CS 162FZ: Introduction to Computer Science II

Lecture 07

Revision for Lab1-3

Dr. Chun-Yang Zhang

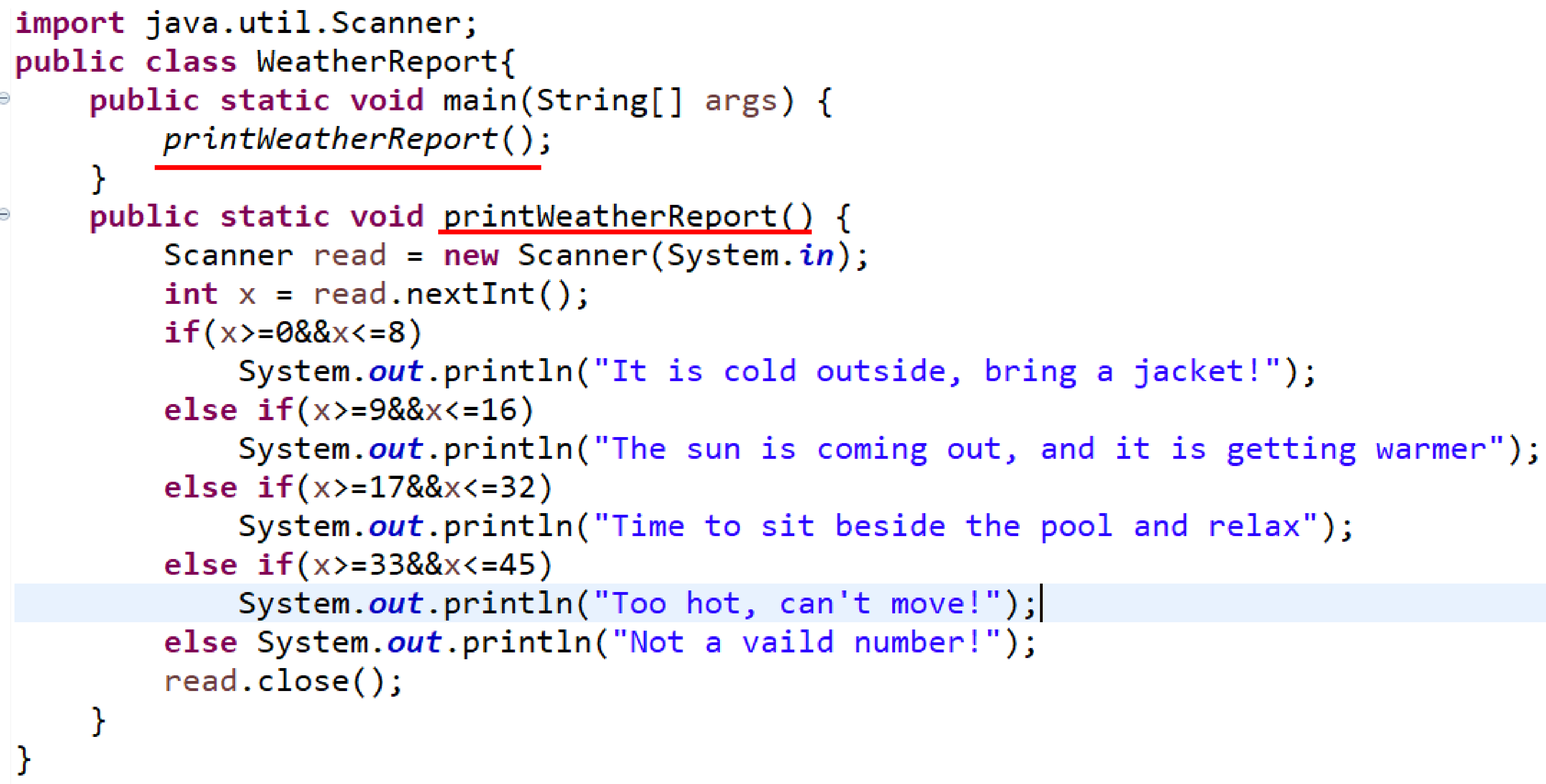
Lab1, Task1: WeatherReport

- Write a Java program, called WeatherReport, which contains a static method called printWeatherReport() which takes no input parameters and returns nothing. The printWeatherReport() method should get the user to input a single integer value between 0 and 45 degrees Celsius and print a message about today's weather (see table below for ranges and required messages). To test the printWeatherReport() method you should call it from the main method.
- Note: The message in your code should match EXACTLY the message in the table to pass all test cases.

Range	Message
0 <= x <= 8	It is cold outside, bring a jacket!
9 <= x <= 16	The sun is coming out, and it is getting warmer
17 <= x <= 32	Time to sit beside the pool and relax
33 <= x <= 45	Too hot, can't move!
x < 0 & x > 45	Not a valid number!

Solution

Static method, no input and returning value



```
import java.util.Scanner;
public class WeatherReport{
    public static void main(String[] args) {
        printWeatherReport();
    }
    public static void printWeatherReport() {
        Scanner read = new Scanner(System.in);
        int x = read.nextInt();
        if(x>=0&&x<=8)
            System.out.println("It is cold outside, bring a jacket!");
        else if(x>=9&&x<=16)
            System.out.println("The sun is coming out, and it is getting warmer");
        else if(x>=17&&x<=32)
            System.out.println("Time to sit beside the pool and relax");
        else if(x>=33&&x<=45)
            System.out.println("Too hot, can't move!");
        else System.out.println("Not a vaild number!");
        read.close();
    }
}
```

Lab1, Task2: StarTriangle

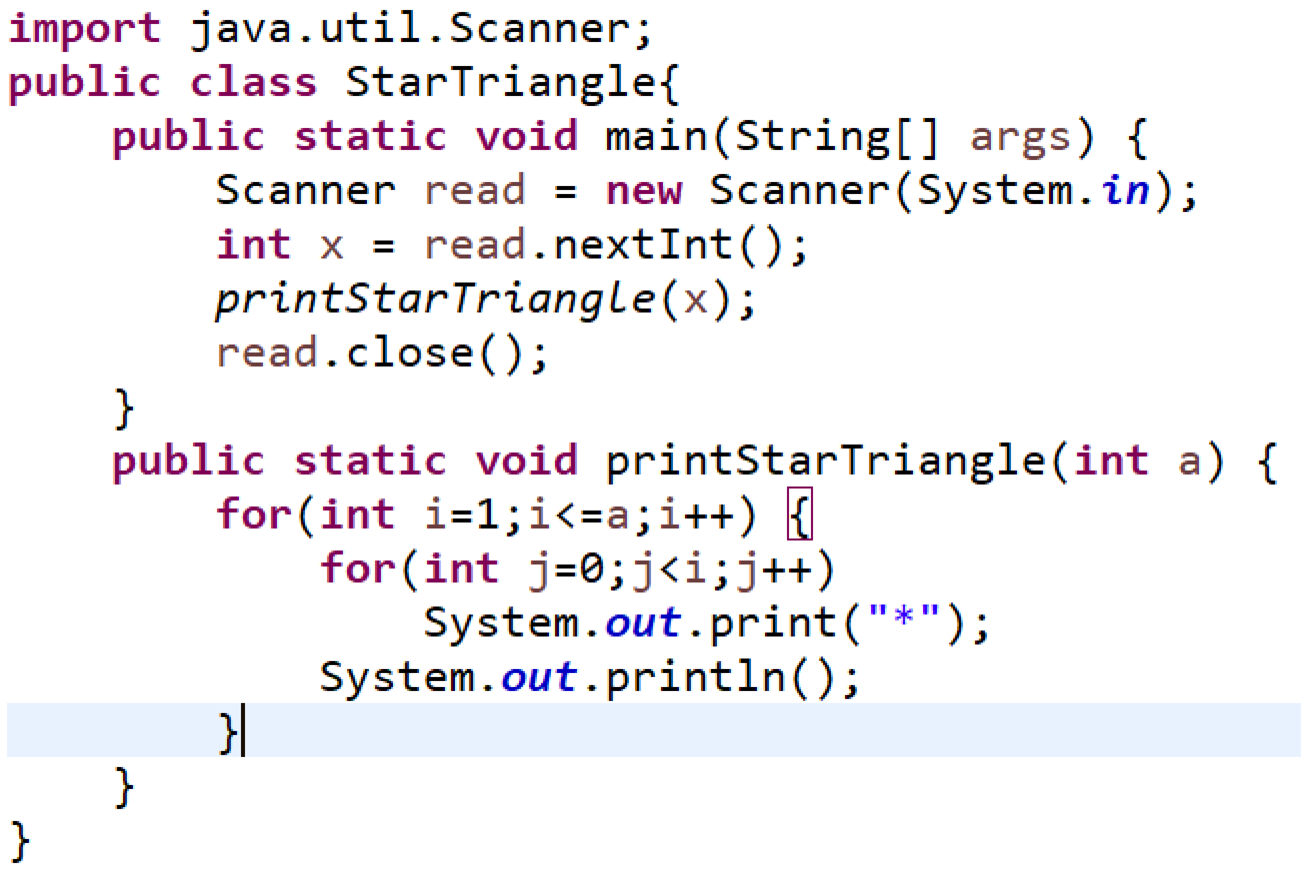
Write a method called printStarTriangle() that accepts an integer parameter and returns nothing. This method should print to the screen rows of asterisk characters that form the shape of a triangle. The amount of rows is based on the value of the parameter. The first row should start with one * and increment by one * for each row thereafter.

