Mohamed-Azouz-Salama-A200134277mini7

December 8, 2020

0.1 Mini Project

0.2 Level 6,7

0.3 Mostafa Mohamed Azouz Salama A

0.4 6 December 2020

0.5 Summary of Question

develop a program that asks the user a question and asks them to answer with the least common answer from the choices.

0.6 Class comp

0.7 What it does

This is a record that contains 5 values the user answer, String question, and 3 Strings for the 3 possible answers. it makes it possible to store all of these 3 values into one record of type comp

0.8 Implementation

you first must define a variable of type comp and then naming that varaible to anything you want. For assigning a value invisually to each of the int value inside the record, the name of the variable is written followed by . and then the varaible inside the record you want to assign the value to eg; comp.question

[2]:
```
class comp
{
    String user_answer;
    String question;
    String answer1;
    String answer2;
    String answer3;
}
```

1 Testing

[3]:
```
comp test= new comp();
test.answer1="anything";
test.user_answer="test";
System.out.println(test.answer1);
System.out.println(test.user_answer);
```

```
anything
test
```

2 class Questionbank

3 what it does

it includes an array of class comp and a int score and another int of how mnay questions the user wants to be asked

4 Implemantation

you first must define a variable of type comp and then naming that varaible to anything you want. For assigning a value invisually to each of the int value inside the record, the name of the variable is written followed by . and then the varaible inside the record you want to assign the value to eg; Questionbank.score

[4]:
```
public class Questionbank
{
    comp[] info= new comp[10];
    int score=0;
    int users=0;
}
```

5 Testing

[3]:
```
Questionbank test1= new Questionbank();
test1.score=1;
test1.users=2;
System.out.println(test1.score);
System.out.println(test1.users);
```

```
1
2
```

There is a file output in the main method which allows the admin(creator of the quiz) to enter all the questions and answers for the user to answer. It calls the ask method and scans the answer in a string and uses outputStream.println to store the answer that is stored in the string toput it in

one line of the file public static void anything() PrintWriter outputStream= new PrintWriter(new FileWriter("C:

Users

mosta

Desktop

Desktop

Mostafa Azouz

Queen mary

eclipse

mini6

src

quizinfo.txt")); String temp; for(int counter=1;counter<=4;counter++) outputStream.println(temp=ask("what is question "+counter)); outputStream.println(temp=ask("Enter the highest scoring answer for question "+ counter)); outputStream.println(temp=ask("Enter the second highest scoring answer for question "+ counter)); outputStream.println(temp= ask("Enter the lowest scoring answer for question "+ counter)); outputStream.close(); return;

5.1 ask

5.2 What it does?

it calls the method System.out.println to print out a message for the user to read. Then strores the users answer in a variable and returns it. it takes an argument of string and returns string as well

5.3 Implementation

it declare a scanner of type Scanner to scan the user's answer to it then prints out the desired string message

[3]:
```
public static String ask (String message)
{
    Scanner scanner= new Scanner (System.in);
        String text_input;
        System.out.println(message);
        text_input= scanner.nextLine();
        return text_input;
}
```

6 Testing

[9]:
```
ask("whats your name");
```

```
whats your name
hgf
```

[9]: hgf

7 CreateQandA

8 What it does

this basically is the accesor method that sets the question and answers for each section of the array # Implementation it runs a for loop for the array to store each line of the file input in one space of the array of type comp then closes the file and returns the Questionbank

[5]:
```
public static Questionbank CreateQandA (Questionbank b) throws IOException
        {

                BufferedReader inputStream = new BufferedReader (new␣
↪FileReader("C:\\Users\\mosta\\Desktop\\Desktop\\Mostafa Azouz\\Queen␣
↪mary\\eclipse\\mini6\\src\\quizinfo.txt"));
               for(int counter=1;counter<=4;counter++)
               {
                       b.info[counter].question=inputStream.readLine();
                   b.info[counter].answer1= inputStream.readLine();
                   b.info[counter].answer2=inputStream.readLine();
                   b.info[counter].answer3=inputStream.readLine();
               }
                inputStream.close();

               return b;
        }
```

