AssessedExerciseTemplate-level7-Copy1

November 10, 2020

1 Short Assessed Exercise

2 Level 7

2.1 Ifthy Fairoze

2.2 5 November 2020

2.3 Version 1

2.4 Summary of the Question

Paralympic Swimming Relay: Paralympic swimming relay competitions involve competitors of different disability levels making up a team. For those with physical disabilities, each swimmer's disability is rated on a scale of 1-10 and the total points rating for the 4 swimmers must be no greater than a given number of points. For example, a legal relay team for a 34 point freestyle relay may consist of two S8 swimmers and two S9 swimmers (9 + 9 + 8 + 8 = 34), or an S10 swimmer and three S8 swimmers (10 + 8 + 8 + 8 = 34).

Write a program that gives information about relay teams given the country and the maximum points score for the event. It should check that the team is legal. A new type called ParaRelayTeam should be created (a record type) and each separate piece of information (country - a String, and points - 4 integers) about a team should be stored in fields of the record. This record must be defined in a new class containing NOTHING but the field definitions (no methods at all in the record class).

In the main class (the one containing the main method), an initialisation method should be provided to create records. It should take as arguments the values previously input about a team. Accessor methods must also be defined for the fields in the main class (they should NOT be in the ParaRelayTeam class for this exercise). These methods must be the only way the record is accessed.

The following is an example run of the program:

```
What is the classification (maximum points) of this relay event? 34
What country is the team representing? GB
What is the disability category of Swimmer 1? 10
What is the disability category of Swimmer 2? 8
What is the disability category of Swimmer 3? 8
What is the disability category of Swimmer 4? 8
That GB team has 34 points so is legal.
```

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, createTeam, getCountry, getTotal and teamDetails. For each method, I will explain about the method, what is 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 prints out a question to the user and allows the user totype an integer response to that question and it returns that response as a result of the method.

Implementation (how it works) It first sets up a local scanner to the method, linked to the keyboard which is done in the first line in the method. This makes the method scanner.nextLine() available. Then it prints out the String question to the user. Then it allows the user to type a response using nextLine and then read and stored in textInput (integer response stored as string at first). Then the value in textInput is converted from String to integer type and stored in numberInput variable as integer type. Then the value in numberInput is returned as a result of the method.

[7]:
```java
// This method prints out a question where the integer response entered by the user is returned as a result.
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

[3]:
```java
System.out.println(inputInt("What is the answer?"));
```

```
What is the answer?
34
34
```

2.5.2 inputString

What it does This method asks a question to the user and then waits for a String response from the user and returns that response as a result of the method.

Implementation (how it works) First it sets up a local scanner to the method, linked to the keyboard (similar to method inputInt). Then prints out the String question to the user. Then using scanner.nextLine (made avaliable from setting up local scanner in the first line) it allows the user to enter a String response and reads it and stores it into answer variable. Then the value in answer is returned as a result of the method.

[8]:

```
/* This method prints out a question to the user and allows the user to enter a␣
 ↪String response and that response
is returned as a result of the 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

[6]:

```
System.out.println(inputString("What is the answer?"));
```

```
What is the answer?
I don't know
I don't know
```

2.5.3 createTeam

What it does This method creates and intialises a record of new type ParaRelayTeam which contains the field details of the paralympic swimming relay team which are their country and each of the four disability rating.

Implementation (how it works) This is an intialisation method of creating a record of new ADT ParaRelayTeam. It first gives the parameters which will be given values through as arguments when calling this method. Each of the parameters are the country the relay team is from and each of the 4 swimmers disability ratings. First the new record is declared by creating new ParaRelayTeam value and stored in variable newTeam. Then each of the fields of the record are assigned with the corresponding parameters. So field country of record newTeam gets value of team_country and field points_1 of record newTeam gets value of level1 and so on. The record newTeam is now set and intialised. Then it returns the record stored in variable newTeam as a result of the method. A seperate class has to be made to use new type ParaRelayTeam with the required fields.