For example, if the parameter held the value 3 the method would print:

```
*
**
***
```

3 rows of asterisks, starting with one asterisk and increasing by one for every row after. In the main method you should call the method with a user inputted integer being passed into the method printStarTriangle().

Solution



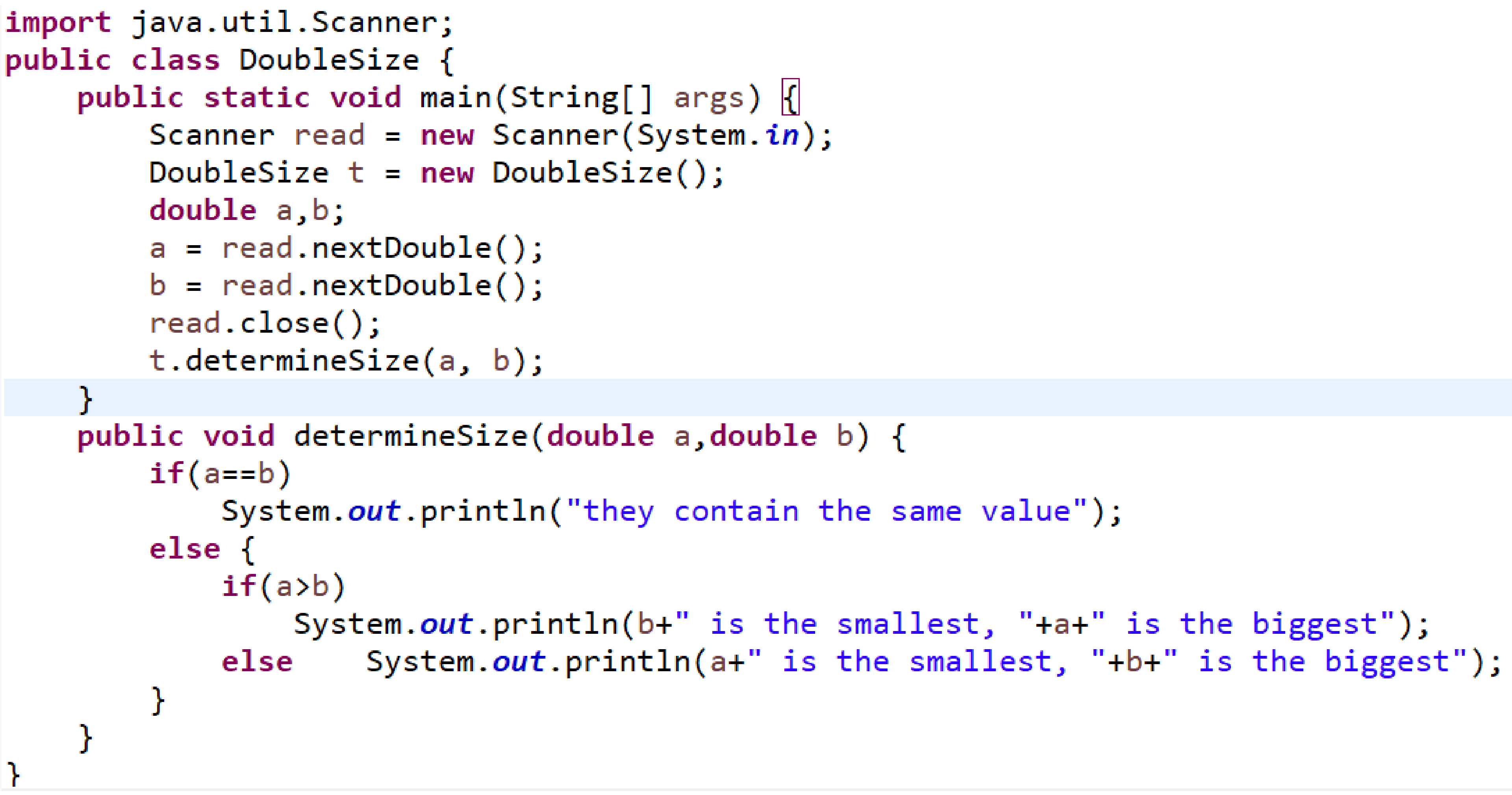
```java
import java.util.Scanner;
public class StarTriangle{
    public static void main(String[] args) {
        Scanner read = new Scanner(System.in);
        int x = read.nextInt();
        printStarTriangle(x);
        read.close();
    }
    public static void printStarTriangle(int a) {
        for(int i=1;i<=a;i++) {
            for(int j=0;j<i;j++)
                System.out.print("*");
            System.out.println();
        }
    }
}
```

Note: 1. actual parameters and formal parameters;
2. parameter list, passing by value
3. return nothing

Lab2, Task3: DoubleSize

Write a static method called determineSize() that accepts two doubles as parameters. The method should print a message to the screen that states which number is the smallest, which is the biggest or that they contain the same value. Test the method by calling it from the main method using user input.

