

AssessedExerciseTemplate-level5-Copy1

November 3, 2020

1 Short Assessed Exercise

2 Level 5

2.1 Ifthy Fairoze

2.2 30 October 2020

2.3 Version 1

2.4 Summary of the Question

Write a program that uses a for loop to ask the user to name 4 female footballers. They should say how many times each has scored in the World Cup and what their country is. The names of the footballers should be kept in one array, the number of goals in a second array in the corresponding position, and their country in a third. A final variable should be used to store the array size.

The program should then name the footballer that was top scorer and the total number of goals all these players scored. It should finally print out each player's information in a list with commas separating the name, country and goals scored.

```
Name footballer 1? Ellen White
How many goals did they score? 6
What country did they play for? England
```

```
Name footballer 2? Megan Rapinoe
How many goals did they score? 7
What country did they play for? USA
```

```
Name footballer 3? Ajara Nchout
How many goals did they score? 2
What country did they play for? Cameroon
```

```
Name footballer 4? Cristiane
How many goals did they score? 4
What country did they play for? Brazil
```

The highest scorer was Megan Rapinoe.
These players scored 19 goals between them.

The details of the players are:

Ellen White, England, 6
Megan Rapinoe, USA, 7
Ajara Nchout, Cameroon, 2
Cristiane, Brazil, 4

2.5 The literate program development

Building on the what I have learned in from the ECS401U Jupyter Notebook, to answer the question, I created six methods of `inputInt`, `inputString`, `index`, `total`, `printDetails` and `player_details`. For each method, I will explain about the method, what it does and how I tested it and provide the code of the method itself. Next, I show how to run the solution to the question. Each code fragment can be run in Jupyter notebooks, here. Finally, I show the full program that can only be run outside of notebooks, along with instructions on how to do this and a summary of the structure of the full program.

2.5.1 `inputInt`

What it does This method asks the user a question to the user which is printed. Then waits for the user to enter a response (integer response). Then returning the response as a result of the method.

Implementation (how it works) It sets up a local scanner to the method, linked to the keyboard (done from the first line in method). This makes the method `scanner.nextLine()` available. Then it prints out the question to the user. Then it allows the user to type a response (takes a whole String up until the user hits enter as the response) using `nextLine`. This read and stored into `textInput` variable as String value. Then the value of `textInput` is transformed to integer type and stored into `numberInput` variable which is an integer type. The the value in `numberInput` is returned as a result of this method.

```
[9]: /* This method asks a String question to the user which then allows the user to  
→type a integer response which is  
returned as a result of the method. */  
public static int inputInt(String message)  
{  
    Scanner scanner = new Scanner(System.in);  
    String textInput;  
    int numberInput;  
    System.out.println(message);  
    textInput = scanner.nextLine();  
    numberInput = Integer.parseInt(textInput);  
    return numberInput;  
} // END inputInt
```

Testing

```
[10]: System.out.println(inputInt("What is the answer to this question?"));
```

What is the answer to this question?

5

2.5.2 inputString

What it does This asks the user a String question which is printed and waits for the user to enter a response (where response must be String) to that question and returns that response.

Implementation (how it works) In the first line of the method it sets up a local scanner to the method, linked to the keyboard. This makes available the method `scanner.nextLine()`. Next the question is printed and allows the user to enter a response of a whole String up until the user hits enter which is stored in answer variable (String type). Then returns that response stored in answer as a result of the method.

```
[11]: /* This method prints out a question to the user and then allows the user to  
↪ enter a response (which is an integer  
response) and returns that response as a result of this method. */  
public static String inputString(String message)  
{  
    Scanner scanner = new Scanner(System.in);  
    String answer;  
    System.out.println(message);  
    answer = scanner.nextLine();  
    return answer;  
} // END inputString
```

Testing

```
[12]: System.out.println(inputString("What is the answer to this question?"));
```

What is the answer to this question?

I don't know

I don't know

2.5.3 index

What it does The method index finds the index of the maximum value of an integer type array. This integer array has to be initialised and its size must be given. Both must be given as arguments when calling this method in the program. Then the index which contains the maximum value is returned as result of the method.