9 Testing

[8]:
```
Questionbank temp= new Questionbank();
temp.info[0]= new comp();
temp.info[1]= new comp();
temp.info[2]= new comp();
temp.info[3]= new comp();
temp.info[4]= new comp();
temp.info[5]= new comp();
CreateQandA(temp);
```

```
↪----------------------------------------------------------------------------

       java.io.FileNotFoundException: C:\Users\mosta\Desktop\Desktop\Mostafa␣
↪Azouz\Queen mary\eclipse\mini6\src\quizinfo.txt (No such file or directory)

               at java.base/java.io.FileInputStream.open0(Native Method)

               at java.base/java.io.FileInputStream.open(FileInputStream.java:
↪219)

               at java.base/java.io.FileInputStream.<init>(FileInputStream.java:
↪157)

               at java.base/java.io.FileInputStream.<init>(FileInputStream.java:
↪112)

               at java.base/java.io.FileReader.<init>(FileReader.java:60)

               at .CreateQandA(#12:4)

               at .(#35:1)
```

10 Createquiz

11 What it does

it takes the user's input from the previous method and calculates the user's score based on his answer's # Implementation It runs a for loop to ask 4 questoins . it then, based on the user's answers, calculates the score and stores it in the score int in the Question bank class. it takes an argument Questionbank and returns an int

[5]:
```
public static int Createquiz(Questionbank d)
        {

                for(int counter=1; counter<=4; counter++)
                {
                    d.info[counter].user_answer=ask(d.info[counter].question);
                    // compares the user's answer to the answers stored and␣
 ↪gives a score
                    if(d.info[counter].user_answer.equals(d.info[counter].
 ↪answer1))
                    {
                        d.score=d.score+50;
                    }
```

```
                    else if(d.info[counter].user_answer.equals(d.
↪info[counter].answer2))
                {
                    d.score=d.score+35;
                }
                else if(d.info[counter].user_answer.equals(d.info[counter].
↪answer3))
                {
                    d.score=d.score+15;
                }
                else
                {
                    System.out.println("your answer is wrong you gain 100␣
↪points");
                    d.score= d.score+100;
                }
                System.out.println(d.score);

              }
                return d.score;
}
```

12 testing

[6]:
```
test1.score=Createquiz(test1);
```

```
|   public static int Createquiz(Questionbank d)
|            {
|
|                    for(int counter=1; counter<=4; counter++)
|                    {
|                        d.info[counter].user_answer=ask(d.
↪info[counter].question);
|                        // compares the user's answer to the answers␣
↪stored and gives a score
|                        if(d.info[counter].user_answer.equals(d.
↪info[counter].answer1))
```

```
        |                         {
        |                             d.score=d.score+50;
        |                         }
        |                          else if(d.info[counter].user_answer.equals(d.
↪info[counter].answer2))
        |                     {
        |                         d.score=d.score+35;
        |                     }
        |                     else if(d.info[counter].user_answer.equals(d.
↪info[counter].answer3))
        |                     {
        |                         d.score=d.score+15;
        |                     }
        |                     else
        |                     {
        |                         System.out.println("your answer is wrong you␣
↪gain 100 points");
        |                         d.score= d.score+100;
        |                     }
        |                     System.out.println(d.score);
        |
        |                   }
        |                     return d.score;
        |   }
        Unresolved dependencies:
```

```
- method ask(java.lang.String)
```

13 viewanswers

14 What it does

it views the correct answers for each question the user has been aksed after the quiz is over and also the final score # Implementation it runs a for loop to print out the answers for each of the questions by accessing the info array of type comp, using System.out.println. after the loop it only prints out the final score once.