Solution



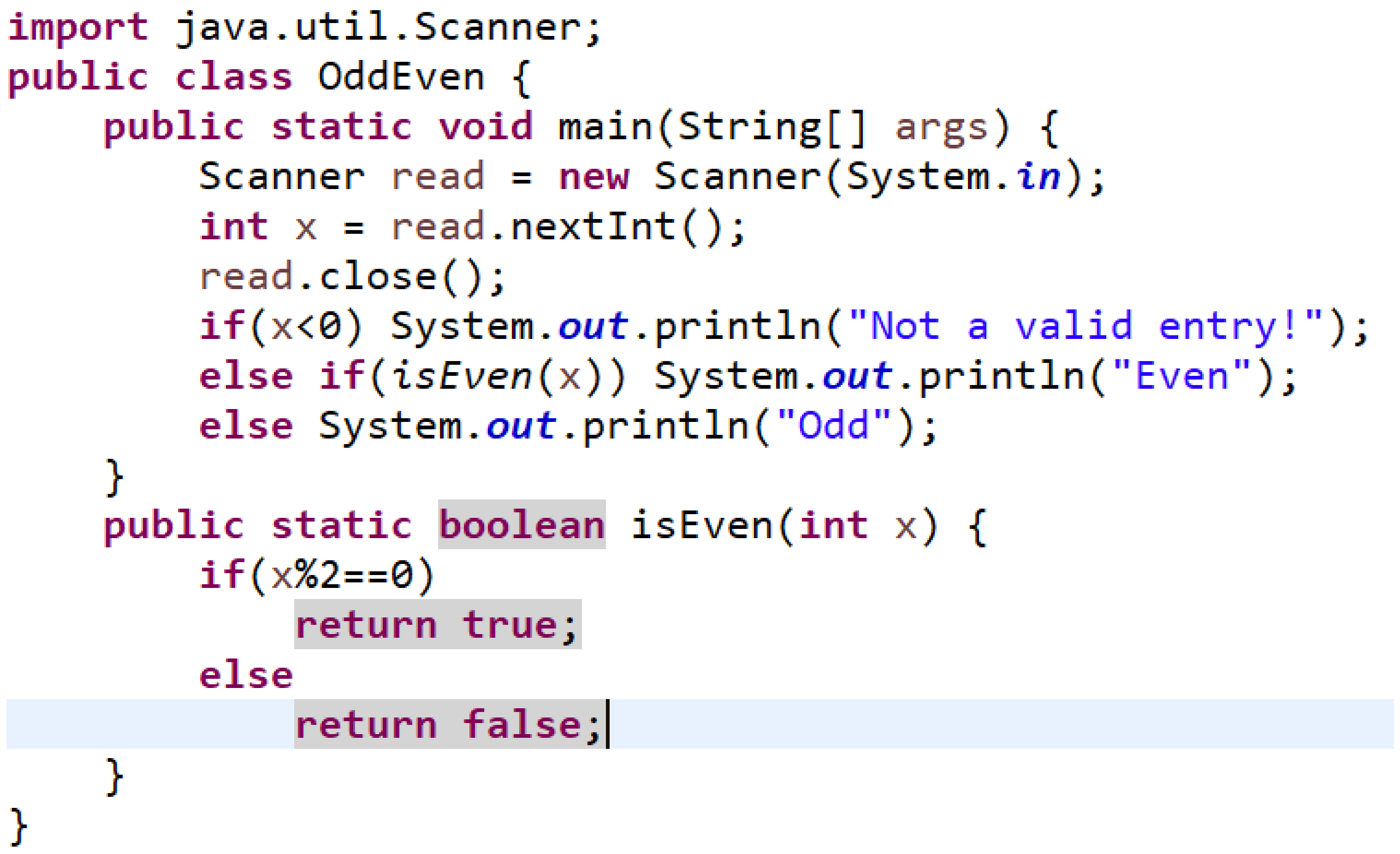
```java
import java.util.Scanner;
public class DoubleSize {
    public static void main(String[] args) {
        Scanner read = new Scanner(System.in);
        DoubleSize t = new DoubleSize();
        double a,b;
        a = read.nextDouble();
        b = read.nextDouble();
        read.close();
        t.determineSize(a, b);
    }
    public void determineSize(double a,double b) {
        if(a==b)
            System.out.println("they contain the same value");
        else {
            if(a>b)
                System.out.println(b+" is the smallest, "+a+" is the biggest");
            else    System.out.println(a+" is the smallest, "+b+" is the biggest");
        }
    }
}
```

Note: non-static method with two inputs

Lab1, Task 4: OddEven

Write a Java program, called OddEven that contains a static method called isEven() that accepts a single non-negative integer as its parameter and returns true if the number is even and false otherwise. The input parameter should be read in from the user in the main method and passed to isEven(). The main method should print a message ("Odd" or "Even") to the screen, depending on the return value. If the input is negative an error message should print to the screen stating "Not a valid entry!"

Solution



```java
import java.util.Scanner;
public class OddEven {
    public static void main(String[] args) {
        Scanner read = new Scanner(System.in);
        int x = read.nextInt();
        read.close();
        if(x<0) System.out.println("Not a valid entry!");
        else if(isEven(x)) System.out.println("Even");
        else System.out.println("Odd");
    }
    public static boolean isEven(int x) {
        if(x%2==0)
            return true;
        else
            return false;
    }
}
```

Note: the type of returning value is boolean

Lab1, Task 5: PrintDigits

Write a Java program using 2D arrays to print numbers in digital form. If the number is longer than 1 digit the program should print each digit to a new line. The input should be taken from the user. Your source file should be name as PrintDigits.

Sample Input:

5

Sample output:

```
 --
|
 --
    |
 --
```

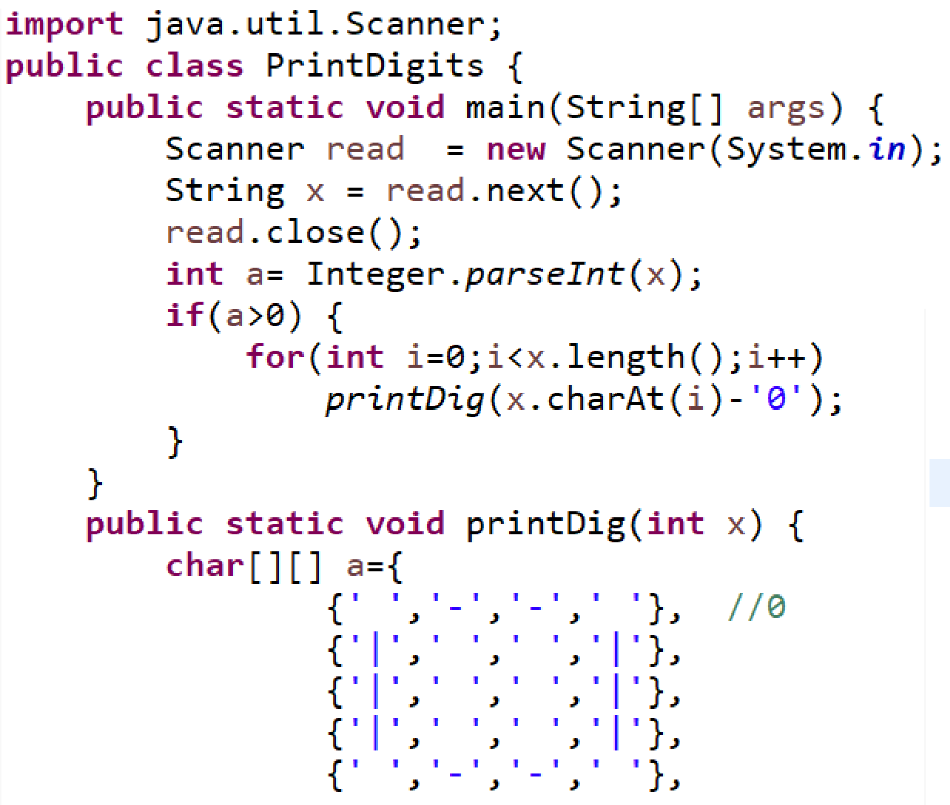
Sample Input:

