

EcologicalFootprint.java:

-----

import java.util.\*;

public class EcologicalFootprint

{

    //declared static to be used across all methods

    //scanner variable to gather user input

    private static Scanner input;

    private static ArrayList<Profile> profiles;

    private static String prompt, stringInput;

    private static int intInput, min, max;

    //food variables

    private static int meatQuantityScore, meatQualityScore;

    //housing variables

    private static int houseTypeScore, houseMaterialScore, householdSizeScore,  
houseSizeScore;

    //consumption variables

    private static int electricityScore, wasteGenerationScore;

    //transportation variables

    private static int travelDistanceScore, vehicleEfficiencyScore, carpoolScore,  
publicTransportationScore, flightTimeScore;

    public static void main(String[] args) {

        EcologicalFootprint footprintCalculator = new EcologicalFootprint();

        footprintCalculator.runQuiz();

    }

    public EcologicalFootprint()

    {

        this.input = new Scanner(System.in);

        this.profiles = new ArrayList<Profile>();

    }

    public static void runQuiz()

    {

        while (true)

        {

            System.out.println("WELCOME TO THE ECOLOGICAL FOOTPRINT QUIZ!\n\n");

            //calling methods to get users name

            String name = getProfileName();

            String occupation = getProfileOccupation();

            int age = getProfileAge();

            //printing all methods to simulate quiz

            titlePage(); //title

            clearScreen();

            meatQuantityScore = meatConsumption(); //question 1

            clearScreen();

            meatQualityScore = meatQuality(); //question 2

            clearScreen();

            houseTypeScore = housingType(); //question 3

            clearScreen();

            houseMaterialScore = housingMaterial(); //question 4

            clearScreen();

            householdSizeScore = householdSize(); //question 5

            clearScreen();

            houseSizeScore = houseSize(); //question 6

            clearScreen();

            electricityScore = electricityInfo(); //question 7, 8, 9

            clearScreen();

            wasteGenerationScore = wasteGeneration(); //question 10

            clearScreen();

            travelDistanceScore = travelDistance(); //question 11

            clearScreen();

```

        vehicleEfficiencyScore = vehicleEfficiency(); //question 12
        clearScreen();
        carpoolScore = carpool(); //question 13
        clearScreen();
        publicTransportationScore = publicTransportation(); //question 14
        clearScreen();
        flightTimeScore = flightTime(); //question 15
        clearScreen();

        //calculates users final score to attach to their profile
        double earthCount = calculateEarthCount(calculateFinalScore());

        //creating an object for the user's quiz
        Profile profile = new Profile(name, occupation, age, earthCount);
        profiles.add(profile); //adding to ArrayList of Profiles

        //printing the results
        boolean quit = printResults();

        if (quit) { // checks if the user wants to exit the quiz
            break;
        }
        clearScreen();
    }
    displayLeaderboard();
}

//obatains the user's name
public static String getProfileName()
{
    //loop to disallow integers for a name
    System.out.print("Enter your Full Name: ");
    while (true)
    {
        if(!input.hasNextInt()) { //rejecting integer user input
            stringInput = input.nextLine().trim();
            if (!stringInput.isEmpty())
                break;
            else {
                input.nextLine();
                clearScreen();
                System.out.println("WELCOME TO THE ECOLOGICAL FOOTPRINT QUIZ!\n\n");
                System.out.println("***Invalid Input. Please only enter letters.***\n");
                System.out.print("Enter your Full Name: ");
            }
        } else { // moves onto next input to not store previous attempt
            input.nextLine();
            clearScreen();
            System.out.println("WELCOME TO THE ECOLOGICAL FOOTPRINT QUIZ!\n\n");
            System.out.println("***Invalid Input. Please only enter letters.***\n");
            System.out.print("Enter your Full Name: ");
        }
    }
    return stringInput; // returning users name
}

//obtains the users occupation
//acts the same as getProfileName()
public static String getProfileOccupation()
{
    //loop to disallow integers for an occupation
    System.out.print("Enter your Occupation: ");
    while (true)
    {
        if(!input.hasNextInt()) { //rejecting integer user input
            stringInput = input.nextLine().trim();

