.insertLast(c8);
        list.insertLast(c9);
        list.insertLast(c10);
        System.out.println("\n-----Part(a)-----");
        -----");
        System.out.println("\nAccount Number\t"+"Customer Name\t\t"+"Account
Type\t"+"Account Balance(as of 2024.01.01)");
        list.traverseList();

        System.out.println("\n-----Part(b)-----");
        -----");
        System.out.println("\nAccount Number\t"+"Customer Name\t\t"+"Interest
Amount");
        list.printinterestAmount();

        System.out.println("\n-----Part(c)-----");
        -----");
        System.out.println("\nAccount Number\t"+"Customer Name\t\t"+"Account
Balance");
        list.totalAccountBalance();
    }
}
```

```
        System.out.println("\n-----Part(d)-----\n-----");
        list.quickSort(0, 9);
        System.out.println("\nAccount Number\t"+"Customer Name\t\t"+"Account
Type\t"+"Account Balance(as of 2024.01.01)");
        list.traverseList();

        System.out.println("\n-----Part(e)-----\n-----");
        System.out.println("\nCustomers with more than 1,000,000 after one year:
");
        list.displaycustomerWithHightBalance();

        System.out.println("\n-----Part(f)-----\n-----");
        System.out.println("\nCustomers with Savings Accounts: ");
        System.out.println("");
        System.out.println("Total Savings Account Balance:
"+list.calculateTotalSavingsBalance("savings"));

        System.out.println("\n-----Part(g)-----\n-----");
        System.out.println("\nPercentage of Savings in Total Balance: ");
        list.calculateSavingspercentage();
        System.out.println("\n-----\n-----");

    }
}
```

Output:-

```
PS D:\MY University Doc UOK\2nd Year 1st Semester\COSC 21063 Data Structure & Algorithms\Peactical Tu
Tutorial\Labsheet-09'; & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-agentlib:jdwp=transport=dt_soc
ng\Code\User\workspaceStorage\5639ac53777367358a08076e8456907\redhat.java\jdt_ws\Labsheet-09_9dd16f1
```

-----Part(a)-----

Account Number	Customer Name	Account Type	Account Balance(as of 2024.01.01)
1001	Kamal Dissanayake	savings	500000.0
1002	Namal Perera	current	975000.0
1003	Sithumm Udovita	current	100000.0
1004	Manel Dias	savings	1250000.0
1005	Chethiya Munasinghe	savings	950000.0
1006	Sanju Perera	current	1500000.0
1007	Lahiru Karunarathna	savings	600000.0
1008	Tharanga Prasad	savings	400000.0
1009	Shashi Dayarathna	savings	250000.0
1010	Anju Senanayake	current	1100000.0

-----Part(b)-----

Account Number	Customer Name	Interest Amount
1001	Kamal Dissanayake	40000.0
1002	Namal Perera	78000.0
1003	Sithumm Udovita	8000.0
1004	Manel Dias	100000.0
1005	Chethiya Munasinghe	76000.0
1006	Sanju Perera	120000.0
1007	Lahiru Karunarathna	48000.0
1008	Tharanga Prasad	32000.0
1009	Shashi Dayarathna	20000.0
1010	Anju Senanayake	88000.0

-----Part(c)-----

Account Number	Customer Name	Account Balance
1001	Kamal Dissanayake	540000.0
1002	Namal Perera	1053000.0
1003	Sithumm Udovita	108000.0
1004	Manel Dias	1350000.0
1005	Chethiya Munasinghe	1026000.0
1006	Sanju Perera	1620000.0
1007	Lahiru Karunarathna	648000.0
1008	Tharanga Prasad	432000.0
1009	Shashi Dayarathna	270000.0
1010	Anju Senanayake	1188000.0

-----Part(d)-----

Account Number	Customer Name	Account Type	Account Balance(as of 2024.01.01)
1003	Sithumm Udovita	current	100000.0
1009	Shashi Dayarathna	savings	250000.0
1008	Tharanga Prasad	savings	400000.0
1001	Kamal Dissanayake	savings	500000.0
1007	Lahiru Karunarathna	savings	600000.0
1005	Chethiya Munasinghe	savings	950000.0
1002	Namal Perera	current	975000.0
1010	Anju Senanayake	current	1100000.0
1004	Manel Dias	savings	1250000.0
1006	Sanju Perera	current	1500000.0

-----Part(e)-----

Customers with more than 1,000,000 after one year:

1010	Anju Senanayake	current	1100000.0
1004	Manel Dias	savings	1250000.0
1006	Sanju Perera	current	1500000.0

-----Part(f)-----

Customers with Savings Accounts:

1009	Shashi Dayarathna	savings	250000.0
1008	Tharanga Prasad	savings	400000.0
1001	Kamal Dissanayake	savings	500000.0
1007	Lahiru Karunarathna	savings	600000.0
1005	Chethiya Munasinghe	savings	950000.0
1004	Manel Dias	savings	1250000.0

Total Savings Account Balance: 4048750.0

-----Part(g)-----

Percentage of Savings in Total Balance:

1009	Shashi Dayarathna	savings	250000.0
1008	Tharanga Prasad	savings	400000.0
1001	Kamal Dissanayake	savings	500000.0
1007	Lahiru Karunarathna	savings	600000.0
1005	Chethiya Munasinghe	savings	950000.0
1004	Manel Dias	savings	1250000.0

Total savings Balance: 4048750.0

Total Balance: 7815625.0

Percentage of Savings in Total Balance: 51.80327868852459 %

PS D:\MY University Doc UOK\2nd Year 1st Semester\COSC 21063 Data Structure & Algorithms\Peactical T