Implementation (how it works) First the parameters are declared which will be used in the method. The parameters are the integer array called `arr` and variable `size` (integer type). This is done to get the array and size through as an argument when calling the method. Then we declare and initialise the variable `indexMax` with value 0 which will be used in the for loop. The for loop is then used which is looped for the size of the array times so it can go through each value in the array using the size variable. Then an if statement is used inside the loop where the test is if the index of value of `i` of the array is greater than index of value of `indexMax` of the array. If this is true then `indexMax` gets a copy of the value of `i`. This is done for each value of the array to compare and get

the index of the maximum value in the array which is stored in indexMax and then returned as a result.

```
[13]: /* This method finds the index or position of the highest value in an integer ↵  
      ↪array. Then it returns the  
      index number of that array that contains the highest value. */  
public static int index(int [] arr, int size)  
{  
    int indexMax = 0;  
    for (int i=0; i<size; i++)  
    {  
        if (arr[i] > arr[indexMax])  
        {  
            indexMax = i;  
        }  
    }  
    return indexMax;  
} // END index
```

Testing

```
[14]: int[] numbers = {6,44,8,12,2,55,28,11,33,26}; //Finds the index of this array ↵  
      ↪which has the highest value.  
System.out.println(index(numbers, 10));
```

5

2.5.4 total

What it does This method gets the total sum of an integer array using a for loop to keep a running total by going through each value in the array. Then returns the total sum as a result of the method.

Implementation (how it works) Parameters are declared and will be used in the method. The parameters are the integer type array called arr and a size variable (integer type). Similar to method index, this is done to get the array and size through as an argument when calling this method. So in the first line in the method totalScore variable is declared and initialised which will be used as the start of the running total. Then a for loop is used which will go through the whole array by each value by using the size variable (counter control). Then for each repeat in the loop it will add on the value in the array of index i to the running totalScore and store it in the totalScore variable. This is repeated until the end of the for loop (end of the array size). Then totalScore will have to total sum of the array as its value. This value is returned as a result of the method.

```
[15]: // This method finds the sum of all the values in an array and returns that ↵  
      ↪value as a result of the method.  
public static int total(int [] arr, int size)  
{  
    int totalScore = 0;
```

```

    for (int i=0; i<size; i++)
    {
        totalScore = totalScore + arr[i];
    }
    return totalScore;
} // END total

```

Testing

```

[16]: int[] numbers2 = {6,44,8,12,2,55,28,11,33,26}; //Finds and prints the total sum
      ↪ of this array.
      System.out.println(total(numbers2, 10));

```

225

2.5.5 printDetails

What it does This method prints out the 3 arrays given through as arguments (when method is called) but only its first index of the 3 arrays in a line and then it prints out the next index of the 3 arrays in a line until the ending index of the 3 arrays.

Implementation (how it works) So similar to methods index and total the parameters are used in the code. This is done to get the 3 arrays called a, b and c needed to be printed and the variable size through as an argument when calling this method. So this method starts with a for loop which is looped until the value of size which is the size of the arrays. This is done so it would go through all values of the arrays. So in the for loop it prints out the arrays and their values in order of their index as a line. This is repeated for each of the indexes of the arrays until the last index of the arrays. So it will be printed out as a column of array a values and then a column of array c values and then a column of array b values. Each with their corresponding index values in rows.

```

[17]: /* This method prints out the array a index 0, array b index 0 and array c
      ↪ index 0 in a line. Then it prints
      the next index of those arrays in a line until the ending index of the arrays.
      ↪ */
      public static void printDetails(int size, String[] a, int[] b, String[] c)
      {
          for (int i=0; i<size; i++)
          {
              System.out.println(a[i] + ", " + c[i] + ", " + b[i]);
          }
      } // END printDetails

```

Testing

```

[19]: String[] names = {"Bob", "Jack", "Bill", "Joe", "Harry"}; // Prints arrays with
      ↪ corresponding index.
      int[] numbers3 = {6,44,8,12,2};

```

```
String[] colours = {"Black", "White", "Blue", "Green", "Purple"};
printDetails(5, names, numbers3, colours);
```

```
Bob, Black, 6
Jack, White, 44
Bill, Blue, 8
Joe, Green, 12
Harry, Purple, 2
```

2.5.6 player_details

What it does This method does the primary job. Where it asks and prints the question of naming 4 female footballers, how many goals they scored and what country they play for. Then prints out who is the highest goal scorer out of the 4 footballers and the total goals the footballers have scored altogether. Then it outputs the details of each of the footballers (their names, number of goals scored and what country they play for).

