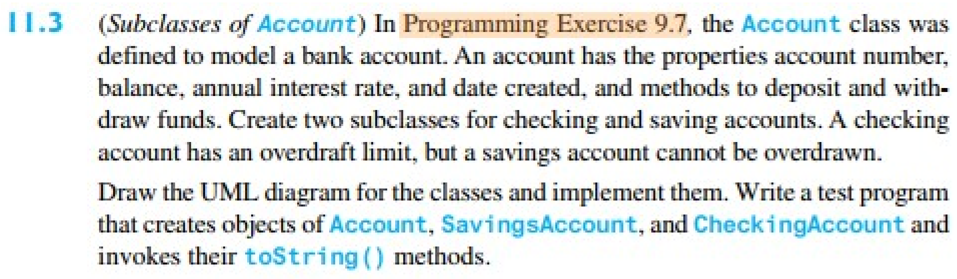


**11.3** (*Subclasses of* `Account`) In Programming Exercise 9.7, the `Account` class was defined to model a bank account. An account has the properties account number, balance, annual interest rate, and date created, and methods to deposit and withdraw funds. Create two subclasses for checking and saving accounts. A checking account has an overdraft limit, but a savings account cannot be overdrawn.

Draw the UML diagram for the classes and implement them. Write a test program that creates objects of `Account`, `SavingsAccount`, and `CheckingAccount` and invokes their `toString()` methods.

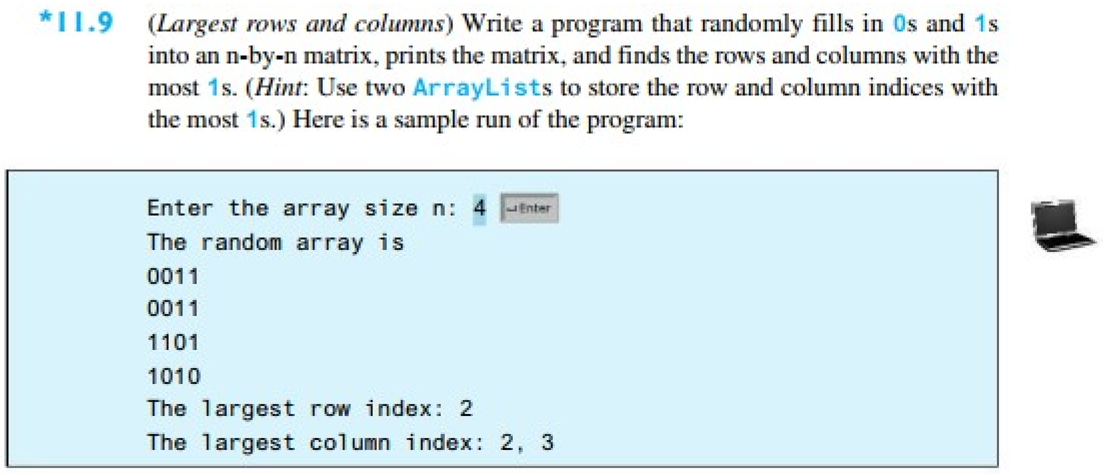
| checkingAccount |
|---|
| - overDraftLimit: double<br>+ CheckingAccount(id: int, bal: double)<br>+withdraw (w: double)<br>+ deposit (d: deposit) |
| + toString |

| SavingsAccount |
|---|
| -overDraftLimit: double<br>+ CheckingAccount(id: int, bal: double)<br>+withdraw (w: double)<br>+ deposit (d: deposit) |
| + toString(): String |

```
public static void main(String[] args) throws Exception {
      Account a= new Account(1122,20000.00);
      System.out.println(a.toString());

      chequingAccount chAccount= new chequingAccount(1122,20000.00);
      System.out.println( chAccount.toString() );

      savingsAccount svAccount= new savingsAccount(1122,20000.00);
      System.out.println( svAccount.toString());
}
```



***11.9** (*Largest rows and columns*) Write a program that randomly fills in 0s and 1s into an n-by-n matrix, prints the matrix, and finds the rows and columns with the most 1s. (*Hint*: Use two ArrayLists to store the row and column indices with the most 1s.) Here is a sample run of the program:

```
Enter the array size n: 4
The random array is
0011
0011
1101
1010
The largest row index: 2
The largest column index: 2, 3
```

```
public class ArrayCounter  {

    private int[][] array;
    public StackofIntegers RowCountStack;
    public StackofIntegers ColCountStack;
    private int CurrentMaxCount;
    private int CurrentMaxCount2;

    public static void main(String[] args) throws Exception {
        Scanner input = new Scanner(System.in);
        System.out.println("Please enter the size of the array");
        int size= input.nextInt();
        System.out.println("The Random array is");

        ArrayCounter x = new ArrayCounter(size);
        x.printArray();
        x.maxRow();
        x.maxCol();
        System.out.print("\nThe largest row index: " );
        do{
            System.out.print( x.RowCountStack.pop()+" ");

        }while(!x.RowCountStack.empty() );
        System.out.println();

        System.out.print("The largest column index: ");

        do{
            System.out.print( x.ColCountStack.pop()+" ");

        }while(!x.ColCountStack.empty() );
        System.out.println();
        input.close();

    }

    public ArrayCounter(int sz) {
        this.array= new int[sz][sz];
        this.RowCountStack= new StackofIntegers();
        this.ColCountStack= new StackofIntegers();
        this.fillArray();
    }

    public void fillArray() {
        int temp;
        for (int i = 0; i < array.length; i++) {
            for (int j = 0; j < array[i].length; j++) {
                temp = (int)(Math.random() * 2);
            array[i][j] = temp;
        }
    }
}

public void printArray() {
    for (int i = 0; i < array.length; i++) {
        System.out.println();
        for (int j = 0; j < array.length; j++) {
            System.out.print(array[i][j] + " ");
        }
    }
}

public void maxRow() {
    int row=  -1;  int tempMax=0;
    int count=0; //int rowCount= 0;
    for (int i = 0; i < array.length; i++) {
        row = i; count = 0; tempMax=0;
        for (int j = 0; j < array.length; j++) {
            if(array[i][j]==1){
                count++;
            }
            if(count>tempMax){
                tempMax= count;
            }
        }
        addRowEntry(RowCountStack, row, tempMax);
    }
}
```