```

```

        if (!stringInput.isEmpty())
            break;
        else {
            input.nextLine();
            System.out.println("***Invalid Input. Please only enter letters.***\n");
            System.out.print("Enter your Occupation: ");
        }
    } else { // moves onto next input to not store previous attempt
        input.nextLine();
        System.out.println("***Invalid Input. Please only enter letters.***\n");
        System.out.print("Enter your Occupation: ");
    }
}

return stringInput; // returning users name
}

//obtains the users age
public static int getProfileAge()
{
    //ensures the user does not enter an unreasonable age or a String
    System.out.print("Enter your age in years: ");
    while (true)
    {
        if (input.hasNextInt()) // must enter an int to continue
        {
            intInput = input.nextInt();
            input.nextLine();
            if (intInput > 0 && intInput <= 115) //valid age check
                break;
            else {
                System.out.println("***Invalid Input. Please enter a valid age (1-115).***\n");
                System.out.print("Enter your age in years: ");
            }
        } else {
            input.nextLine();
            System.out.println("***Invalid Input. Please only enter numbers.***\n");
            System.out.print("Enter your age in years: ");
        }
    }
    return intInput;
}

//method to create a title page
public static void titlePage()
{
    //printing instructions
    clearScreen();
    System.out.println("PAGE 1/17\n\n\n\n\n\n\n\n\n\t\t\t what is your\n\t\t ECOLOGICAL FOOTPRINT?\n\n\t\t How many Earths would we need to live like you?\n\n\n\n\n\n\n\n\n\t\t Press 'ENTER' to take the First Step\n\n\n");
    System.out.print("Your Input ('ENTER'): ");

    //loop to obtain the correct user input
    while (true) {
        stringInput = input.nextLine().trim();
        //checks to see if 'enter' key was pressed
        if (stringInput.isEmpty()) {
            break;
        }
        //if not, resets screen for visual purposes
        clearScreen();
        System.out.println("PAGE 1/17\n\n\n\n\n\n\n\n\n\t\t\t what is your\n\t\t ECOLOGICAL FOOTPRINT?\n\n\t\t How many Earths would we need to live like you?\n\n\n\n\n\n\n\n\n\t\t Press 'ENTER' to take the First Step\n\n\n");
        System.out.println("***Invalid input. Please press the 'ENTER' key.***\n");
    }
}

```

```

        System.out.print("Your Input ('ENTER'): ");
    }
}

public static int meatConsumption()
{
    //prompt to pass as parameter for printing
    prompt = "PAGE 2/17\n\n\n\n\n\n\n\t\t\t\t\t food\n\n\t\t\t\t\t HOW OFTEN DO YOU EAT ANIMAL\n\n\t\t\t\t\t BASED PRODUCTS?\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t 0\t\t\t\t\t 25\t\t\t\t\t 50\t\t\t\t\t 75\n\n\t\t\t\t\t 100\n\n\t\t\t\t\t Never---Infrequently---Occasionally---Often---Very Often\n\n\n\t\t\t\t\t ANSWER FROM 0-100\n\n\n";
    min = 0;
    max = 100;

    //assigning quiz output to variable
    int meatQtyScore = intInputChecker(prompt, min, max);

    //returning variable for future final score
    return meatQtyScore;
}

public static int meatQuality()
{
    prompt = "PAGE 3/17\n\n\n\n\n\n\n\n\t\t\t\t\t food\n\n\t\t\t\t\t HOW MUCH OF THE FOOD\n\n\t\t\t\t\t THAT YOU EAT IS UNPROCESSED,\n\n\t\t\t\t\t UNPACKAGED OR LOCALLY GROWN?\n\n\n\t\t\t\t\t (less than 320\n\n\t\t\t\t\t kilometers/200 miles away)\n\n\n\n\n\n\n\n\t\t\t\t\t 0\t\t\t\t\t 25\t\t\t\t\t 50\t\t\t\t\t 75\n\n\t\t\t\t\t 100\n\n\t\t\t\t\t None-----All\n\n\n\t\t\t\t\t ANSWER FROM 0-100(%) \n\n\n";
    min = 0;
    max = 100;

    int meatQtyScore = intInputChecker(prompt, min, max);
    /*this inverses the score based on user input
    this is because more of these specific actions is actually beneficial
    making the calculator accurate
    */
    meatQtyScore = (-1 * meatQtyScore) + 100;
    return meatQtyScore;
}

public static int housingType()
{
    prompt = "PAGE 4/17\n\n\n\n\n\n\n\n\t\t\t\t\t housing\n\n\t\t\t\t\t WHICH HOUSING TYPE\n\n\t\t\t\t\t BEST DESCRIBES YOUR HOME?";
    String opt1 = "Freestanding, no running water", opt2 = "Freestanding, running water", opt3 = "Multi-storey apartment", opt4 = "Duplex/row house", opt5 = "Luxury condominium";

    int houseTypeScore = houseChecker(prompt, opt1, opt2, opt3, opt4, opt5);

    return houseTypeScore;
}

public static int housingMaterial()
{
    prompt = "PAGE 5/17\n\n\n\n\n\n\n\n\t\t\t\t\t housing\n\n\t\t\t\t\t WHAT MATERIAL IS YOUR\n\n\t\t\t\t\t HOUSE CONSTRUCTED WITH?";
    String opt1 = "Straw/bamboo", opt2 = "Adobe", opt3 = "Wood", opt4 = "Brick/concrete", opt5 = "Steel/other";

    int houseMaterialScore = houseChecker(prompt, opt1, opt2, opt3, opt4, opt5);

    return houseMaterialScore;
}

public static int householdSize()