[20]:
```
public static void viewanswers (Questionbank a)
{
    for(int counter=1;counter<=4;counter++)
    {
        System.out.println("The correct answers for question "+counter+ " were␣
↪" +a.info[counter].answer1+","+a.info[counter].answer2+", and "+a.
↪info[counter].answer3);
    }
    System.out.println("Your final score was " +a.score);
    return;
}
```

15 Testing

[21]:
```
viewanswers(test1);
```

```
The correct answers for question 0 were n,u, and g
The correct answers for question 1 were nk,ko, and ml
Your final score was 201
```

16 sortnum

17 what it does

it sorts the scores of the users from highest to lowest in the form of an array and prints them out # Implementation It takes an argument of an array that contains the scores of the user's who played this quiz and the number of users who played the quiz. it runs a for loop with the amount of users that played the quiz. it runs another for loop inside the bigger one to compare the selected value of the array to all the ones before it. it the value is greater than the one before it the the two values switch and the value of the counter decreases so that it points to the original value. else nothing happens. then at the ned the counter resets to run the bigger for loop

[7]:
```
// sorts the numbers from lowest to highest to create the leaderboard
public static void sortnum (int[]arr, int limit)
```

```
{
        int temp=0;
        int increment=0;
        int return_counter;
        for(int counter=0; counter<limit;counter++)
        {

                if (counter==0)
                {
                        arr[0]=arr[0];
                }
        else
{
        // stores the value of the counter in a temp value to restore the␣
 ↪original value of the counter later
        return_counter=counter;
         increment= counter-1;
        // runs a loop to check the value of the  position of the array in␣
 ↪question to each value before it
                for(int x=increment ;x>=0;x--)
        {

        // checks if the value in position counter is less that position x
                if (arr[counter]<=arr[x])
// if it is, it switches the variables (as in the smaller moves one pos ahead␣
 ↪and the bigger one pos backwards)
{

        temp=arr[counter];
        arr[counter]=arr[x];
        arr[x]=temp;
// then decrements the counter by one to point to the value of the array that␣
 ↪was switched
        counter--;
}
        else
        {
                        arr[counter]=arr[counter];

        }

                }
// returns the counter to its original value to continue the big first for␣
 ↪loop
                counter=return_counter;
}
```

```
}


        System.out.println(Arrays.toString(arr));
        return;
}
```

18 testing

[ ]:

19 Show

20 What it does

it calls all of the above method to have the complete quiz and it also has a loop to make it possible for the user to complete the quiz again # Implementation it declares quiz od type question bank and then declares an array of int to store each user's score it then asks how many users will play and creates a fir loop based on the answer. at the end of the loop it stores the quiz.score in a section of the int array for score. it then asks wether the user wants to view the correct answers or not

[8]:
```
public static void show() throws IOException
{
         System.out.println("user mode ");
        Questionbank quiz= new Questionbank();
        quiz.info[0]= new comp();
        quiz.info[1]= new comp();
        quiz.info[2]= new comp();
        quiz.info[3]= new comp();
        quiz.info[4]= new comp();
        quiz.info[5]= new comp();

        String temp=ask("How many users will play this quiz");
        quiz.users= Integer.parseInt(temp);

        int[]user_scores= new int[quiz.users];
        quiz=CreateQandA(quiz);

        for(int counter=0;counter<quiz.users;counter++)
        {

                quiz.score=0;

                quiz.score=Createquiz(quiz);
```

```
                String ans =ask("Would you like to view the correct answers for␣
↪the questions ?");
                if(ans.equals("yes"))
                {
                        viewanswers(quiz);
                }
                else
                {
                        System.out.println("your final score was "+quiz.score);
                }
        user_scores[counter]=quiz.score;

        }
        System.out.println("The final score leaderboards was (first left and␣
↪last right)");
        sortnum(user_scores,quiz.users);

        return;
}
```