32

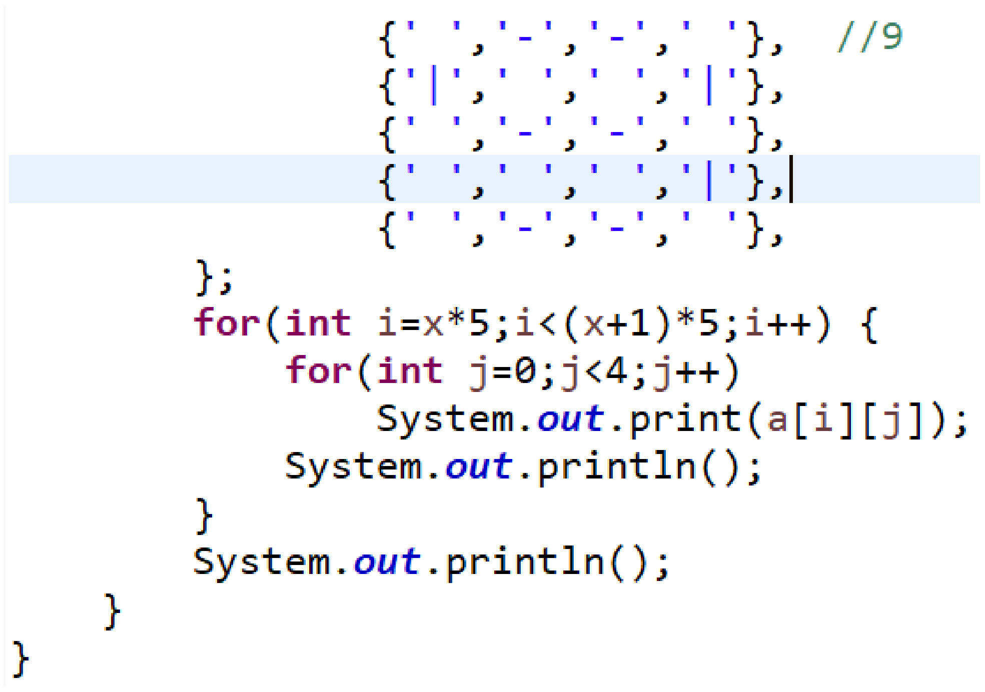
Sample output:

```
--
    |
--
    |
--
 --
    |
 --
|
 --
```

Solution



```
import java.util.Scanner;
public class PrintDigits {
    public static void main(String[] args) {
        Scanner read  = new Scanner(System.in);
        String x = read.next();
        read.close();
        int a= Integer.parseInt(x);
        if(a>0) {
            for(int i=0;i<x.length();i++)
                printDig(x.charAt(i)-'0');
        }
    }
    public static void printDig(int x) {
        char[][] a={
                {' ','-','-',' '},  //0
                {'|',' ',' ','|'},
                {'|',' ',' ','|'},
                {'|',' ',' ','|'},
                {' ','-','-',' '},
```



```
                {' ','-','-',' '},  //9
                {'|',' ',' ','|'},
                {' ','-','-',' '},
                {' ',' ',' ','|'},
                {' ','-','-',' '},
        };
        for(int i=x*5;i<(x+1)*5;i++) {
            for(int j=0;j<4;j++)
                System.out.print(a[i][j]);
            System.out.println();
        }
        System.out.println();
    }
}
```

Lab2, Task1: Factorial

Write a Java program, called Factorial that contains a static method called getFactorial(). The getFactorial() method should accept an integer value as n for the intial number as well as m as the last number, both as parameters. It will return nothing and print the factorial of the input parameter to the screen. The main method should read in user input in the form of two integer numbers, n and m. When the code executes the factorial of all numbers from n up to m should be printed to the screen each on a new line using a loop. If either of the user inputs are negative, an error message should print to the screen stating "Not a valid entry!"

Input

n m , where n and m are integer values

Sample input

```
2 5
```

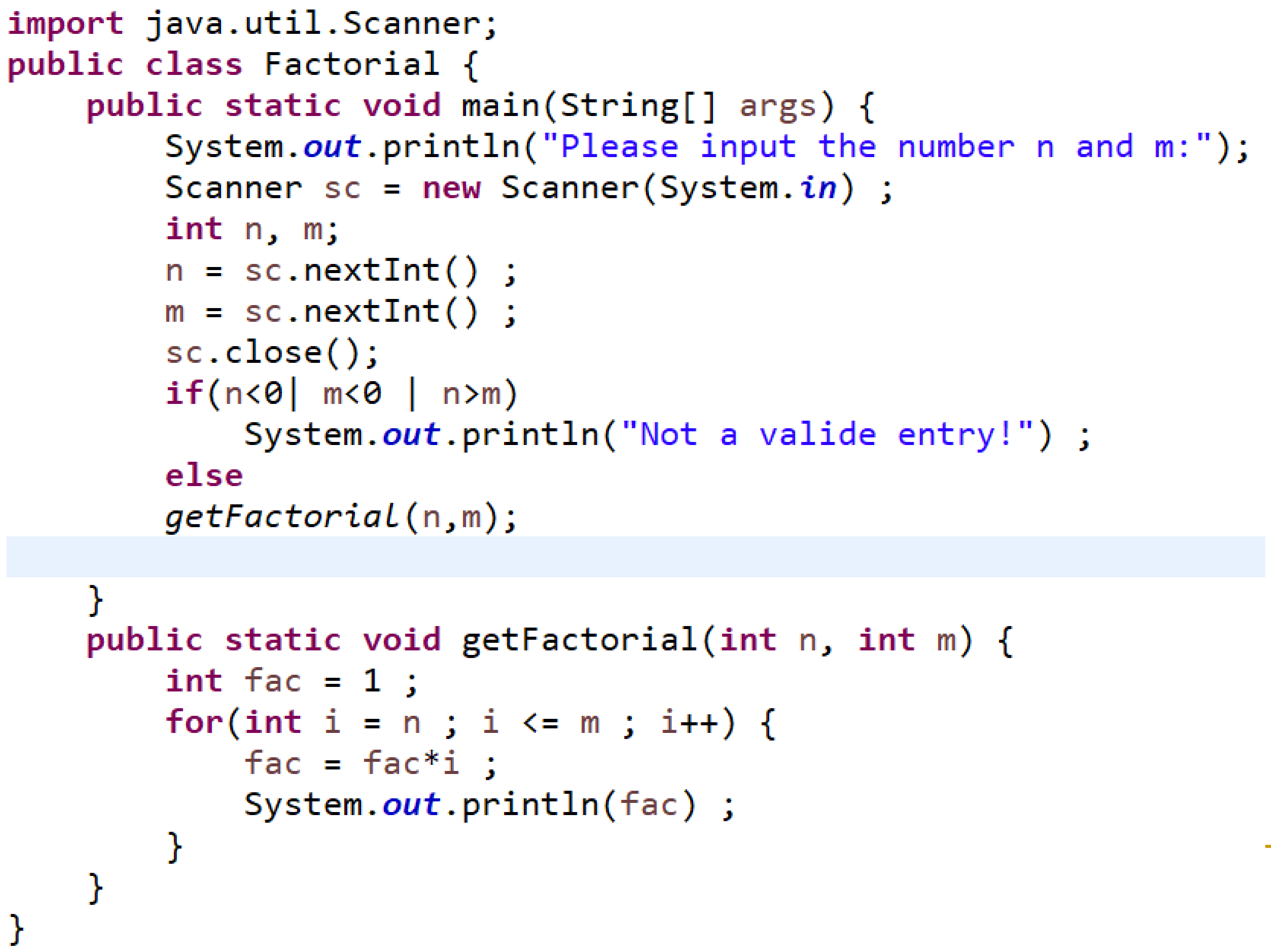
Output

The Factorial of all numbers from n up to m

Sample output

```
2
6
24
120
```

Solution



```java
import java.util.Scanner;
public class Factorial {
    public static void main(String[] args) {
        System.out.println("Please input the number n and m:");
        Scanner sc = new Scanner(System.in) ;
        int n, m;
        n = sc.nextInt() ;
        m = sc.nextInt() ;
        sc.close();
        if(n<0| m<0 | n>m)
            System.out.println("Not a valide entry!") ;
        else
        getFactorial(n,m);

    }
    public static void getFactorial(int n, int m) {
        int fac = 1 ;
        for(int i = n ; i <= m ; i++) {
            fac = fac*i ;
            System.out.println(fac) ;
        }
    }
}
```

Lab2, Task2: MethodsArrays

Write a Java program, called MethodsArrays that has 4 static methods called fillArray(), sumArray(), avgArray(), and printArray(). The fillArray() method should be called from the main method. The fillArray() method should use a Scanner to take in a number representing the length of the array and then read in numbers to fill the array. The sumArray() method should take an int array as its input parameter and returns an integer value that is the sum of all the elements in the array. The avgArray() method should take an int array as its input parameter and returns an double value that is the average of all the elements in the array.
The printArray() method should take an int array as its input parameter and has no return value. It should then print out the elements of the array on the same line separated by a space (“ “). All methods should work for integer arrays.

