

```

/*  Havin Lim / Patrick Sales Garcia
Lab: 20 - Random Number Generation (Simple Games)
October 10th 2022
Sources : None
Help obtained : None
We confirm that the above list of sources is complete AND that we have not
talked to anyone else (e.g., CSC207 students) about the solution to this
problem
*/

// GuessMyNumber.java

package SimpleGames;

import java.util.Scanner;

/**
 *
 * @author Havin Lim / Patrick Sales Garcia
 *
 */
public class GuessMyNumber {

    /**
     * Method that generates a random number and asks the user to guess that
     number in 10 guesses
     * @return  boolean
     */
    public static boolean guessNumGame() {
        System.out.println("Enter your username !");
        Scanner sc = new Scanner(System.in);

        String userName;
        int answer;

        userName = sc.nextLine();

        System.out.println("Hello, " + userName + ". Let's play a
game!\n");

        int targetNum = (int) ((Math.random()*100)+1);
        int guess = 10;

```

```

        while(guess > 0) {
            System.out.println("Make a guess! The number is between 1
and 100.");
            System.out.println("\n You currently have " + guess + "
guesses left.");
            sc = new Scanner(System.in);
            answer = sc.nextInt();

            if(answer < targetNum) {
                System.out.println("Your guess is less than the target
number.");
                guess--;
            }
            else if (answer > targetNum) {
                System.out.println("Your guess is bigger than the
target number.");
                guess--;
            }
            else {
                System.out.println("Congratulations! You guessed the
number right!");
                return true;
            }
        }
        System.out.println("\n The number was " + targetNum + ".\nSorry,
you ran out of guesses! Try again later! :) ");
        return false;
    }

    public static void main(String[] args) {
        guessNumGame();
    }
}

```

/* Testing Output

Enter your username !

Havin

Hello, Havin. Let's play a game!

Make a guess! The number is between 1 and 100.

You currently have 10 guesses left.

50

Your guess is less than the target number.

Make a guess! The number is between 1 and 100.

You currently have 9 guesses left.

75

Your guess is bigger than the target number.

Make a guess! The number is between 1 and 100.

You currently have 8 guesses left.

62

Your guess is less than the target number.

Make a guess! The number is between 1 and 100.

You currently have 7 guesses left.

70

Your guess is bigger than the target number.

Make a guess! The number is between 1 and 100.

You currently have 6 guesses left.

66

Your guess is less than the target number.

Make a guess! The number is between 1 and 100.

You currently have 5 guesses left.

69

Congratulations! You guessed the number right!

*/

```

// MagicEightBall.java
package SimpleGames;

import java.util.ArrayList;
import java.util.Scanner;

/**
 *
 * @author Havin Lim / Patrick Sales Garcia
 *
 */
public class MagicEightBall {

    ArrayList<String> answers = new ArrayList<>();

    public MagicEightBall() {
        answers.add("It is certain.");
        answers.add("It is decidedly so.");
        answers.add("Without a doubt");
        answers.add("Yes definitely.");
        answers.add("You may rely on it.");
        answers.add("As I see it, yes.");
        answers.add("Most likely.");
        answers.add("Outlook good.");
        answers.add("Yes.");
        answers.add("Signs point to yes.");
        answers.add("Reply hazy, try again.");
        answers.add("Ask again later.");
        answers.add("Better not tell you now.");
        answers.add("Cannot predict now.");
        answers.add("Concentrate and ask again.");
        answers.add("Don't coun on it.");
        answers.add("My reply is no.");
        answers.add("My sources say no.");
        answers.add("Outlook not so good.");
        answers.add("Very doubtful.");
    }

    /**
     * Method that gives a random string from the constructor method previously
     * created
     * @return String randomly chosen from the MagicEightBall constructor method

```

```

    */
    public String returnAnswer() {
        int randomInt = (int) (Math.random()*20);
        return answers.get(randomInt);
    }
    /*
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MagicEightBall eightBall = new MagicEightBall();
        System.out.println("your QUESstIOOn?\n");
        String userQ;
        userQ = sc.nextLine();

        while(! userQ.toLowerCase().equals("no")) {
            System.out.println("\n" + userQ);
            System.out.println(eightBall.returnAnswer());
            System.out.println("your QUESstIOOn?\n");
            sc = new Scanner(System.in);
            userQ = sc.nextLine();
        }
    }
    */
}

```

/* **Testing Output**

your QUESstIOOn?

