By: Hussain Alsafar

November 3, 2016

Computer Science. 1033

**Single and Multi-Dimensional Arrays**

***Introduction:***

In this project, I have 2 programs one for chapter 7 and the other for chapter 8. Both chapters are about Arrays, Single and Multi-Dimensional Arrays. According to techopedia.com, an array, in the context of Java, is a dynamically-created object that serves as a container to hold constant number of values of the same type. By declaring an array, memory space is allocated for values of a particular type. At the time of creation, the length of the array must be specified and remains constant. In first one we build programs by using one-dimensional arrays to store linear collections of elements and use it to store a matrix or a tables. The other one using single array variables that can reference a large collection of data.

***Discussion:***

First code 7.29, is basically a Game, pick four cards game. The program should picks four cards from a deck of 52 cards and the program should computes their sum. More, the program should display the number of picks yields the sum of 24. It will ask the user to enter three numbers. The program will display them in increasing order. On the output, the program will read An Ace, King, Queen, and Jack that represent 1, 13, 12, and 11, respectively. The Second Code is a Game too. The game is “nine heads and tails”. It is similar to the flipping coins game that we play by hand. The Nine coins here are placed in a 3-by-3 matrix with some face up and some face down. The exercise says that the Values 0 (heads) and 1 (tails) each state can also be represented using a binary number. The numbers 0, 1, 2, 3, and 511 to represent all states of the matrix. The program should ask a user to enter a number between 0 and 511 and the output should read it using matrix with the characters H and T.

***Summary:***

The two programs one for chapter 7 and the other for chapter 8. Both chapters are about Arrays, Single and Multi-Dimensional Arrays. In first one we build programs by using one-dimensional arrays to store linear collections of elements and use it to store a matrix or a tables. The other one using single array variables that can reference a large collection of data.

7.29:Code

**public** **class** h29 {

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

// **TODO** Auto-generated method stub

// Create the array

**int**[] deck;

**do** {

//deck

deck = **new** **int**[52];

// Pick four cards

*pickFourCards*(deck);

} **while** (*sum*(deck) != 24);

// Display the number that yields the sum of 24

*print*(deck);

}

//Randomly picks four cards

**public** **static** **void** pickFourCards(**int**[] deck) {

**for** (**int** i = 0; i < 4; i++) {

deck[(**int**)(Math.*random*() \* 52)]++;

}

}

//sum computes the sum of cards picked \*/

**public** **static** **int** sum(**int**[] deck) {

**int** sum = 0;

**for** (**int** i = 0; i < deck.length; i++) {

sum += ((i + 1) % 13) \* deck[i];

}

**return** sum;

}

//displays the picks

**public** **static** **void** print(**int**[] deck) {

String[] ranks = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9",

"10", "Jack", "Queen", "King"};

System.***out***.print("The number of picks that yields the sum of 24: ");

**for** (**int** i = 0; i < deck.length; i++) {

**if** (deck[i] > 0)

System.***out***.print(ranks[i % 13] + " ");

}

System.***out***.println();

}

}

Console:

The number of picks that yields the sum of 24: 6 King Jack 7

8.11 Code:

import java.util.Scanner;

public class huss8\_9 {

public static void main(String[] args) {

// TODO Auto-generated method stub

int[][] m = new int[3][3];

Scanner input = new Scanner(System.in);

System.out.print("Enter a number between 0 and 511: ");

int n = input.nextInt();

String binary = decimalToBinaryModified(n);

// put 1's and 0's using binary string

int bIndex = 0;

// binary string index

for (int i = 0; i < m.length; i++) {

for (int k = 0; k < m[i].length; k++) {

int coinSide = (binary.charAt(bIndex++) == '0') ? 0 : 1;

m[i][k] = coinSide;

}

}

for (int i = 0; i < m.length; i++) {

for (int k = 0; k < m[i].length; k++) {

char ch = (m[i][k] == 0) ? 'H' : 'T';

System.out.print(ch + " ");

}

System.out.println("");

}

}

public static String decimalToBinaryModified(int n) {

StringBuilder s = new StringBuilder();

while (n != 0) {

s.append(n & 1);

n = n >> 1;

}

while (s.length() < 9) {

s.insert(0, "0");

}

return s.toString();

}

}

Console:

Enter a number between 0 and 511: 60

H H H

H H T

T T T

Reference:

I practice the programs by using GitHub Websit

https://github.com/jsquared21/Intro-to-Java-Programming/blob/master/Exercise\_07/Exercise\_07\_29/Exercise\_07\_29.java

https://github.com/LuizGsa21/intro-to-java-10th-edition/blob/master/src/Chapter\_08/Exercise\_11.java