```

```
{
    prompt = "PAGE 6/17\n\n\n\n\n\n\n\t\t\t housing\n\n\t HOW MANY PEOPLE LIVE IN YOUR HOUSEHOLD?\n\n\n\n\n\n\n\n\n\n\n Just me-----\n\n\n ANSWER FROM 1-11 \n\n";
    min = 1;
    max = 11;

    int householdSizeScore = (intInputChecker(prompt, min, max) - 1 ) * 10;

    return householdSizeScore;
}

public static int houseSize()
{
    prompt = "PAGE 7/17\n\n\n\n\n\n\n\t\t\t housing\n\n\t\tWHAT IS THE SIZE OF YOUR HOME?\n\n\n\n\n\n\n\n\n\n 50          3750          7500          11150          15000\n\n Tiny-----Huge\n\n ANSWER FROM 50-15000(sq ft)\n\n";
    min = 50;
    max = 15000;

    int houseSizeScore = (int)((double)(intInputChecker(prompt, min, max) - 50) / 149.5);
    return houseSizeScore;
}

public static int electricityInfo()
{
    System.out.println("PAGE 8/17\n\n\n\n\n\n\n\t\t\t housing\n\n\t DO YOU HAVE ELECTRICITY IN YOUR HOME?\n\n\n\n\n\n\n\n\n\n\n\t\t\t 'No' or 'Yes'\n\n\n");
    System.out.print("Your Input (yes/no): ");

    while (true) {
        stringInput = input.nextLine().trim().toLowerCase();
        //checks to see if 'enter' key was pressed
        if (stringInput.equals("no")) {
            break;
        } else if (stringInput.equals("yes")){
            return (energyEfficiency() + energyRenewal());
        }
        //if not, resets screen for visual purposes
        clearScreen();
        System.out.println("PAGE 8\n\n\n\n\n\n\n\n\t\t\t housing\n\n\t DO YOU HAVE ELECTRICITY IN YOUR HOME?\n\n\n\n\n\n\n\n\n\n\n\t\t\t 'No' or 'Yes'\n\n\n");
        System.out.println("***Invalid input. Please enter 'No' or 'Yes'.***\n");
        System.out.print("Your Input ('No'/'Yes'): ");
    }
    return 0;
}

public static int energyEfficiency()
{
    prompt = "PAGE 9/17\n\n\n\n\n\n\n\n\t\t\t housing\n\n\t HOW ENERGY EFFICIENT IS YOUR HOME?\n\n\n\n\n\n\n\n\n\n 0          25          50          75\n100\n\n Hardly--Below Average--Average--Energy Efficient--Fully\n\n ANSWER FROM 0-100(%) \n\n";
    min = 0;
    max = 100;

    clearScreen();

    int energyEfficiencyScore = intInputChecker(prompt, min, max);
    energyEfficiencyScore = (-1 * energyEfficiencyScore) + 100;

    return energyEfficiencyScore;
}
```

```

public static int energyRenewal()
{
    prompt = "PAGE 10/17\n\\n\\n\\n\\n\\n\\n\\n\\t\t\t housing\n\\n\\tWHAT PERCENTAGE OF YOUR  
HOME'S ELECTRICITY\n\\t COMES FROM RENEWABLE SOURCES??\n\\n\\n\\n\\n\\n\\n 0  
25          50          75          100\n\\n Low-----  
-----High\n\\n ANSWER FROM 0-100(%) \n\n";
    min = 0;
    max = 100;

    clearScreen();

    int energyRenewalScore = intInputChecker(prompt, min, max);
    energyRenewalScore = (-1 * energyRenewalScore) + 100;

    return energyRenewalScore;
}

public static int wasteGeneration()
{
    prompt = "PAGE 11/17\n\\n\\n\\n\\n\\n\\n\\n\\t\t\t housing\n\\n\\tCOMPARED TO YOU NEIGHBORS,  
HOW MUCH TRASH\n\\t\t DO YOU GENERATE??? \n\\n\\n\\n\\n\\n\\n 0          25  
50          75          100\n\\n Much Less-----Less-----Same-----More-----Much More\n\\n  
ANSWER FROM 0-100 \n\n";
    min = 0;
    max = 100;

    int wasteGenerationScore = intInputChecker(prompt, min, max);

    return wasteGenerationScore;
}

public static int travelDistance()
{
    prompt = "PAGE 12/17\n\\n\\n\\n\\n\\n\\n\\n\\t\t\t transportation\n\\n\\tHOW FAR DO YOU TRAVEL  
BY VEHICLE EACH WEEK?\n\\n\\n\\n\\n\\n\\n\\n\\n\\n 0          125          250          375  
500\n\\n Zero-----Extremely Far\n\\n ANSWER FROM 0-  
500(miles) \n\n";
    min = 0;
    max = 500;

    int travelDistanceScore = (intInputChecker(prompt, min, max)/5);

    return travelDistanceScore;
}

public static int vehicleEfficiency()
{
    prompt = "PAGE 13/17\n\\n\\n\\n\\n\\n\\n\\n\\t\t\t transportation\n\\n\\tHOW EFFICIENT IS THE  
VEHICLE YOU USE MOST?\n\\n\\n\\n\\n\\n\\n\\n\\n\\n 10          45          80          115  
150\n\\n Inefficient-----Very Efficient/Electric\n\\n ANSWER FROM 10-  
150(miles/gallon) \n\n";
    min = 10;
    max = 150;

    int vehicleEfficiencyScore = (int)((double)(intInputChecker(prompt, min, max) -  
10)/1.4);
    vehicleEfficiencyScore = (-1 * vehicleEfficiencyScore) + 100;

    return vehicleEfficiencyScore;
}

public static int carpool()
{

```

```

        prompt = "PAGE 14/17\n\\n\\n\\n\\n\\n\\n\\n\\n\\t\\t\\tttransportation\\n\\n\\t\\t    HOW OFTEN DO YOU  
CARPOOL?\\n\\n\\n\\n\\n\\n\\n\\n\\n\\n\\n      0          25          50          75          100\\n\\n  
Never-----Always\\n\\n    ANSWER FROM 0-100(%) \\n\\n";  
  
        min = 0;  
        max = 100;  
  
        int carpool = intInputChecker(prompt, min, max);  
        carpool = (-1 * carpool) + 100;  
  
        return carpool;  
    }  
  
    public static int publicTransportation()  
    {  
        prompt = "PAGE 15/17\n\\n\\n\\n\\n\\n\\n\\n\\n\\t\\t\\tttransportation\\n\\n    HOW FAR DO YOU TRAVEL  
BY PUBLIC TRANSPORTATION EACH WEEK?\\n\\n\\n\\n\\n\\n\\n\\n\\n\\n\\n      0          125          250  
375          500\\n\\n    Not Far-----Very Far\\n\\n  
ANSWER FROM 0-500(miles) \\n\\n";  
        min = 0;  
        max = 500;  
  
        int publicTransportationScore = (intInputChecker(prompt, min, max)/5);  
  
        return publicTransportationScore;  
    }  
  
    public static int flightTime()  
    {  
        prompt = "PAGE 16/17\n\\n\\n\\n\\n\\n\\n\\n\\n\\t\\t\\tttransportation\\n\\n\\t    HOW MANY HOURS DO  
YOU FLY EACH YEAR?\\n\\n\\n\\n\\n\\n\\n\\n\\n\\n\\n      0          50          100          150  
200\\n\\n    None-----Many\\n\\n    ANSWER FROM 0-  
200 \\n\\n";  
        min = 0;  
        max = 200;  
  
        int flightTimeScore = (intInputChecker(prompt, min, max)/2);  
  
        return flightTimeScore;  
    }  
  
    //prints out the final result  
    public static boolean printResults()  
    {  
        //calls other methods to calculate these scores  
        double finalScore = calculateFinalScore(), earthCount =  
calculateEarthCount(finalScore);  
        boolean retakeQuiz;  
        String message;  
  
        if (earthCount <= 1)  
            message = "Hurray we would be able to live on this Earth alone!";  
        else  
            message = "Whoops, looks like we would need more Earths to live on!\n Try  
        System.out.print("PAGE 17/17\n\\n\\n\\n\\n\\n\\n\\n\\n\\t\\t    Here Are Your Results!\n\\n\\n  
AMOUNT OF EARTHS NEEDED FOR EVERY PERSON TO LIVE LIKE YOU:\\n\\n\\n\\n\\n\\t\\t    |Earth's  
Required: " + earthCount + "|\\n\\n\\n    " + message + "\\n\\n\\n\\n\\n\\n\\n\\n\\n    Press 'ENTER'  
to take the First Step again or 'Q' to Quit\n\\n\\nYour Input ('Enter' key or 'Q'): ");  
  
        retakeQuiz = quitQuiz();  
  
        return retakeQuiz;  
    }  
  
    //calculates the persons final score
```

```

public static double calculateFinalScore()
{
    double finalScore = ((double)(meatQuantityScore + meatQualityScore +
houseTypeScore + houseMaterialScore + householdSizeScore + houseSizeScore +
electricityScore + wasteGenerationScore + travelDistanceScore + vehicleEfficiencyScore +
carpoolScore + publicTransportationScore + flightTimeScore)/14);

    return finalScore;
}

//converts the final score into earth count
public static double calculateEarthCount(double finalScore)
{
    //an exponential equation that ranges from .2 to 27.2
    // no true meaning behind it
    //starts at .2 because no score should result in 0 Earth's to live on
    double earthCount = (0.0021 * Math.pow(finalScore, 2)) + (0.09 * finalScore) +
0.2;
    //Rounding to two decimal places for better look/space
    return Math.round(earthCount * 100.0) / 100.0;
}

//method to quit the quiz and verify user input
public static boolean quitQuiz()
{
    boolean cont;
    while (true) {
        stringInput = input.nextLine().trim().toLowerCase();
        //checks to see if 'enter' key or "q" were pressed
        if (stringInput.isEmpty()) { // if user wants to retry the quiz
            cont = false;
            return cont;
        } else if (stringInput.equals("q")) { //If user wants to quit the quiz
            cont = true;
            return cont;
        }
        //if not, resets screen for visual purposes and prints screen again
        clearScreen();
        printResults();
    }
}

//creates a leaderboard
public static void displayLeaderboard()
{
    clearScreen();
    //size of the profiles ArrayList
    int leaderboardSize = profiles.size();
    //using the bubble sort algorithm to sort the profiles
    //in descending order based on earthCount
    for (int i = 0; i < leaderboardSize - 1; i++)
    {
        for (int j = 0; j < leaderboardSize - i - 1; j++)
        {
            //Compare profiles based on earthCount
            if(profiles.get(j).getEarthCount() > profiles.get(j + 1).getEarthCount())
            {
                //swap profiles[j] and profiles[j+1]
                Profile temp = profiles.get(j);
                profiles.set(j, profiles.get(j + 1));
                profiles.set(j + 1, temp);
            }
        }
    }

    System.out.println("***FINAL RESULTS***\n\n\n\nLeaderboard:");
}