Should I sleep early today?

Should I sleep early today?

As I see it, yes.

your QUESstIOOn?

Can I eat McDonalds for dinner?

Can I eat McDonalds for dinner?

Cannot predict now.

your QUESstIOOn?

No

*/

```

// PigGame.java
package SimpleGames;

import java.util.Scanner;

/**
 *
 * @author Havin Lim / Patrick Sales Garcia
 */
public class PigGame {

    private int score1 = 0;
    private int score2 = 0;

    public PigGame() {

    }

    /**
     * Method that starts the pig game between two players
     */
    public void PIG() {
        System.out.println("Welcome to PIG!");

        int turnScore1 = 0;
        int turnScore2 = 0;
        Scanner sc = new Scanner(System.in);
        String answer;
        int dice;

        while (score1 < 100 && score2 < 100) {

            answer = "yes";

            while(!answer.toLowerCase().equals("stop")) {
                // This is player 1
            }
        }
    }
}

```

```

        dice = (int) ((Math.random()*6)+1);
        System.out.println("Player 1, your roll is " + dice);

        if(dice == 1) {
            turnScore1 = 0;
            System.out.println("You rolled 1 the turn goes
to the opponent.");
            break;
        }
        else {
            turnScore1 += dice;
        }

        System.out.println("Would you like to keep going?");
        System.out.println("If you would like to stop, type
\"stop\" ");

        sc = new Scanner(System.in);
        answer = sc.nextLine();
    }

    score1 += turnScore1;
    turnScore1 = 0;

    answer = "yes";

    if (score1 >= 100) {
        System.out.println("Congratulations! Player 1 won the
Pig Game!");
        break;
    }

    while(!answer.toLowerCase().equals("stop")) {
        // This is player 2
        dice = (int) ((Math.random()*6)+1);
        System.out.println("Player 2, your roll is " + dice);

        if(dice == 1) {
            turnScore2 = 0;
            break;
        }
        else {

```

```

        turnScore2 += dice;
    }

    System.out.println("Would you like to keep going?");
    System.out.println("If you would like to stop, type
\"stop\" ");

    sc = new Scanner(System.in);
    answer = sc.nextLine();
}

score2 += turnScore2;
turnScore2 = 0;

if (score2 >= 100) {
    System.out.println("Congratulations! Player 2 won the
Pig Game!");
    break;
}

}

}

public static void main (String[] args) {
    PigGame object = new PigGame();
    object.PIG();
}

}

```

/ Testing Output (for PigGame.java)*

```

Welcome to PIG!
Player 1, your roll is 4
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 1, your roll is 1
You rolled 1 the turn goes to the opponent.
Player 2, your roll is 3
Would you like to keep going?

```


If you would like to stop, type "stop"
q
Player 2, your roll is 3
Would you like to keep going?
If you would like to stop, type "stop"
r
Player 2, your roll is 2
Would you like to keep going?
If you would like to stop, type "stop"
w
Player 2, your roll is 6
Would you like to keep going?
If you would like to stop, type "stop"
w
Player 2, your roll is 4
Would you like to keep going?
If you would like to stop, type "stop"
stop
Player 1, your roll is 1
You rolled 1 the turn goes to the opponent.
Player 2, your roll is 2
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 3
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 1
Player 1, your roll is 4
Would you like to keep going?
If you would like to stop, type "stop"
stop
Player 2, your roll is 2
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 3
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 4

Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 3
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 3
Would you like to keep going?
If you would like to stop, type "stop"
stop
Player 1, your roll is 6
Would you like to keep going?
If you would like to stop, type "stop"
stop
Player 2, your roll is 2
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 5
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 5
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 3
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 6
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 2
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 5
Would you like to keep going?
If you would like to stop, type "stop"

stop
Player 1, your roll is 5
Would you like to keep going?
If you would like to stop, type "stop"
stop
Player 2, your roll is 6
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 6
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 3
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 5
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 6
Would you like to keep going?
If you would like to stop, type "stop"
stop
Player 1, your roll is 5
Would you like to keep going?
If you would like to stop, type "stop"
stop
Player 2, your roll is 3
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 2
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 1
Player 1, your roll is 5
Would you like to keep going?
If you would like to stop, type "stop"
stop