Implementation (how it works) So this method combines the other methods to do the primary function. First the players variable (which is declared as integer) is initialised by giving a value 4. This is a final variable so it stays constant as this is the number of women footballers we will give details for. Then we declare and initialise each array with the size of variable players for all. So the array player_names (String type) will hold the 4 female footballers names. The array goals (integer type) will hold the goals scored of each footballer. Then the player_country array (String type) will hold the countries the footballers play for. Then a for loop is used to ask the questions of the footballers name, number of goals scored and what country they played for and get the user's responses. This is done using inputString and inputInt methods. Then each of the details are held in the corresponding array. This is repeated 4 times (as counter ends at value of players which is the size of the arrays) for each footballer and their details so they are stored in each index of the array. Then we get the highest scoring footballer by getting the index of the highest value of array goals using method index and arguments goals and players. Then the index value that contains the highest value in array goals is stored in high_value. Then the highest scorer name is printed out by concatenating the array player_name value of high_value index. This will give the footballer's name of the highest scorer as the arrays are stored with their corresponding index of each player's details. Then to get the total sum of all the goals scored of all 4 footballers we call method total and give arguments goals and players. This will get a full total of the goals scored by the footballers. It is then stored in variable total_score (integer type). Then the total sum of goals scored by all footballers is printed through concatenation of variable total_score. Then the details of each footballer (which include name, goals scored and country they play for) are printed out by calling method printDetails and using arguments players (size of arrays), player_names (array holding footballers names), goals (array holding each of the footballers goals scored) and player_country (array holding the countries each footballer plays for). It is printed as a list with commas separating the name, country and goals scored for each player.

```
[20]: /* This method will ask the user to input 4 female footballers, number of
    ↳goals they scored and what country
    they play for. Then it will tell the user who is the highest scoring footballer
    ↳out of the 4 and the total
```

```

goals scored out of the 4 and then print out the details of the footballers:
↳ their names, their goals and
their country the play for. */
public static void player_details()
{
    final int players = 4;
    String [] player_names = new String[players];
    int [] goals = new int[players];
    String [] player_country = new String[players];
    for (int i=0; i<players; i++)
    {
        player_names [i] = inputString("Name footballer " + (i+1) + "?");
        goals [i] = inputInt("How many goals did they score?");
        player_country [i] = inputString("What country did they play for?");
    }
    int high_value = index(goals, players);
    System.out.println("The highest scorer was " + player_names[high_value] + ".
↳");
    int total_score = total(goals, players);
    System.out.println("These players scored " + total_score + " goals between
↳ them.");
    System.out.println("The details of the players are:");
    printDetails(players, player_names, goals, player_country);
} // END player_details

```

Testing

```
[21]: player_details();
```

```

Name footballer 1?
Ellen White
How many goals did they score?
6
What country did they play for?
England
Name footballer 2?
Megan Rapinoe
How many goals did they score?
7
What country did they play for?
USA
Name footballer 3?
Ajara Nchout
How many goals did they score?
2
What country did they play for?
Cameroon

```

Name footballer 4?
Cristiane
How many goals did they score?
4
What country did they play for?
Brazil
The highest scorer was Megan Rapinoe.
These players scored 19 goals between them.
The details of the players are:
Ellen White, England, 6
Megan Rapinoe, USA, 7
Ajara Nchout, Cameroon, 2
Cristiane, Brazil, 4

2.5.7 Running the program

Run the following call to simulate running the complete program.

```
[22]: player_details();
```