Solution

```
import java.util.Scanner;
public class MethodsArrays {
    public static void main(String[] args) {
        int[] arr = fillArray();
        int sum = sumArray(arr);
        System.out.println("Sum = " + sum);
        double avg = avgArray(arr);
        System.out.println("Average = " + avg);
        printArray(arr);
    }
    public static int[] fillArray() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Please input the length of the array:");
        int arrLength = sc.nextInt();
        int array[] = new int[arrLength];
        for(int i = 0 ; i < arrLength ; i++) {
            array[i] = sc.nextInt();
        }
        return array;

    }
```

Solution cont'd

```
        public static int sumArray(int arr[]) {
            int sum = 0 ;
            for(int i =0 ; i < arr.length ; i++) {
                sum += arr[i] ;
            }
            return sum;
        }
        public static double avgArray(int arr[]) {
            double sum = 0 ;
            for(int i =0 ; i < arr.length ; i++) {
                sum += arr[i] ;
            }
            return sum/arr.length;
        }
        public static void printArray(int arr[]) {
            for(int i =0 ; i < arr.length ; i++) {
                System.out.print(arr[i]+" ");
            }
        }
    }
```

Lab2, Task3: LinearSearch

Write a Java program, called LinearSearch that contains a static method called findElement() that accepts a double array and a double as its parameters and return a boolean value if the double number entered is contained in the double array. In the main method you should use user input to take in the length of the array to be filled, then the elements of the array, and finally the element you wish to search for in the array. You program should call findElement() to check for the occurence of the double number entered in the array. You should print an appropriate message to the screen if the element is found.

Sample input 1

```
4.0 345.6 678.954 234.534 -89.23 678.954
```

Sample input 2

```
4.0 345.6 678.954 234.534 -89.23 67.4
```

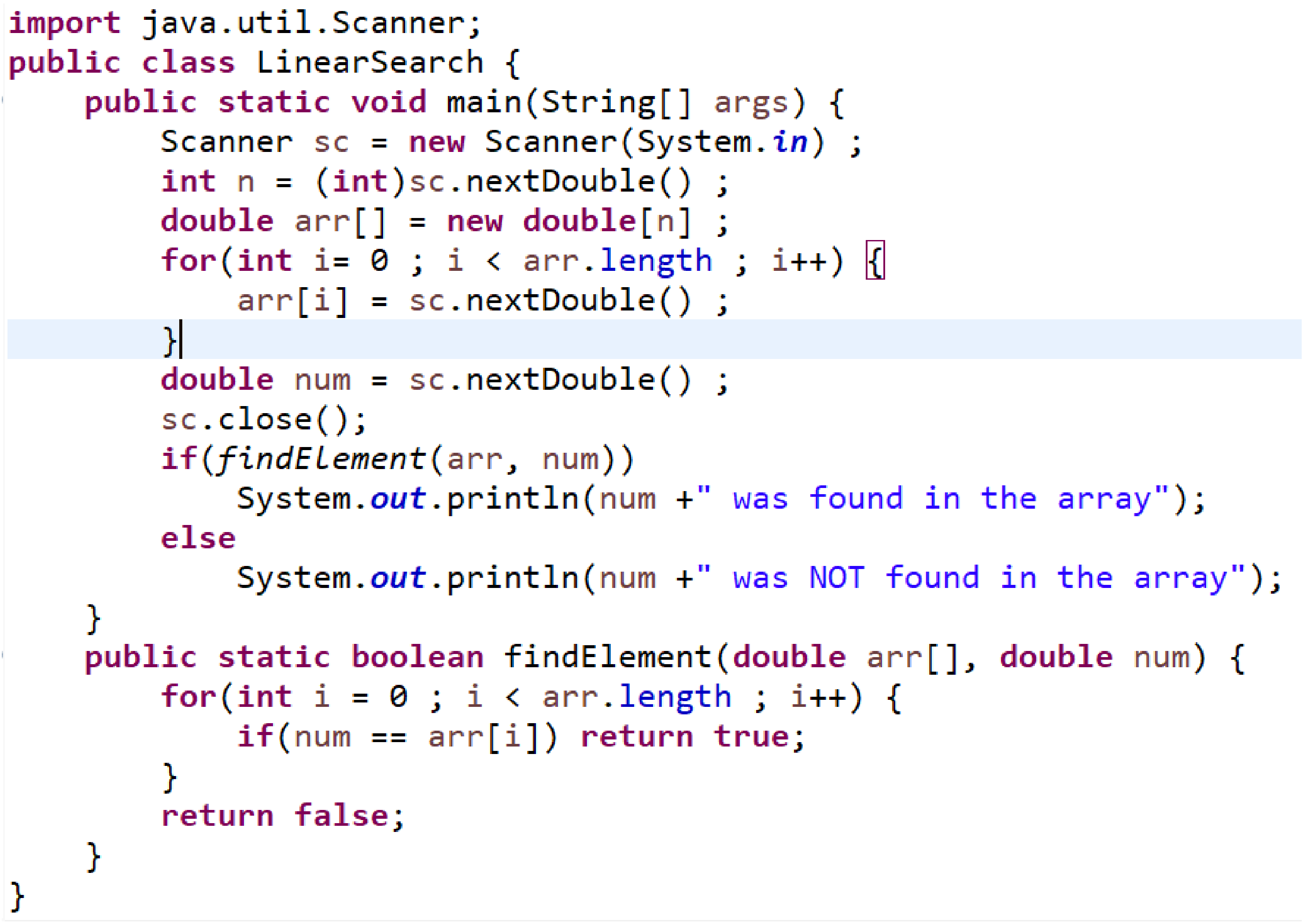
Sample output 1

```
678.954 was found in the array
```

Sample output 2

```
67.4 was NOT found in the array
```

Solution



```java
import java.util.Scanner;
public class LinearSearch {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in) ;
        int n = (int)sc.nextDouble() ;
        double arr[] = new double[n] ;
        for(int i= 0 ; i < arr.length ; i++) {
            arr[i] = sc.nextDouble() ;
        }
        double num = sc.nextDouble() ;
        sc.close();
        if(findElement(arr, num))
            System.out.println(num +" was found in the array");
        else
            System.out.println(num +" was NOT found in the array");
    }
    public static boolean findElement(double arr[], double num) {
        for(int i = 0 ; i < arr.length ; i++) {
            if(num == arr[i]) return true;
        }
        return false;
    }
}
```

Lab2, Task4

See the code template given in Programming2.java your task is to write a text based version of 2 player game of Tic-tac-toe (https://www.exploratorium.edu/brain_explorer/tictactoe.html). You should use your knowledge of methods to complete this task.

Your game should indicate which player has the current move and print the current state of the board after the move and check if the player has a winning combination.

You must show your demonstrator your algorithm and fully commented code for this task.

Solution

- Demonstrate directly.

Lab3, Task1:String Acceptance

Write a method that takes a String as a parameter and returns true if the string contains 0's and 1's only and will return false otherwise. You must use the .matches(<...>) method with a regular expression. In your main method, call the method with user input and print to the screen the answer returned.

