Java Assignments (Java – 001)

Table of Contents

Instructions

All assignments will be created in a root folder:
`java_training/assignements/java_001/`

1. Ensure all assignments can compile
2. Ensure all assignments have comments
3. Each assignment will be a folder/project of their own, to be included in the above root folder

Delivery (Due Date/Time)

Your assignment are due on Monday July 8th at the beginning of the class. (8 am CST)

The instructor will work with you to ensure you can upload this to your GitHub repo

Assignment 1: Head or Tail

Overview

The goal of the program to write is to display “The Coin Flip is: Head” or “The Coin Flip is: Tail” every time it executes.

Specifications

- You will create a method that will return a value of either 1 or 2.
- You will create another method that will take a value and will assign “Head” if 1 is passed, “Tail” if 2 is passed.

Expectations

Your main function will print out the message:

- `“Coin Flip Program”`
  `“The Coin Flip is: Head”`

or

- `“Coin Flip Program”`
  `“The Coin Flip is: Tail”`

Save Project Name as: **`Assignment01_HeadOrTail`**

Assignment 2: Keeping Stats

Overview

The goal of the program to display a tally of occurrences of the state of “Head” and/or “Tail” in an iteration loop.

Specifications

- You will loop 1000 times
- You will have an single dimensional array name tally and it will be holding an integer value and it will be able to holder 2 integers
- You will ensure that all values in the tally array are initialized with the value of 0
- Each occurrence in the loop of either 1 or 2 will be ‘tallied’ as a count and you must account for the fact that array index are zero bound (0), so you must ensure you increase the correct array index value by 1
- At the end of the loop, around tally array should contain a count for each occurrence of 1 and 2
- If the loop and tally process is correct, the summation of the 2 will result in 1000.
- By the end of the loop, you will display a 2 lines message which will display the number of occurrences for each state of “Head” and “Tail”.
- Use Assignment 1 as a reference for specifications on “Head” and “Tail” values

Expectations

Your main function will print out the message:

- `“1000 Coin Flips”`
  `“Count of Head: XXX”`
  `“Count of Tail: XXX”`

Save Project Name as: **`Assignment02_HeadOrTailTally`**

Assignment 3: Throw a die

Overview

This program will allow the user to display a die and will keep asking the user if they wish to throw another one or ask the user to exit the program.

Specifications

- Ask the user to press any Key and Enter to throw a die
- Display a message of which die face is shown
- Ask the user if they wish to play again (Y or y) another other key will quit the game
- Value of die can be between 1 and 6 inclusively.

Expectations

Your main function will print out the message:

- `"Press any key to throw a die and press Enter (or Q and Enter to quit)"`
- `"You have rolled a 1"`
- `"Play Again? (Y or y) and Enter, any other key and Enter to Quit"`

Save Project Name as: **`Assignment03_SimpleDie`**

Added Note: Because you are executing your code in an IDE console, nice features such as being able to clear the console are not available programmatically. However, those who decide to try their hand at running their code in an actual command prompt window or (OS) terminal, you can research the topic of programmatically clearing a Java console at run-time. This is optional and not required for your project

Assignment 4: Pair of Dice anyone?

Overview

This program will allow the user to display the result of throwing 2 dice.

Specifications

- Ask the user to press any Key and Enter to throw a die
- Display a message of which die face is shown
- Ask the user if they wish to play again (Y or y) another other key will quit the game
- Value of each die can be between 1 and 6 inclusively.

Expectations

Your main function will print out the message:

- `"Press any key to throw a pair of dice and press Enter (or Q and Enter to quit)"`
- `"You have rolled the following: "`
- `"First Die: X"`
- `"Second Die: Y"`
- `"Play Again? (Y or y) and Enter, any other key and Enter to Quit"`

Save Project Name as: **`Assignment04_PairOfDice`**

Assignment 5: Keeping Way More Stats

Overview

Based on Assignment 3 and 4, we will now keep track of dice rolls in a way where we tally the types of pair combo rolled.