[2]:

```
class ParaRelayTeam
{
    String country;
    int points_1;
    int points_2;
    int points_3;
    int points_4;
}

/* This method creates and intialises a record by setting the fields of the␣
 ↪record, and then returns that record
as a result of the method. */
public static ParaRelayTeam createTeam(String team_country, int level1, int␣
 ↪level2, int level3, int level4)
{
    ParaRelayTeam newTeam = new ParaRelayTeam();
    newTeam.country = team_country;
    newTeam.points_1 = level1;
    newTeam.points_2 = level2;
    newTeam.points_3 = level3;
    newTeam.points_4 = level4;
    return newTeam;
} // END createTeam
```

Testing

[8]:

```
ParaRelayTeam Team = createTeam("GB", 8, 8, 10 ,8); // Creates new record Team.
```

2.5.4 getCountry

What it does This accessor method gets the country field from a record of ParaRelayTeam and returns it as a result of the method.

Implementation (how it works) This accessor method gets the country field of record Team1 of type ParaRelayTeam and returns that value in that field as a result of the method. Record is passed through as argument as it is needed to get the fields.

[4]:

```
// Given a record Team1 return the country part of that record.
public static String getCountry(ParaRelayTeam Team1)
{
    return Team1.country;
} // END getCountry
```

Testing

[5]:

```
ParaRelayTeam Team2 = createTeam("GB", 8, 8, 10 ,8);
System.out.println(getCountry(Team2));
```

```
GB
```

2.5.5 getTotal

What it does This accessor method accesses the points fields from the record and gets the total from them and returns the total value as a result of the method.

Implementation (how it works) The accessor method gets the points fields (points_1, points_2, points_3 and points_4) from record Team1 of type ParaRelayTeam and add them all up and stores the total in variable total (declared as integer type). Now total is intialised and the value in total is returned as a result of the method.

[11]:
```
// Given a record Team1 accesses the points fields and get total points and␣
 ↪returns the value of the total points.
public static int getTotal(ParaRelayTeam Team1)
{
    int total = Team1.points_1 + Team1.points_2 + Team1.points_3 + Team1.
 ↪points_4;
    return total;
} // END getTotal
```

Testing

[7]:
```
ParaRelayTeam Team3 = createTeam("GB", 8, 8, 10 ,8);
System.out.println(getTotal(Team3));
```

```
34
```

2.5.6 teamDetails

What it does This is method called in main method which asks the user what is the maximum points a relay team has and what country the relay team represents and the disability category of the 4 swimmers and then prints out if the team is legal or not.

Implementation (how it works) This method is the method that calls the other methods to do the main function of the whole program. First it asks the user the maximum points of the relay event and allows the user to give an integer response using method call inputInt and store the response into variable classification. Then asks the user what country a relay team represents and allows a String response from the user using inputString method call and stores the response in variable teamCountry. Then it asks the user what is the disability category for each of the 4 swimmers seperately and gets a response from the user using inputInt method call. Then stores each of the categories of the swimmer (from the response) to the corresponding variables. So the first swimmer category is stored in variable category_1 and so on. Then the record of the relay team details is created by calling out the initialisation method with corresponding variables of teamCountry, category_1, category_2, category_3 and category_4 as arguments which creates and initilaises the record. Then an if statement is used to print out if the team is legal or not depending on their total points. So the test asks if the total points value (gotten from calling accessor method getTotal) is less than or equal to value of classification (value of maximum legal points). If this test is true it will print out that the team is legal using concatenation of accessor methods to get total points and the country the team represent. Otherwise if the test is false it will print out that the team is not legal using concatenation of accessor methods to get total points and the country the the team represents.

[5]:

```java
/* This method is the method called out in the main method where it asks the info of the team to the user and
asks to input the answers and then prints out if that relay team is legal or not using
primitive method calls. */
public static void teamDetails()
{
    int classification = inputInt("What is the classification (maximum points) of this relay event?");
    String teamCountry = inputString("What country is the team representing?");
    int category_1 = inputInt("What is the disability category of Swimmer 1?");
    int category_2 = inputInt("What is the disability category of Swimmer 2?");
    int category_3 = inputInt("What is the disability category of Swimmer 3?");
    int category_4 = inputInt("What is the disability category of Swimmer 4?");
    ParaRelayTeam RelayTeam = createTeam(teamCountry, category_1, category_2, category_3, category_4);
    if (getTotal(RelayTeam) <= classification)
    {
        System.out.println("That " + getCountry(RelayTeam) + " team has " + getTotal(RelayTeam) + " points so is legal.");
    }
    else
    {
        System.out.println("That " + getCountry(RelayTeam) + " team has " + getTotal(RelayTeam) + " points so is not legal.");
    }
    return;
} // END teamDetails
```

Testing

[9]:

```java
teamDetails();
```

```
What is the classification (maximum points) of this relay event?
34
What country is the team representing?
GB
What is the disability category of Swimmer 1?
10
What is the disability category of Swimmer 2?
8
What is the disability category of Swimmer 3?
8
What is the disability category of Swimmer 4?
8
That GB team has 34 points so is legal.
```

2.5.7 Running the program

Run the following call to simulate running the complete program.

[10]:
```
teamDetails();
```

```
What is the classification (maximum points) of this relay event?
34
What country is the team representing?
GB
What is the disability category of Swimmer 1?
10
What is the disability category of Swimmer 2?
10
What is the disability category of Swimmer 3?
8
What is the disability category of Swimmer 4?
8
That GB team has 36 points so is not legal.
```

2.6 The complete program

This version will only compile here.

[12]:
```
// NAME: Ifthy Fairoze
// DATE: 05/11/2020
// VERSION: 1
/* BRIEF OVERVIEW OF PURPOSE: This program asks the user the detils of the␣
 ↪Paralympic Relay Team max legal points
and the details of the Paralympic Relay Team counrty, and each of the 4␣
 ↪swimmers disability rating and then tells
the user if that Paralympic Relay Team is legal or not. */

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

class ParaRelayTeam // New class needed to create new type ParaRelayTeam to␣
 ↪create and access records of this type.
{
    String country;
    int points_1;
    int points_2;
    int points_3;
    int points_4;
}

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

    // This method prints out a question where the integer response entered by␣
↪the user is returned as a result.
    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 allows the user to␣
↪enter a String response
    and that response is returned as a result of the 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 creates and intialises a record by setting the fields of the␣
↪record,
    and then returns that record as a result of the method. */
    public static ParaRelayTeam createTeam(String team_country, int level1, int␣
↪level2, int level3, int level4)
    {
        ParaRelayTeam newTeam = new ParaRelayTeam();
        newTeam.country = team_country;
        newTeam.points_1 = level1;
        newTeam.points_2 = level2;
        newTeam.points_3 = level3;
        newTeam.points_4 = level4;
        return newTeam;
    } // END createTeam

    // Given a record Team1 return the country part of that record.
    public static String getCountry(ParaRelayTeam Team1)
    {
        return Team1.country;
    } // END getCountry

    // Given a record Team1 accesses the points fields and get total points and␣
↪returns the value of the total points.
    public static int getTotal(ParaRelayTeam Team1)
    {
        int total = Team1.points_1 + Team1.points_2 + Team1.points_3 + Team1.
↪points_4;
        return total;
    } // END getTotal

    /* This method is the method called out in the main method where it asks␣
↪the info of the team to the user and
    asks to input the answers and then prints out if that relay team is legal␣
↪or not using
    primitive method calls. */
    public static void teamDetails()
    {
        int classification = inputInt("What is the classification (maximum␣
↪points) of this relay event?");
        String teamCountry = inputString("What country is the team representing?
↪");
        int category_1 = inputInt("What is the disability category of Swimmer 1?
↪");
        int category_2 = inputInt("What is the disability category of Swimmer 2?
↪");
        int category_3 = inputInt("What is the disability category of Swimmer 3?
↪");
        int category_4 = inputInt("What is the disability category of Swimmer 4?
↪");
        ParaRelayTeam RelayTeam = createTeam(teamCountry, category_1,␣
↪category_2, category_3, category_4);
        if (getTotal(RelayTeam) <= classification)
        {
            System.out.println("That " + getCountry(RelayTeam) + " team has " +␣
↪getTotal(RelayTeam) + " points so is legal.");
        }
        else
        {
            System.out.println("That " + getCountry(RelayTeam) + " team has " +␣
↪getTotal(RelayTeam) + " points so is not legal.");
        }
        return;
    } // END teamDetails
```

```

}
```

END OF LITERATE DOCUMENT