Name footballer 1?
Ellen White
How many goals did they score?
10
What country did they play for?
England
Name footballer 2?
Megan Rapinoe
How many goals did they score?
5
What country did they play for?
USA
Name footballer 3?
Ajara Nchout
How many goals did they score?
7
What country did they play for?
Cameroon
Name footballer 4?
Cristiane
How many goals did they score?
4
What country did they play for?
Brazil
The highest scorer was Ellen White.
These players scored 26 goals between them.
The details of the players are:
Ellen White, England, 10

Megan Rapinoe, USA, 5
Ajara Nchout, Cameroon, 7
Cristiane, Brazil, 4

2.6 The complete program

This version will only compile here. To run it copy it into a file called initials.java on your local computer and compile and run it there.

```
[24]: // NAME: Ifthy Fairoze
// DATE: 30/10/2020
// VERSION: 1
/* BRIEF OVERVIEW OF PURPOSE: Asks the user details of 4 footballers. Their
    ↳ name, how many goals they scored and
    ↳ what country they play for. Then it prints out the highest goal scorer out of
    ↳ the 4 footballers and the total
    ↳ goals they all have scored. Then it prints out the details of each footballer.
    ↳ */

import java.util.Scanner; // Needed to make Scanner available

class topGoalscorer
{
    public static void main (String [] a)
    {
        player_details();
        System.exit(0);
    }

    /* This method asks a String question to the user which then allows the
    ↳ user to type a integer
    ↳ response which is returned as a result of the method. */
    public static int inputInt(String message)
    {
        Scanner scanner = new Scanner(System.in);
        String textInput;
        int numberInput;
        System.out.println(message);
        textInput = scanner.nextLine();
        numberInput = Integer.parseInt(textInput);
        return numberInput;
    } // END inputInt

    /* This method prints out a question to the user and then allows the user
    ↳ to enter a
    ↳ response (which is an integer response) and returns that response as a
    ↳ result of this method. */
```

```

public static String inputString(String message)
{
    Scanner scanner = new Scanner(System.in);
    String answer;
    System.out.println(message);
    answer = scanner.nextLine();
    return answer;
} // END inputString

/* This method finds the index or position of the highest value in an
→integer array. Then it returns the
index number of that array that contains the highest value. */
public static int index(int [] arr, int size)
{
    int indexMax = 0;
    for (int i=0; i<size; i++)
    {
        if (arr[i] > arr[indexMax])
        {
            indexMax = i;
        }
    }
    return indexMax;
} // END index

// This method finds the sum of all the values in an array and returns that
→value as a result of the method.
public static int total(int [] arr, int size)
{
    int totalScore = 0;
    for (int i=0; i<size; i++)
    {
        totalScore = totalScore + arr[i];
    }
    return totalScore;
} // END total

/* This method prints out the array a index 0, array b index 0 and array c
→index 0 in a line. Then it prints
the next index of those arrays in a line until the ending index of the
→arrays. */
public static void printDetails(int size, String[] a, int[] b, String[] c)
{
    for (int i=0; i<size; i++)
    {
        System.out.println(a[i] + ", " + c[i] + ", " + b[i]);
    }
}

```

```

    } // END printDetails

    /* This method will asks the user to input 4 female footballers, number of
    ↪goals they scored and what country
    ↪they play for. Then it will tell the user who is the highest scoring
    ↪footballer out of the 4 and the total
    ↪goals scored out of the 4 and then print out the details of the footballers:
    ↪their names, their goals and
    ↪their country the play for. */
    public static void player_details()
    {
        final int players = 4;
        String [] player_names = new String[players];
        int [] goals = new int[players];
        String [] player_country = new String[players];
        for (int i=0; i<players; i++)
        {
            player_names [i] = inputString("Name footballer " + (i+1) + "?");
            goals [i] = inputInt("How many goals did they score?");
            player_country [i] = inputString("What country did they play for?");
        }
        int high_value = index(goals, players);
        System.out.println("The highest scorer was " + player_names[high_value]
    ↪+ ".");
        int total_score = total(goals, players);
        System.out.println("These players scored " + total_score + " goals
    ↪between them.");
        System.out.println("The details of the players are:");
        printDetails(players, player_names, goals, player_country);
    } // END player_details
}

```

END OF LITERATE DOCUMENT