Specifications

- Since each die has 6 faces from 1 to 6, you have a total of 21 unique combinations of roles, assuming you do not care about the order of the dice
  i.e. (2, 1) or (1, 2) are the same
- Ensure you have an object of 21 unique combinations that you will tally and set all initial values in the array with the value of 0
- Program will iterate through 1000 times
- For each iteration you will be 'rolling' 2 dice
- sort each iteration by high, low, so if you role a 4,3 or a 3,4, you will tally a 4,3, you can consider the result of this as "an occurrence"
- Increase the count of occurrence by 1
- At the end of the loop, create a report which will display in order of count of occurrence from high to low

Expectations

Your main function will print out the message:

- `"1000 Random 2 dice toss stats"`
  `"Dice 2 and 1: 300 times"`
  `"Dice 6 and 5: 100 times"`
  `"Dice 4 and 3: 0 occurrence"`

  If the report is correct, the tally of (times) will be 1000 and you should have a total of 2 lines including the initial header

Save Project Name as: **`Assignment05_DicePairStats`**

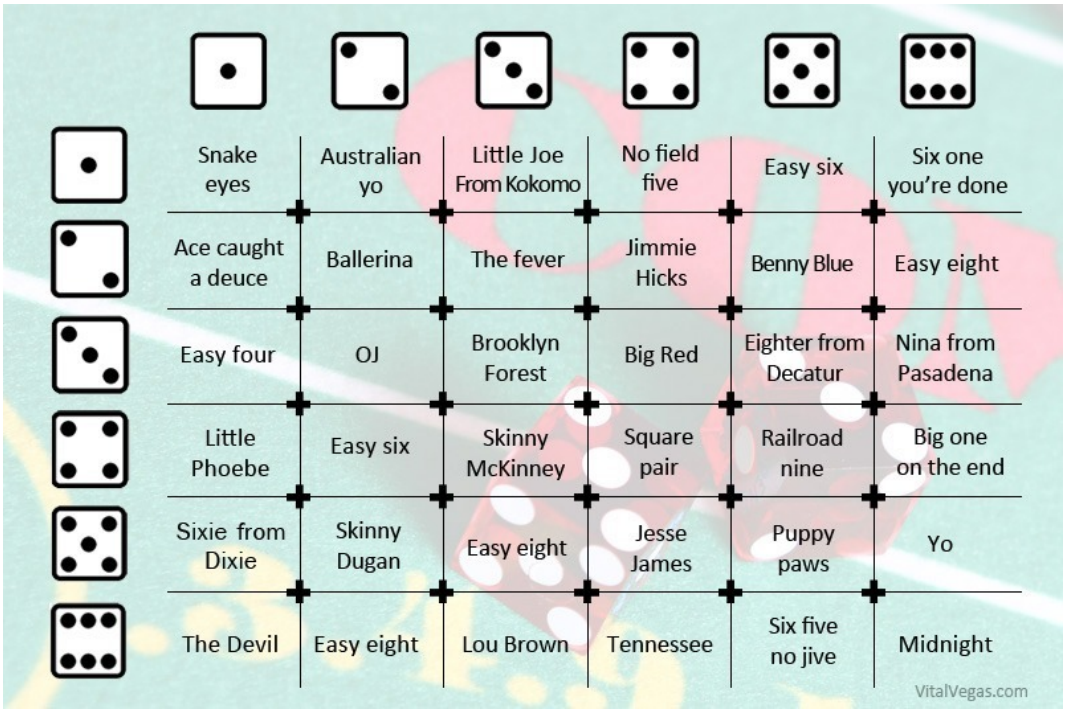
Assignment 6: Labelling Pair of Dices (Matrix)

Overview

Based on Assignment 3, 4 and 5, we will now create a two-dimensional matrix where each unique combination of 2 dices are assigned a label.

