*Assignment 5 - 6*

*Instructor: Faisal Khan*

|  |  |  |
| --- | --- | --- |
| *Name* | *Student ID* | *Group#* |
| Alexander Gutierrez | c0895239 | 1 |
| Can Zorbey | c0895400 | 1 |
| David Barrios | c0893262 | 1 |
| Paul Jordan Untalan | c0899319 | 1 |

1-Use Documentation Comments where needed.

2-In your project src folder add this word document with all your group members names, having the screen shots of your output with source code after each question.

2-Right Click on your Project Folder in Eclipse, Copy Option is available. Create a copy of the Project and upload the zipped folder using Moodle.

|  |  |
| --- | --- |
| **Assignment 5** | **10** |

Coin Toss Simulator

Write a class named Coin. The Coin class should have the following field:

• A String named sideUp. The sideUp field will hold either “heads” or “tails” indicating the side of the coin that is facing up.

The Coin class should have the following methods:

• A no-arg constructor that randomly determines the side of the coin that is facing up (“heads” or “tails”) and initializes the sideUp field accordingly.

• A void method named toss that simulates the tossing of the coin. When the toss method is called, it randomly determines the side of the coin that is facing up (“heads” or “tails”) and sets the sideUp field accordingly.

• A method named getSideUp that returns the value of the sideUp field.

Write a program that demonstrates the Coin class. The program should create an instance of the class and display the side that is initially facing up. Then, use a loop to toss the coin 20 times. Each time the coin is tossed, display the side that is facing up. The program should

keep count of the number of times heads is facing up and the number of times tails is facing up, and display those values after the loop finishes.

**Source Code:**

package com.lcit.assignments;

import java.util.Random;

/\*\*

\* This program consist of two classes, one of them is the Coin class. This

\* class has a private field named sideUp, that will hold the value "tails" or

\* "heads". It also has a no-arg constructor used to initialized the value of

\* the sideUp field. To toss the coin this class comes with a public method

\* called toss which is using a Random object to get a boolean value and

\* according to this value assign either "tails" or "heads" to the sideUp field.

\* Because the sideUp field is private this class provides a getter (accessor)

\* to get the value of the sideUp field. On the main method a Coin instance is

\* created, this coin object is tossed 20 times, and the program displays and

\* keeps record of the side up for every toss action.

\*

\* @param args

\*/

public class Assignment\_5 {

public static void main(String[] args) {

Coin coin = new Coin();

int headsCounter = 0, tailsCounter = 0;

System.out.println("Initial side facing up: " + coin.getSideUp());

System.out.println("\nTossing the coin 20 times...");

System.out.println("\nSide up");

for (int i = 1; i <= 20; i++) {

coin.toss();

System.out.print(coin.getSideUp() + ((i % 10 == 0) ? "\n" : ", "));

if (coin.getSideUp().equals("heads")) {

headsCounter++;

} else {

tailsCounter++;

}

}

System.out.println("\nTotal\n" + "heads tails\n" + "-------------");

System.out.printf("%-8s %s", headsCounter, tailsCounter);

}

}

class Coin {

private String sideUp;

public Coin() {

toss();

}

public void toss() {

Random randomNumber = new Random();

sideUp = randomNumber.nextBoolean() ? "heads" : "tails";

}

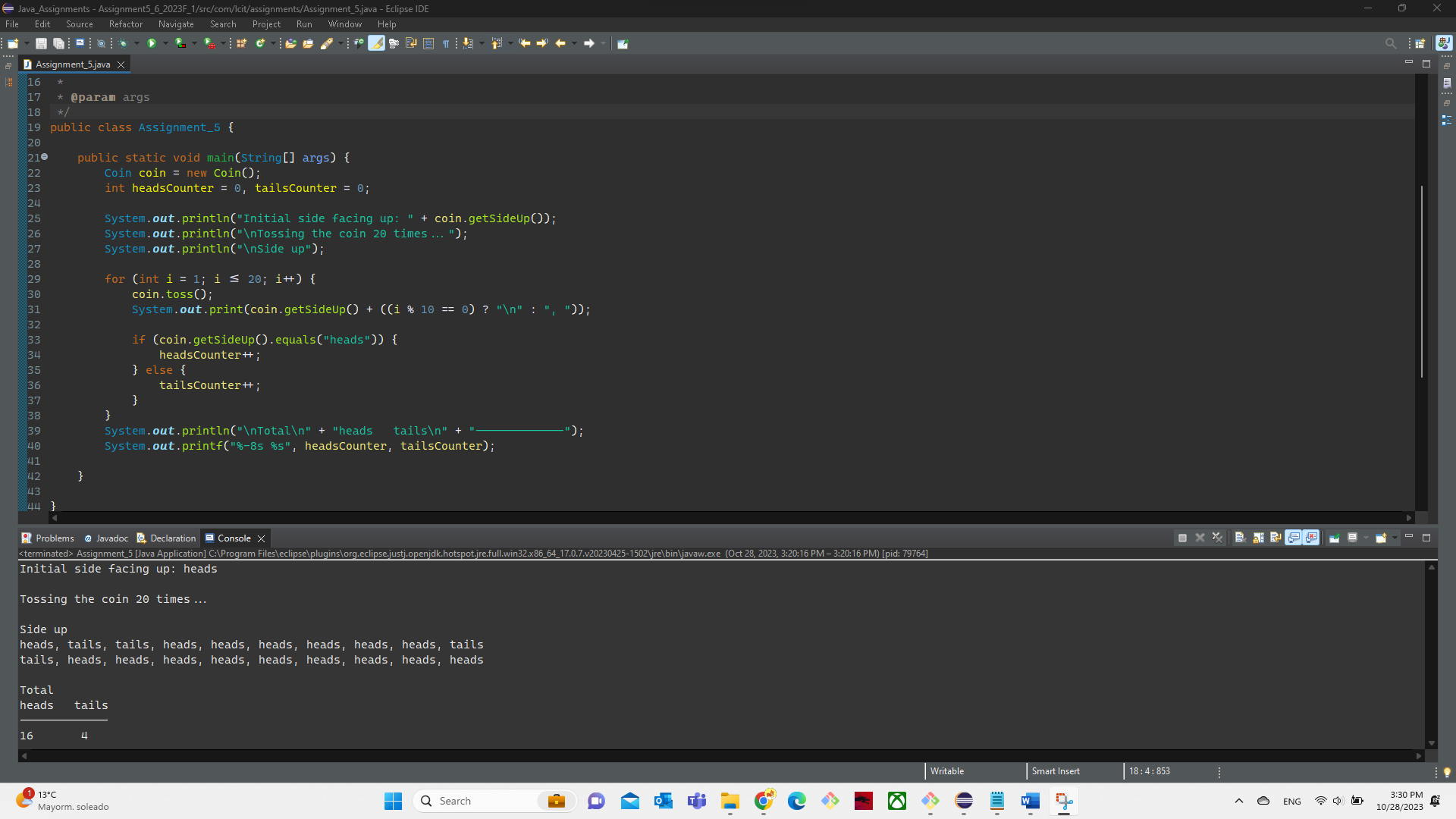
public String getSideUp() {

return sideUp;

}

}

**Output:**



|  |  |
| --- | --- |
| **Assignment 6** | **10** |

Write a program that asks the user to enter today’s sales for five stores. The program should

display a bar chart comparing each store’s sales. Create each bar in the bar chart by displaying

a row of asterisks. Each asterisk should represent $100 of sales. Here is an example of

the program’s output:

Enter today's sales for store 1: 1000 [Enter]

Enter today's sales for store 2: 1200 [Enter]

Enter today's sales for store 3: 1800 [Enter]

Enter today's sales for store 4: 800 [Enter]

Enter today's sales for store 5: 1900 [Enter]

SALES BAR CHART

Store 1: \*\*\*\*\*\*\*\*\*\*

Store 2: \*\*\*\*\*\*\*\*\*\*\*\*

Store 3: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Store 4: \*\*\*\*\*\*\*\*

Store 5: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Source Code:**

package com.lcit.assignments;

import java.util.Scanner;

public class Assignment\_6 {

/\*\*

\* The Sales Bar Chart program takes 5 number inputs from the user. Each input

\* corresponds to the total sales of each store.

\*

\* The output will show a horizontal graph showing "\*", which denotes the total

\* sales of each store. Each asterisk "\*" denotes a sales of $100.

\*

\* As an example, if Store 1 has $600 Total Sales, the graph should show:

\*

\* Store 1: \*\*\*\*\*\* ($600 = 6 asterisks)

\*

\* There are two approaches to produce the asterisks. Please see the single line

\* comments in the source code.

\*

\* @param args

\*/

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

double AmountOfSales;

String barChart = "\n------- SALES BAR CHART -------\n";

// String chart = "";

for (int i = 1; i <= 5; i++) {

System.out.printf("Enter today's sales for store %d: ", i);

AmountOfSales = input.nextDouble();

// first approach. using the repeat() method:

barChart += String.format("Store %d: %s\n", i, "\*".repeat((int) AmountOfSales / 100));

// Another approach using a nested for loop:

// for (int j = 0; j < (AmountOfSales / 100); j++) {

// chart += "\*";

// }

// barChart += String.format("Store %d: %s\n", i, chart);

// chart = "";

}

System.out.println(barChart);

input.close();

}

}

**Output:**

A computer screen shot of a program

Description automatically generated