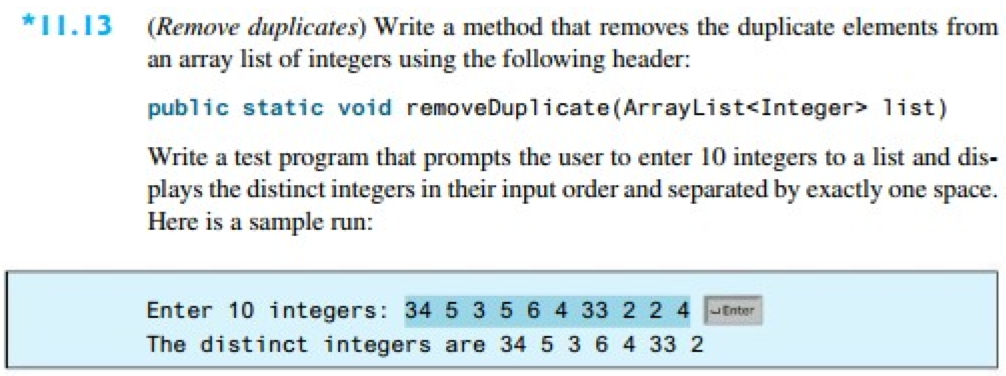
```
public void addRowEntry(StackofIntegers stack, int row, int tempMax)
{
    if(stack.empty()){
        stack.push(row+1);
        this.CurrentMaxCount= tempMax;
    }
    else{
        if( tempMax > this.CurrentMaxCount  ){
            do { stack.pop();}
                while(!stack.empty());
            stack.push(row+1);
            this.CurrentMaxCount= tempMax;
        }
        else{
            if( tempMax >= this.CurrentMaxCount ){
                stack.push(row+1);
            }
        }
    }
}

public void addColEntry(StackofIntegers stack, int col, int tempMax)
{
    if(stack.empty()){
        stack.push(col+1);
        this.CurrentMaxCount2= tempMax;
    }
    else{
        if( tempMax > this.CurrentMaxCount2  ){
}

            do { stack.pop();}
                while(!stack.empty());
            stack.push(col+1);
            this.CurrentMaxCount2= tempMax;
        }
        else{
            if( tempMax >= this.CurrentMaxCount2 ){
                stack.push(col+1);
            }
        }
    }
}

 public void maxCol() {
    int col=  -1; int tempMax=0;
    int count=0; int colCount= 0;
    for (int i = 0; i < array.length; i++) {
        col = i; count =0;
        for (int j = 0; j < array.length; j++) {
            if(array[j][i]==1){
                count++;
            }
            if(count>colCount){
                tempMax= count;
            }
        }
        addColEntry(ColCountStack, col, tempMax);
    }
}
```

}



***11.13** (*Remove duplicates*) Write a method that removes the duplicate elements from an array list of integers using the following header:

```
public static void removeDuplicate(ArrayList<Integer> list)
```

Write a test program that prompts the user to enter 10 integers to a list and displays the distinct integers in their input order and separated by exactly one space. Here is a sample run:

```
Enter 10 integers: 34 5 3 5 6 4 33 2 2 4 ↵Enter
The distinct integers are 34 5 3 6 4 33 2
```

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
//import java.lang.StringBuilder;

public class StringIntegers {
    //public  static  int[] rawInput;
    public static ArrayList<Integer> rawInput;
    public static void main(String[] args) throws Exception {
        System.out.println("Please Enter 10 inetgers");

        Scanner input = new Scanner(System.in);
        //rawInput = new int[10];
        //strElements = new StringBuilder();
        rawInput = new ArrayList <Integer>();
        captureInputs(input);
        removeDuplicate(rawInput);
        printList(rawInput);

        input.close();
    }

    public static void captureInputs(Scanner in){
        int i = 0;
        while( i < 10){
            rawInput.add(  in.nextInt() );
            i++;
        }
    }

    public static void removeDuplicate ( ArrayList<Integer> list){
        for (int i = 0; i < list.size(); i++) {
            for (int j = i+1; j < list.size(); j++) {
                if( list.get(i) == list.get(j)){
                    list.remove(j);
                }
            }
        }
    }

    public static void printList(ArrayList<Integer> list){
        System.out.print("\nThe distinct integers are \n");
        for (int i = 0; i < list.size(); i++) {
            System.out.print(list.get(i)+" ");
        }
        System.out.println();
    }

}
```