Specifications

- The total amount of combinations of 2 six sided die is 36.
- We will create an array where for each combination, a 'label' will be assigned.
- We will then print out the array 'matrix' like using formatting characters such as tab
- You will need to hold all this data into a two-dimensional array
- The graphic below is the matrix you are recreating programmatically



Expectations

Your main function will print out the message:

```
"Dice Combinations Label Matrix"
"       Die 1               Die 2           Die 3                       Die 4           Die 5       Die 6

Die 1   Snake Eyes          Autralian Yo    Little Joe from Kolomo      No Field Five   Easy Six    Six one you're done

Die 2   Ace caught a deuce  Ballerina       The Fever                   Jimmie Hicks    Benny Blue  Easy eight

…
```

You will need to eye-ball the report in your console and adjust the numbers of tabs per columns so that it actually looks formatted as a grid. The columns will be straight, but not necessary of equal width to each other.

Save Project Name as: **`Assignment06_DiceLabels`**

Assignment 7: Game of Dice

Overview

You will create a program that will pit your luck against the computer in a game of dice throws.

Specifications

- The computer will challenge you to a friendly game of dice throw
- If you accept, you will be asked to throw the dice
- The computer will do the same
- Your score's will be compared, one is a winner, one is a loser, or you have both tied.
- You will be asked if you wish to play again.
- Each throw from player and computer will use the matrix label system from the previous assignment
- A winner means
    - Highest summed value, so (1, 6) vs (2, 4) the (1, 6) is a winner because the sum is 7.
    - if both players have identical summed values
        - the one who threw a double will win (for example, (4, 4) vs (5, 3), the (4, 4) will win)
        - else you tie (for example (2, 1) vs (1, 2) is a tie
- The "Expectations" portion of this assignment provides more guideline of the program's behavior

Expectations

Start of the program

```
"Welcome to the dice throw challenge!"
"Do you feel lucky? "Punk"… Do ya?"
"Press any key and ENTER to throw your dice or press (Q or q) to chicken your
butt off…"
```

… assuming they press any key… but not Q/q

```
"Rolling the dice!!"
```

.. if the computer wins

"You have rolled a 1 and 2 > the "Autralian Yo""
"I have rolled a 3 and 4 > the "Big Red""
"I win!!!"
"Wanna try again? Press any key and ENTER to throw your dice or press (Q or q) to quit"

.. if the computer looses

"You have rolled a 3 and 4 > the "Big Red""
"I have rolled a 1 and 2 > the "Autralian Yo""
"You win!!!"
"Did you cheat?? Gimme another try, c'mon Press any key and ENTER to throw your dice or press (Q or q) to quit"

.. if computer looses on a tie because of a double

"You have rolled a 4 and 4 > the "Square Pair""
"I have rolled a 5 and 3 > the "Easy Eight"
"Damn! You win on a Double! What Luck!"
"Did you cheat?? Gimme another try, c'mon Press any key and ENTER to throw your dice or press (Q or q) to quit"

.. if computer wins on a tie because of a double

"You have rolled a 5 and 3 > the "Easy Eight"
"I have rolled a 4 and 4 > the "Square Pair""
"Damn! I win on a Double! Ah! I pity you fool!"
"Wanna try again? Press any key and ENTER to throw your dice or press (Q or q) to quit"

.. if it's a real tie regardless of how it is achieved

"You have rolled a 5 and 3 > the "Easy Eight"
"I have rolled a 3 and 5 > the "Eighter from Decatur"
"Stalemate! You're tough, let's try for a tie-breaker Press any key and ENTER to throw your dice or press (Q or q) to quit"

.. As long as you keep playing..

"Rolling the dice!!"
…
…

… When player quits and there was never any game played

"See ya later alligator!"

… When player quits and there was at least one round of play

"Thanks for being a sport! laters!"