Player 2, your roll is 1
Player 1, your roll is 6
Would you like to keep going?
If you would like to stop, type "stop"
stop
Player 2, your roll is 1
Player 1, your roll is 3
Would you like to keep going?
If you would like to stop, type "stop"
stop
Player 2, your roll is 6
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 1
Player 1, your roll is 1
You rolled 1 the turn goes to the opponent.
Player 2, your roll is 5
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 3
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 5
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 4
Would you like to keep going?
If you would like to stop, type "stop"
q
Player 2, your roll is 6
Would you like to keep going?
If you would like to stop, type "stop"
stop
Congratulations! Player 2 won the Pig Game!
*/

```
/* Havin Lim / Ahmed Cheema / Emily Rhoades
```

```
Lab: 21 - Recursion
```

```
October 14th 2022
```

```
Sources : None
```

```
Help obtained : None
```

```
We confirm that the above list of sources is complete AND that we have not  
talked to anyone else (e.g., CSC207 students) about the solution to this  
problem
```

```
*/
```

```
// Convert.java
```

```
package Lab21;
```

```
public class Convert {
```

```
    private static String store = "";
```

```
    /**
```

```
     * This method takes a number and then converts it to the given base
```

```
     * @param num int to be converted
```

```
     * @param base int of desired base
```

```
     * @return int of converted number
```

```
     */
```

```
    public static int convert(int num, int base) {
```

```
        if (base <= 0 || num < 0)
```

```
            throw new IllegalArgumentException("Base must be greater  
than 0 and number must be 0 or greater.");
```

```
        if (num == 0) {
```

```
            return(Integer.parseInt(store));
```

```
        }
```

```
        else {
```

```
            String insert = (" " + num % base);
```

```
            store = insert + store;
```

```
            return convert(num / base, base);
```

```
        }
```

```
    }
```

```
    public static void main(String args[]) {
```

```
        System.out.println(convert(100, 2));
```

```
    }
```

```
}
```

```
/* Testing Output (for Convert.java)
(for convert(100,2))
1100100
(for convert(100,8))
144
(for convert(100,16))
64
*/
```

```
// Fibonacci.java
```

```
package Lab21;
```

```
public class Fibonacci {  
    /**  
     * Method that calculates number for the fibonacci sequence  
     * @param n1 the initial value to be added to the sequence  
     * @param n2 the second value to be added to the sequence  
     * @param num the desired index of the fibonacci sequence  
     * @return int  
     */  
    public static int fibonacciHelper(int n1, int n2, int num) {  
        if (num==0)  
            return 0;  
        else if(num<=2)  
            return n1+n2;  
  
        return fibonacciHelper(n2, n1+n2, num - 1);  
    }  
  
    /**  
     * Method that calculates number for the fibonacci sequence  
     * @param n the desired index of the fibonacci sequence  
     * @return int  
     */  
    public static int fibonacci(int n) {  
        if(n < 0) {  
            throw new IllegalArgumentException("Input number must be  
greater or equal to 0");  
        }  
        return(fibonacciHelper(0, 1, n));  
    }  
  
    public static void main(String[] args) {  
        System.out.println(fibonacci(10));  
        System.out.println(fibonacci(8));  
        System.out.println(fibonacci(5));  
    }  
}
```

```
/* Testing output (for Fibonacci.java)
55
21
5
*/
```



```

// Sequences.java
package Lab21;

public class Sequences {

    /**
     * Method that returns string with the a symmetric sequence of n
     * numbers composed of descending integers that ends in 1, followed
     * by a sequence of ascending integers that begin with 1
     * @param n length of the symmetric sequence
     * @return String
     */
    public static String writeSequence2(int n) {
        if (n == 1 ) {
            return "1";
        }
        else if (n==2) {
            return "1 1";
        }
        else {
            int n2 = (int) Math.round(n / 2.0);

            return (n2 + " " + writeSequence2(n-2) + " " + n2);
        }
    }

    /**
     * Method that prints the symmetric sequence
     * @param n length of the symmetric sequence
     */
    public static void writeSequence(int n) {
        if(n < 1) {
            throw new IllegalArgumentException("Input number must be
greater or equal to 1");
        }
        System.out.println(writeSequence2(n));
    }

    public static void main(String[] args) {
        writeSequence(10);
        writeSequence(9);
    }
}

```

```
        writeSequence(5);
    }
}

/* Testing output (for Sequences.java)
5 4 3 2 1 1 2 3 4 5
5 4 3 2 1 2 3 4 5
3 2 1 2 3
*/
```