21 Testing

[9]:
```
show();
```

```
|   public static void show() throws IOException

|   {

|            System.out.println("user mode ");

|           Questionbank quiz= new Questionbank();

|           quiz.info[0]= new comp();

|           quiz.info[1]= new comp();

|           quiz.info[2]= new comp();

|           quiz.info[3]= new comp();

|           quiz.info[4]= new comp();

|           quiz.info[5]= new comp();

|
```

```
        |           String temp=ask("How many users will play this quiz");
        |           quiz.users= Integer.parseInt(temp);
        |
        |           int[]user_scores= new int[quiz.users];
        |           quiz=CreateQandA(quiz);
        |
        |           for(int counter=0;counter<quiz.users;counter++)
        |           {
        |
        |                   quiz.score=0;
        |
        |                   quiz.score=Createquiz(quiz);
        |                   String ans =ask("Would you like to view the correct␣
↪answers for the questions ?");
        |                   if(ans.equals("yes"))
        |                   {
        |                           viewanswers(quiz);
        |                   }
        |                   else
        |                   {
        |                           System.out.println("your final score was␣
↪"+quiz.score);
        |                   }
        |           user_scores[counter]=quiz.score;
```

```
        |
        |           }
        |           System.out.println("The final score leaderboards was (first␣
↪left and last right)");
        |           sortnum(user_scores,quiz.users);
        |
        |           return;
        |   }
        Unresolved dependencies:
           - method ask(java.lang.String)
           - method CreateQandA(Questionbank)
           - method viewanswers(Questionbank)
```

22 The Complete Program

For the complete program we - add overarching comments - the scanner library needs to be explicitly imported to allow the input methods to be used - the starting call is put in a vanilla main method - the whole set of methods is enclosed in a## The Complete Program class definition - the order of methods is rearranged to make it more readable as a top-down program

[ ]:

[1]:
```
// author: Mostafa Mohamed Azouz Salama
// date: 17/10/2020
//version 1
import java.util.Arrays;
import java.util.Scanner;
import java.io.*;

class comp
{

    String answer1;
    String answer2;
    String answer3;
    String question;
```

```
    String user_answer;

}
// the ADT
 class Questionbank
{
        comp[] info= new comp[10];
        int score=0;
        int users=0;
}
public class gameshow {
         public static void main (String []args) throws IOException
         {
                 System.out.println("Admin mode");
                 PrintWriter outputStream= new PrintWriter(new FileWriter("C:
↪\\\\Users\\\\mosta\\\\Desktop\\\\Desktop\\\\Mostafa Azouz\\\\Queen␣
↪mary\\\\eclipse\\\\mini6\\\\src\\\\quizinfo.txt"));
                String temp;
                 for(int counter=1;counter<=4;counter++)
                 {
                         outputStream.println( temp=ask("what is question␣
↪"+counter));
                         outputStream.println(temp=ask("Enter the highest␣
↪scoring answer for question  "+ counter));
                         outputStream.println(temp=ask("Enter the second␣
↪highest scoring answer for question  "+ counter));
                         outputStream.println(temp= ask("Enter the lowest␣
↪scoring answer for question  "+ counter));
                 }
                 outputStream.close();
                         show();
                 System.exit(0);
         }


         // asks a question and scans the user's answer and returns it
         public static String ask (String message)
         {
             Scanner scanner= new Scanner (System.in);
                 String text_input;
                 System.out.println(message);
                 text_input= scanner.nextLine();
                 return text_input;

 }

         // the accesor method that creates the questions and their answers
```

```
        public static Questionbank CreateQandA (Questionbank b) throws␣
↪IOException
        {

                BufferedReader inputStream = new BufferedReader (new␣
↪FileReader("C:\\Users\\mosta\\Desktop\\Desktop\\Mostafa Azouz\\Queen␣
↪mary\\eclipse\\mini6\\src\\quizinfo.txt"));
               for(int counter=1;counter<=4;counter++)
               {
                       b.info[counter].question=inputStream.readLine();
                   b.info[counter].answer1= inputStream.readLine();
                   b.info[counter].answer2=inputStream.readLine();
                   b.info[counter].answer3=inputStream.readLine();
               }
                inputStream.close();

               return b;
        }
        // this method runs the main quiz
        public static int Createquiz(Questionbank d)
        {

                 for(int counter=1; counter<=4; counter++)
                 {
                     d.info[counter].user_answer=ask(d.info[counter].question);
                     // compares the user's answer to the answers stored and␣
↪gives a score
                     if(d.info[counter].user_answer.equals(d.info[counter].
↪answer1))
                     {
                         d.score=d.score+50;
                     }
                      else if(d.info[counter].user_answer.equals(d.
↪info[counter].answer2))
                 {
                     d.score=d.score+35;
                 }
                 else if(d.info[counter].user_answer.equals(d.info[counter].
↪answer3))
                 {
                     d.score=d.score+15;
                 }
                 else
                 {
                     System.out.println("your answer is wrong you gain 100␣
↪points");
```

```
                    d.score= d.score+100;
                }
                System.out.println(d.score);

            }
                return d.score;


         }
         // views the correct answers for each question in the quiz
public static void viewanswers (Questionbank a)
{
        for(int counter=1;counter<=4;counter++)
        {
                System.out.println("The correct answers for question␣
 ↪"+counter+" were " +a.info[counter].answer1+","+a.info[counter].answer2+",␣
 ↪and "+a.info[counter].answer3);
        }
        System.out.println("Your final score was " +a.score);
        return;
}
 // sorts the numbers from lowest to highest to create the leaderboard
public static void sortnum (int[]arr, int limit)
{
        int temp=0;
        int increment=0;
        int return_counter;
        for(int counter=0; counter<limit;counter++)
        {

                if (counter==0)
                {
                        arr[0]=arr[0];
                }
        else
{
        // stores the value of the counter in a temp value to restore the␣
 ↪original value of the counter later
        return_counter=counter;
         increment= counter-1;
        // runs a loop to check the value of the  position of the array in␣
 ↪question to each value before it
                for(int x=increment ;x>=0;x--)
        {

        // checks if the value in position counter is less that position x
                if (arr[counter]<=arr[x])
```

```
// if it is, it switches the variables (as in the smaller moves one pos ahead␣
 ↪and the bigger one pos backwards)
{

        temp=arr[counter];
        arr[counter]=arr[x];
        arr[x]=temp;
// then decrements the counter by one to point to the value of the array that␣
 ↪was switched
        counter--;
}
        else
        {
                        arr[counter]=arr[counter];

        }

                }
// returns the counter to its original value to continue the big first for␣
 ↪loop
                counter=return_counter;
}

}


         System.out.println(Arrays.toString(arr));
         return;
}

public static void show() throws IOException
{
         System.out.println("user mode ");
        Questionbank quiz= new Questionbank();
        quiz.info[0]= new comp();
        quiz.info[1]= new comp();
        quiz.info[2]= new comp();
        quiz.info[3]= new comp();
        quiz.info[4]= new comp();
        quiz.info[5]= new comp();

        String temp=ask("How many users will play this quiz");
        quiz.users= Integer.parseInt(temp);

        int[]user_scores= new int[quiz.users];
        quiz=CreateQandA(quiz);
```

```
        for(int counter=0;counter<quiz.users;counter++)
        {

                quiz.score=0;

                quiz.score=Createquiz(quiz);
                String ans =ask("Would you like to view the correct answers for␣
 ↪the questions ?");
                if(ans.equals("yes"))
                {
                        viewanswers(quiz);
                }
                else
                {
                        System.out.println("your final score was "+quiz.score);
                }
        user_scores[counter]=quiz.score;

        }
        System.out.println("The final score leaderboards was (first left and␣
 ↪last right)");
        sortnum(user_scores,quiz.users);

        return;
}
 }
```

[ ]: