THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED.

Programming Exercise 03 – Pokémon Training

Authors: Daniel, Alex, Ruchi, Ricky
Topics: Math, Random, Scanner, enums

Problem Description

This assignment will assess your knowledge of the Math, Random, and Scanner classes, and the creation of an enum type.

You've been training your Pokémon for days, and you decide that you want to practice your attacks at a Pokémon Trainer Gym. Your Pokémon has three possible attacks: scratch, surf, and tackle, each of which does a random amount of damage in each interval. On each turn, you can choose only one attack to carry out. The battle will continue until your rival's Pokémon faints. When you defeat your rival, you will get to walk away with some prize money!

Notes:

1. When generating a random number, you **must** use `java.util.Random`.
2. Only create ONE Scanner object (when prompted to in the main method) and reuse it whenever you need to read more input. **Creating more than one Scanner may result in a zero for the assignment!**
3. Floating-point values in strings should be *displayed* to two decimal places with rounding.
4. Interval notation is used throughout this document.
   a. $[a, b]$ means a value x such that $a \leq x \leq b$.
   b. $(a, b]$ means a value x such that $a < x \leq b$.
   c. $[a, b)$ means a value x such that $a \leq x < b$.
   d. (a, b) means a value x such that $a < x < b$.
5. **Every command-line prompt must occur on a separate line**, with the **user input typed on the same line as the prompt**. For example, asking for you and your rival's Pokémon's nickname should appear as follows:

```
Enter your Pokemon's nickname: Bob
Enter your rival's Pokemon's nickname: Karen
```

Solution Description

You will need to complete and turn in 2 files: `AttackType.java` and `PokemonBattle.java.`

AttackType.java

This is a public enum that represents the three possible attacks your Pokémon can perform. Populate this enum with the values `SCRATCH`, `SURF`, and `TACKLE` (in this order).

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED.

PokemonBattle.java

1. Create a class named `PokemonBattle`.
2. Create the `main` method inside the `PokemonBattle` class. The code for all following steps should be written within the `main` method.
3. Declare a variable named `rand` and initialize it to an instance of the `Random` class.
4. Declare a variable named `scan` and initialize it to an instance of the `Scanner` class that will read user input from the console.
5. Declare a `String` for your Pokémon's nickname. This will be initialized in step 8.
6. Declare a `String` for your rival's Pokémon's nickname. This will also be initialized in step 8.
7. Declare a **double** variable for your rival's Pokémon's health and initialize it to a randomly generated **integer** in the range $[40, 60)$.
8. Prompt for your and your rival's Pokémon's nicknames:
   a. Print the following to the console (without the quotation marks):

      ```
      "Enter your Pokemon's nickname: "
      ```

   b. Read in the input as a String and assign it to the appropriate variable. Note that names may contain spaces in them. Additionally, you should use String's `trim` method to remove any leading and trailing whitespace.
   c. Print the following to the console (without the quotation marks):

      ```
      "Enter your rival's Pokemon's nickname: "
      ```

   d. Read in the input as a String and assign it to the appropriate variable. Note that names may contain spaces in them. Additionally, you should use String's `trim()` method to remove any leading and trailing whitespace.
9. Print the following on its own line (single line, without quotation marks and curly braces):

   ```
   "Your rival has chosen {rival's Pokemon's nickname} to fight,
   which has {rival's Pokemon's health} health."
   ```

10. Create a **do-while** loop that terminates when your rival Pokémon faints (i.e., $\text{health} \leq 0$). **Do NOT use `break` to exit the loop**. Steps 10a – 10e should be performed inside this loop.
    a. Randomly choose the attack: create a local `AttackType` variable (i.e., within the loop) named `attack` and assign a random `AttackType` to it.
       i. You can do so using the following line of code (`rand` is a `Random` object):

          ```
          AttackType attack = AttackType.values()[rand.nextInt(3)];
          ```

    b. Each attack will inflict a different amount of damage to your opponent (i.e., rival's Pokémon's health decreases). You should use the conditional statement ideal for this scenario, where there are a small finite number of possible values for a variable.
       i. If the attack is `SCRATCH`:
          1. Generate a random **integer** in the range of `[1, 3]`. This is the number of times your Pokémon will scratch during that turn.
          2. Generate a random **double** in the range `[1.0, 6.0)`. This is how much damage you will inflict **per scratch** in that turn.
          3. Use these values to determine the **total damage** inflicted during that turn.