```



```

        int rank = 1;
        //for-each loop to print each profile
        for (Profile profile : profiles)
        {
            if (rank == 1) {
                System.out.println("\t\t\t~~~ YOU ARE THE MOST ECO-FRIENDLY, CONGRATS!!!
~~~");
                System.out.println("\t#" + rank + " " + profile +
"\n
\n");
            } else {
                System.out.println("\t#" + rank + " " + profile);
            }
            rank++;
        }
    }

    /*** METHODS TO SIMPLIFY THE CODE ***/
    *
    *
    *
    */
    //method to check if user entered a valid String input
    public static int houseChecker(String prompt, String opt1, String opt2, String opt3,
String opt4, String opt5)
    {
        System.out.println(prompt);
        System.out.println("\n\n\t\t\t1." + opt1 + "\n\n\t\t\t2." + opt2 + "\n\n\t\t\t3." +
opt3 + "\n\n\t\t\t4." + opt4 + "\n\n\t\t\t5." + opt5 + "\n\n");
        System.out.print("Your Input(1-5): ");

        while(true) { //infinite loop until valid input is entered
            if (input.hasNextInt()) { //verifies next input will be an int
                intInput = input.nextInt();
                input.nextLine();
                if (intInput >= 1 && intInput <= 5) { //verifies intInput is within
requested scope
                    return (intInput - 1) * 25;
                } else { //else invalid input enter, allows user to try again
                    clearScreen();
                    System.out.println(prompt);
                    System.out.println("\n\n\t\t\t1." + opt1 + "\n\n\t\t\t2." + opt2 +
"\n\n\t\t\t3." + opt3 + "\n\n\t\t\t4." + opt4 + "\n\n\t\t\t5." + opt5 + "\n\n");
                    System.out.println("***Invalid input. Please enter a number between 1
and 5.***\n");
                    System.out.print("Your Input (1-5): ");
                }
            } else { //if user input is not an Int, tells user to retry
                input.next();
                clearScreen();
                System.out.println(prompt);
                System.out.println("\n\n\t\t\t1." + opt1 + "\n\n\t\t\t2." + opt2 +
"\n\n\t\t\t3." + opt3 + "\n\n\t\t\t4." + opt4 + "\n\n\t\t\t5." + opt5 + "\n\n");
                System.out.println("***Invalid input. Please enter a number between 1 and
5.***\n");
                System.out.print("Your Input (1-5): ");
            }
        }
    }

    //method to check the user has entered a valid input
    public static int intInputChecker(String prompt, int min, int max)
    {
        System.out.println(prompt);
        System.out.print("Your Input (" + min + "-" + max + "): ");
    }

```

```

//infinite loop until valid input is entered
while (true) {
    if (input.hasNextInt()) { //verifies next input will be an int
        intInput = input.nextInt();
        input.nextLine();
        if (intInput >= min && intInput <= max) { //verifies intInput is within
requested scope
            break;
        } else { //else invalid input enter, allows user to try again
            clearScreen();
            System.out.println(prompt);
            System.out.println("***Invalid input. Please enter a number between "
+ min + " and " + max + ".***\n");
            System.out.print("Your Input (" + min + "-" + max + "): ");
        }
    } else { //if user input is not an Int, tells user to retry
        input.next();
        clearScreen();
        System.out.println(prompt);
        System.out.println("***Invalid input. Please enter a number between " +
min + " and " + max + ".***\n");
        System.out.print("Your Input (" + min + "-" + max + "): ");
    }
}
return intInput; //user's answer to the prompt
}

// method that clears the console
public static void clearScreen() //researched to be used
{
    System.out.print("\033[H\033[2J"); //ANSI escape code to clear console
    System.out.flush(); //ensures that any buffered data in the stream is immediately
written out
}
}

```

Profile.java:

-----

```

public class Profile
{
    private String name;
    private String occupation;
    private int age;
    private double earthCount;

    public Profile(String name, String occupation, int age, double earthCount)
    {
        this.name = name;
        this.occupation = occupation;
        this.age = age;
        this.earthCount = earthCount;
    }

    public String getName()
    {
        return name;
    }

    public String getOccupation()
    {
        return occupation;
    }

    public int getAge()
    {

```

```
        return age;
    }

    public double getEarthCount()
    {
        return earthCount;
    }

    @Override
    public String toString()
    {
        return name + " is a " + occupation + " at the age of " + age +
            " and has an Earth Count of: " + earthCount;
    }
}
```