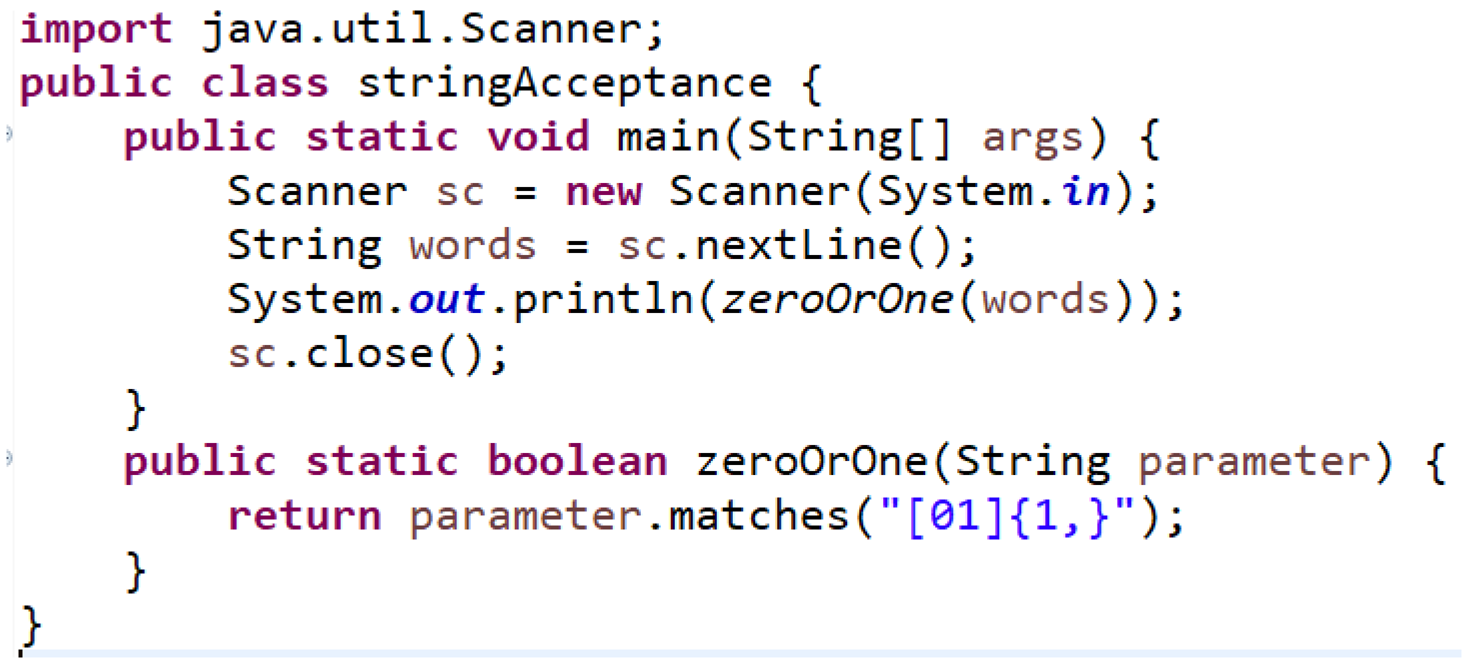
Sample Input: 010110

Sample Output:true

Sample Input: 0210

Sample Output: false

Solution



```java
import java.util.Scanner;
public class stringAcceptance {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String words = sc.nextLine();
        System.out.println(zeroOrOne(words));
        sc.close();
    }
    public static boolean zeroOrOne(String parameter) {
        return parameter.matches("[01]{1,}");
    }
}
```

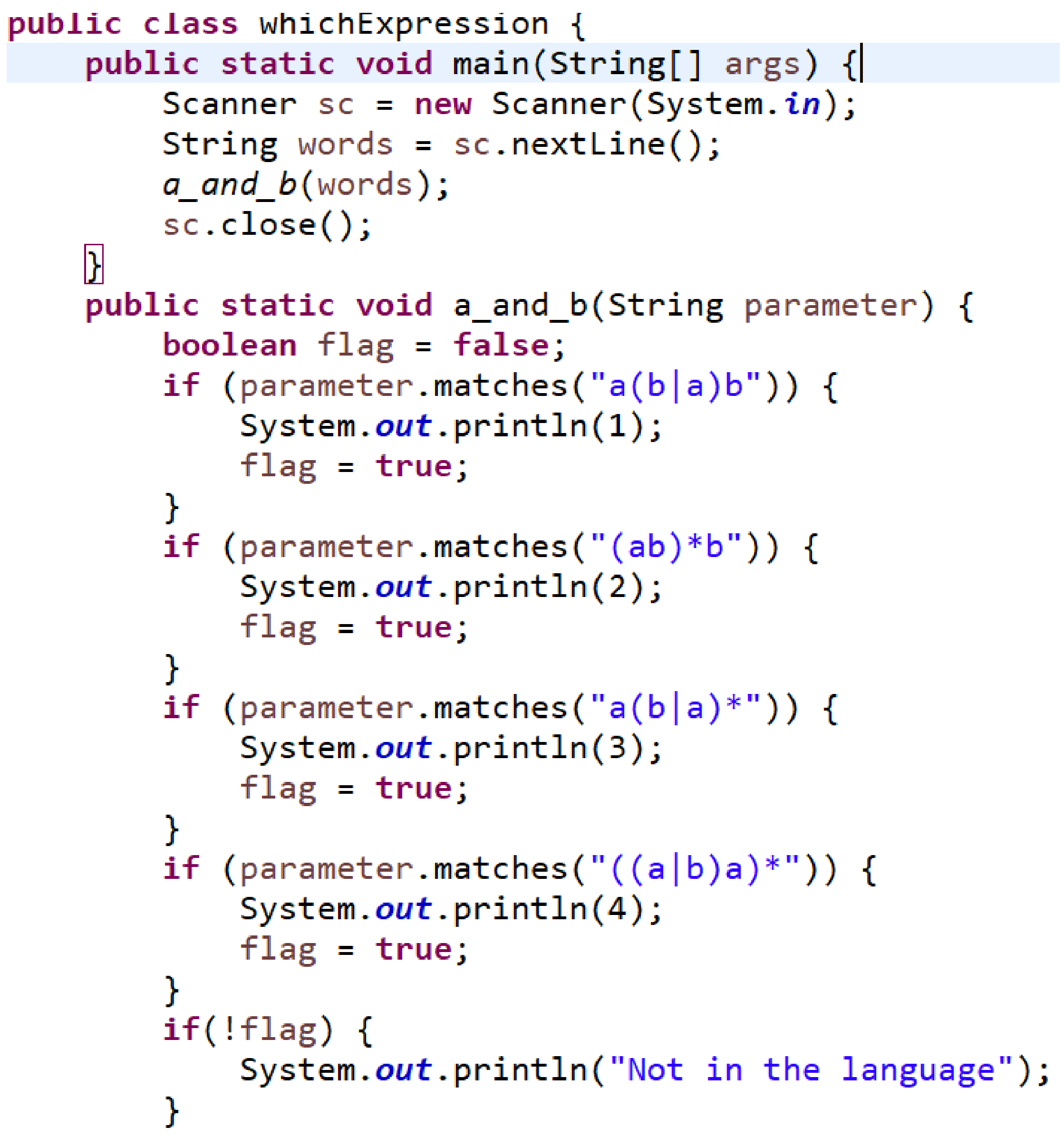
Lab3, Task2: Which Expression

Write a method that takes a String as a parameter and prints to the screen which of the regular expressions shown below is satisfied by the String. The String can only contain the characters a and b and you must use the.matches(<...>) method with a regular expression. The method doesn't have to return anything.

Regular Expression No.	**Regular Expression**
1:	**a(b|a)b**
2:	**(ab)* b**
3:	**a(b|a)***
4:	**((a|b) a)***

If the String satisfies more than one regular expression print each one on a new line. If the String does not satisfy any expression have a valid print statement stating this fact. In your main method call the method with user input.

Solution

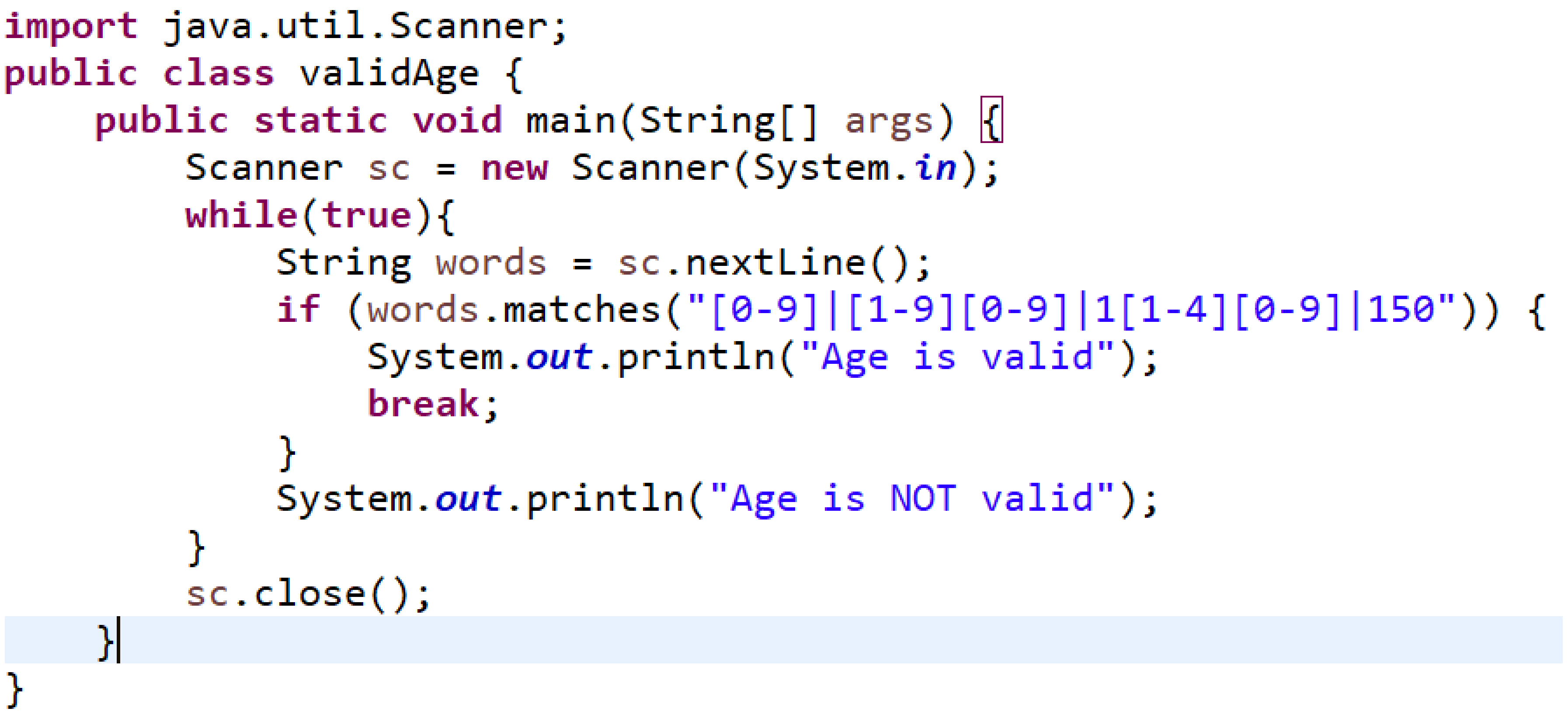


```java
public class whichExpression {
    public static main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String words = sc.nextLine();
        a_and_b(words);
        sc.close();
    }
    public static void a_and_b(String parameter) {
        boolean flag = false;
        if (parameter.matches("a(b|a)b")) {
            System.out.println(1);
            flag = true;
        }
        if (parameter.matches("(ab)*b")) {
            System.out.println(2);
            flag = true;
        }
        if (parameter.matches("a(b|a)*")) {
            System.out.println(3);
            flag = true;
        }
        if (parameter.matches("((a|b)a)*")) {
            System.out.println(4);
            flag = true;
        }
        if(!flag) {
            System.out.println("Not in the language");
        }
```

Lab3, Task3: Valid Age

Write a Java program that requests a user to input their age. The program should verify that the value the user enters is considered valid, lying in the range between 0 and 150. This program should use the.matches(<...>) method with a regular expression. If the user does not enter a valid age, your program should continually request that they enter a valid age until a valid age is entered.

Solution



```java
import java.util.Scanner;
public class validAge {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while(true){
            String words = sc.nextLine();
            if (words.matches("[0-9]|[1-9][0-9]|1[1-4][0-9]|150")) {
                System.out.println("Age is valid");
                break;
            }
            System.out.println("Age is NOT valid");
        }
        sc.close();
    }
}
```

Lab3, Task 4: Valid Name

Write a Java program that requests a user to input their first name. The program should verify that the user enters a reasonable first name. A reasonable first name should begin with a capital letter. It should be between 2 and 25 characters in length. It shouldn't contain any special characters. This program should use the .matches(<...>) method with a regular expression. If the user does not enter a valid name, your program should continually request that they enter a valid name until a valid name is entered.

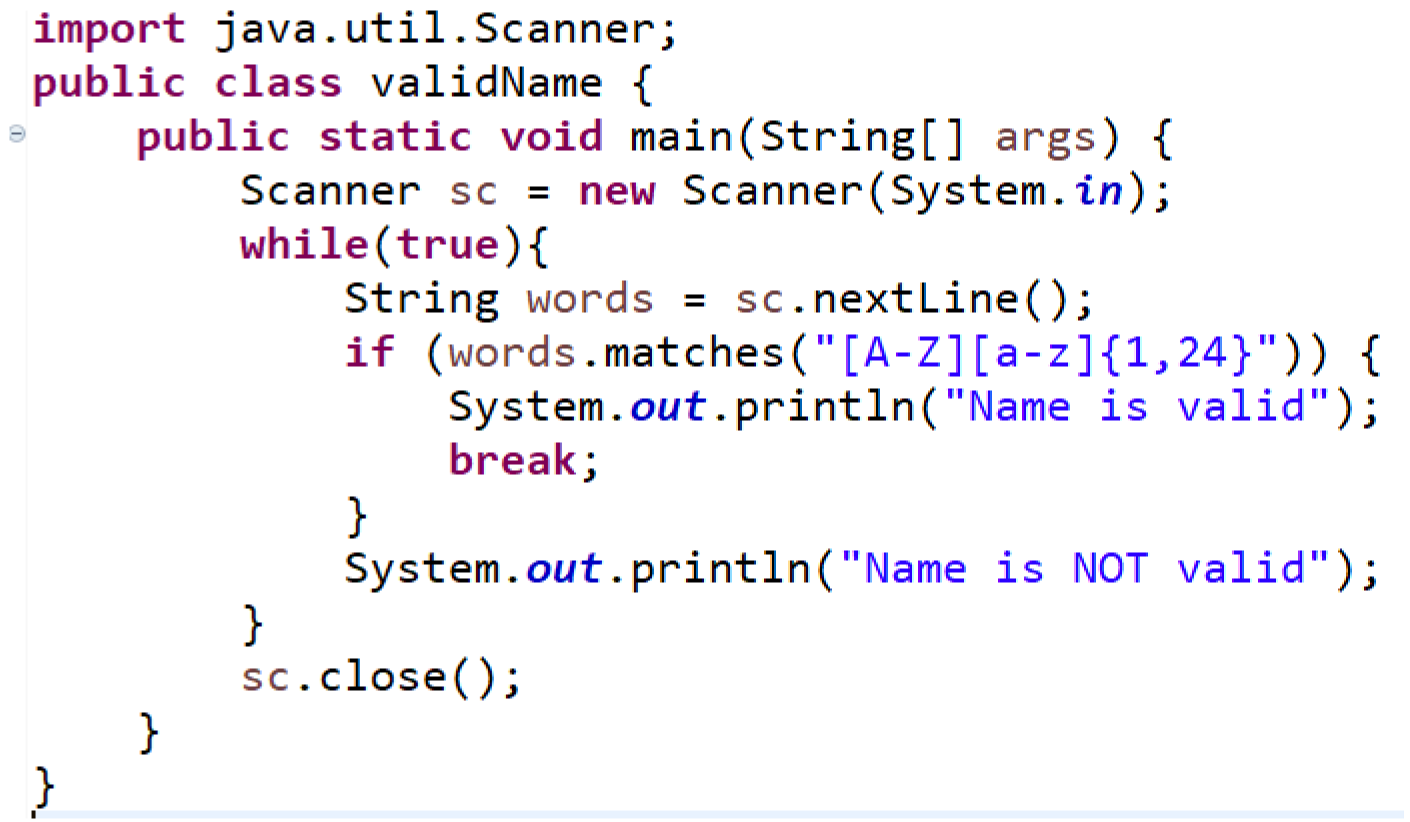
1) Sample run of program:

```
Michael            //user input
Name is valid        //output based on value of name
```

2) Sample run of program:

```
aideen             //user input
Name is NOT valid   //output based on value of name
Aid33n //user input
Name is NOT valid    //output based on value of name
Aideen             //user input
Name is valid        //output based on value of name
```

Solution



```java
import java.util.Scanner;
public class validName {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while(true){
            String words = sc.nextLine();
            if (words.matches("[A-Z][a-z]{1,24}")) {
                System.out.println("Name is valid");
                break;
            }
            System.out.println("Name is NOT valid");
        }
        sc.close();
    }
}
```

Lab3, Task 5: Flight Codes

Write a Java program that requests a user to enter a flight code. The program should verify that the user enters a valid flight code. A valid flight code begins with 2 or 3 capital letters and is followed by 3 or 4 digits. This program should use the .matches(<...>) method with a regular expression. If the user does not enter a valid code, your program should continually request that they enter a valid flight code until a valid code is entered.

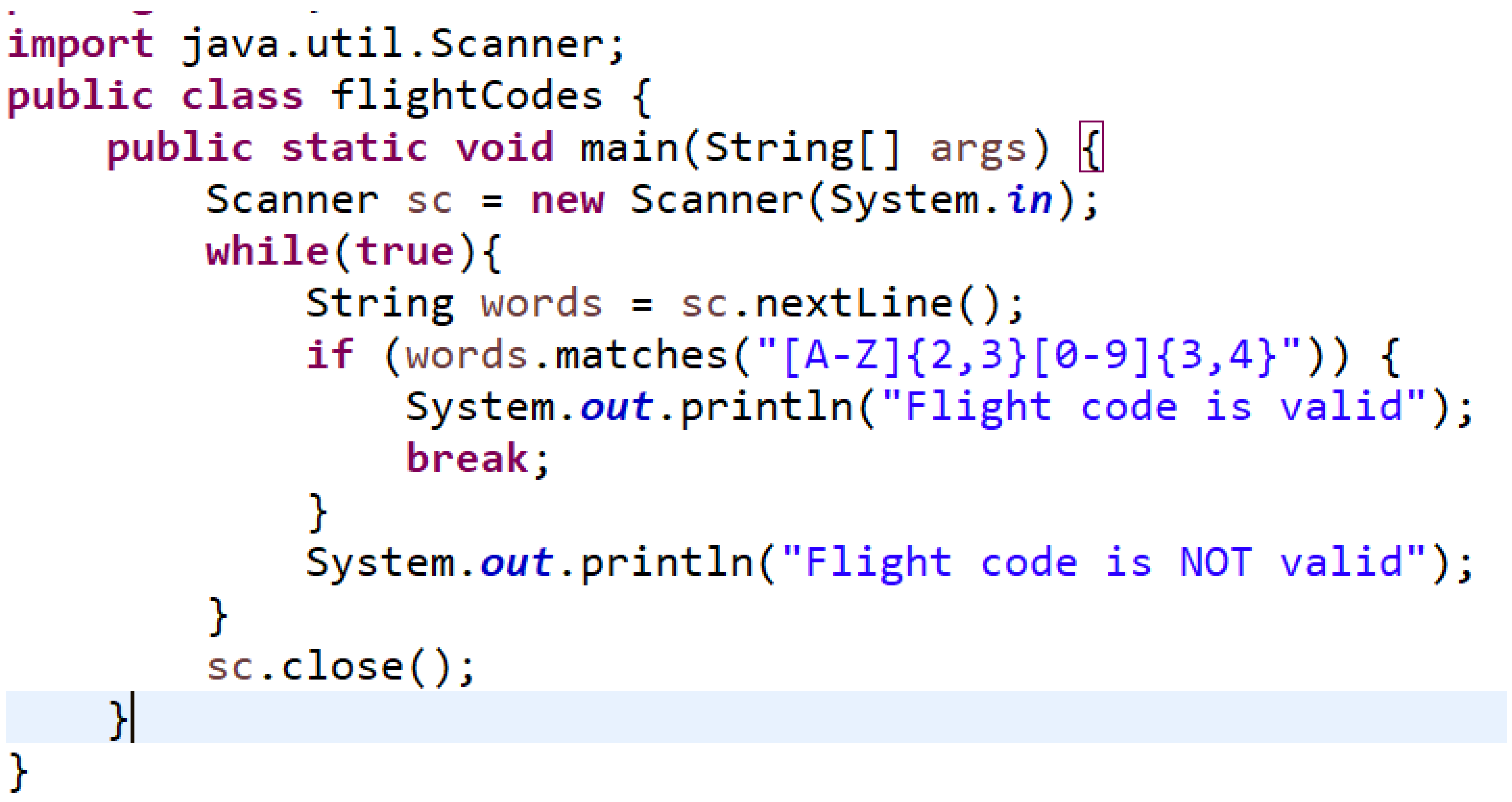
1) Sample run of program:

```
EI320                       //user input
Flight code is valid    //output based on value
```

2) Sample run of program:

```
rea4121                     //user input
Flight code is NOT valid  //output based on value of code
REA4121                          //user input
Flight code is valid        //output based on value
```

Solution



```
import java.util.Scanner;
public class flightCodes {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while(true){
            String words = sc.nextLine();
            if (words.matches("[A-Z]{2,3}[0-9]{3,4}")) {
                System.out.println("Flight code is valid");
                break;
            }
            System.out.println("Flight code is NOT valid");
        }
        sc.close();
    }
}
```

Lab3, Task 6

Using the code template given in Programming3.java and the dictionary.txt file your task is in pairs is to write a text-based version of Hangman. (https://www.wikihow.com/Play-Hangman). You should use your knowledge of methods and Linear Search to complete this task. Instead of drawing the gallows when a wrong guess is made you can use "lives" to indicate the number of guesses that remain.

In Programming3.java you are provided code that will read in all of the words from dictionary.txt and return a String array containing all of the words of length greater than eight. The program will then choose a random word from that list and return it for you to use as your word for the game. (DO NOT change any of the code provided).

IMPORTANT: The dictionary.txt file must be placed in the same folder as the PairProgramming3.java for the given code to work correctly.

Solution

- Demonstrate directly.