Save Project Name as: **Assignment07_GameOfDice**

Assignment 8: String Array Utilities

Overview

We will be creating methods which allows us to determine the content of an array of string.

Specifications

method 1: boolean arrayHasExactMatch(array, string, boolean)

this method will return true or false if an exact match is found or false is there are no items to be found

the 3rd parameter is a boolean, to look for an exact match string which might not be in the same case, so false means not case sensitive and true means case sensitive

the return value is a boolean

if an array as the following values:

```
String myList[] = {"Bozo", "FooBar", "Delta", "Foozball", "Demo", "Money",
"Zoo"};
boolean found;

found = arrayHasExactMatch (myList, "zo", false);
// found will be false
found = arrayHasExactMatch (myList, "zoo", false);
// found will be true
found = arrayHasExactMatch (myList, "zoo", true);
// found will be false
```

method 2: int[] indexOfOccuranceInArray(array, string, boolean)

this method will return an array of one or more indexes that matches the string to be found, it is based on arrayHasExactMatch.

should there be no items found the array will be seeded with a -1 value.

```
String myList[] = {"Bozo", "FooBar", "Delta", "Foozball", "Demo", "Money",
"Zoo"};
int foundIndexes[];

foundIndexes = indexOfOccuranceInArray (myList, "zo", false);
// foundIndexes will contain [-1]
foundIndexes = indexOfOccuranceInArray (myList, "zoo", false);
// foundIndexes will contain [5]
foundIndexes = indexOfOccuranceInArray (myList, "zoo", true);
// foundIndexes will contain [-1]
```

Expectations

Your main function will print out the message should be able to run various samples and your code comment should provide the expectation of the output

This should be your list:

```
String myList[] = {"Bozo", "FooBar", "Delta", "Foozball", "Demo", "Money",
"Zoo"};
```

For both arrayHasExactMatch and indexOfOccuranceInArray, ensure you cover the following scenarios

```
arrayHasExactMatch (myList, "zo", false);
arrayHasExactMatch (myList, "zoo", false);
arrayHasExactMatch (myList, "zoo", true);
arrayHasExactMatch (myList, "foo", true);
arrayHasExactMatch (myList, "foo", false);
arrayHasExactMatch (myList, "delta", true);
arrayHasExactMatch (myList, "Delta", true);
```

```
indexOfOccuranceInArray (myList, "zo", false);
indexOfOccuranceInArray (myList, "zoo", false);
indexOfOccuranceInArray (myList, "zoo", true);
indexOfOccuranceInArray (myList, "foo", true);
indexOfOccuranceInArray (myList, "foo", false);
indexOfOccuranceInArray (myList, "delta", true);
indexOfOccuranceInArray (myList, "Delta", true);
```

Your output should make sense, a single line per scenario in this format:

```
"This is the a test of the arrayHasExactMatch and indexOfOccurenceInArray methods"
"The array to test contains the following items"
"{"Bozo", "FooBar", "Delta", "Foozball", "Demo", "Money", "Zoo"}"
"Scenario 1"
"arrayHasExactMatch (myList, "zo", false): is false"

…

"Scenario 8"
"indexOfOccuranceInArray (myList, "zo", false) returns [-1]"

"Scenario 9"
"indexOfOccuranceInArray (myList, "zoo", false) returns [5]"

…
```

Save Project Name as: **Assignment08_StringArrayUtilities**

Assignment 9: String Array Utilities (Part 2)

Overview

Based on the previous assignment, the goal is to create a package named “arrayutils” and to create a static class/method which are public and can be used without instantiating the class.

Specifications

- Your package name should be “arrayutils” for the methods you will create as static
- You will create another package name “app” and ensure it has a class with the public static void main() method

Expectations

Exact same output, your code is refactored to ensure you correctly call the methods from your package.

Save Project Name as: **`Assignment09_StringArrayUtilities2`**

Note: your ‘class’ should be “ArrUtil” in your package.