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED.

ii. If the attack is `SURF`, you will inflict random damage as a **double** in the range `[2.0, 11.0)`.
iii. If the attack is `TACKLE`, you will inflict random damage as a **double** in the range `[7.0, 9.0)`.
iv. Don't forget to subtract the damage you inflict from your rival's health.

c. After your turn is complete, print the following on its own line (single line, without quotation marks and curly braces):

```
"{Your Pokemon's nickname} used {attack} and did {damage}
damage. Your rival has {rival's Pokemon's health} health
remaining."
```

Note: When printing your rival's remaining health, the value displayed should not be less than 0. You can use a ternary expression or the `Math.max` method to make this easy!

d. Keep track of the total number of turns taken (i.e., the total number of attacks).
e. If your rival's Pokémon fainted, the loop should terminate. **Do NOT use** `break` **to accomplish this**. The loop's continuation condition should control this.

11. The battle goes on until your rival's Pokémon faints. After exiting the loop, print the following on its own line (single line, without quotation marks and curly braces):

```
"{Your rival's Pokemon's nickname} fainted after {total turns}
turns!"
```

12. Store a random **double** in the range `(1200.0, 2400.0]` as the prize money from the battle. **Hint:** Think about how multiplying by -1 affects the inclusivity/exclusivity of the boundaries.
13. Use `printf` to display the following message on its own line (without the curly braces):

```
You have earned ${prize money}!
```

Turn-In Procedure

Submission

To submit, upload the files listed below to the corresponding assignment on Gradescope:

- `AttackType.java`
- `PokemonBattle.java`

Make sure you see the message stating the assignment was submitted successfully. From this point, Gradescope will run a basic autograder on your submission as discussed in the next section. **Any autograder tests are provided as a courtesy to help "sanity check" your work and you may not see all the test cases used to grade your work.** You are responsible for thoroughly testing your submission on your own to ensure you have fulfilled the requirements of this assignment. If you have questions about the requirements given, reach out to a TA or Professor via the class forum for clarification.

You can submit as many times as you want before the deadline, so feel free to resubmit as you make substantial progress on the assignment. We will only grade your latest submission. **Be sure to submit every file each time you resubmit**.

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED.

Gradescope Autograder

If an autograder is enabled for this assignment, you may be able to see the results of a few basic test cases on your code. Typically, tests will correspond to a rubric item, and the score returned represents the performance of your code on those rubric items only. If you fail a test, you can look at the output to determine what went wrong and resubmit once you have fixed the issue. **We reserve the right to hide any or all test cases, so you should make sure to test your code thoroughly against the assignment's requirements.**

The Gradescope tests serve two main purposes:

- Prevent upload mistakes (e.g., non-compiling code)
- Provide basic formatting and usage validation

In other words, the test cases on Gradescope are by no means comprehensive. Be sure to thoroughly test your code by considering edge cases and writing your own test files. You also should avoid using Gradescope to compile, run, or Checkstyle your code; you can do that locally on your machine.

Other portions of your assignment can also be graded by a TA once the submission deadline has passed, so the output on Gradescope may not necessarily reflect your grade for the assignment.

Burden of Testing

You are responsible for thoroughly testing your submission against the written requirements to ensure you have fulfilled the requirements of this assignment.

Be **very careful** to note the way in which text output is formatted and spelled. Minor discrepancies could result in failed autograder cases.

If you have questions about the requirements given, reach out to a TA or Professor via the class forum for clarification.

Allowed Imports

- `java.util.Random`
- `java.util.Scanner`

Feature Restrictions

There are a few features and methods in Java that overly simplify the concepts we are trying to teach or break our autograder. For that reason, do not use any of the following in your final submission:

- `var` (the reserved keyword)
- `System.exit`
- `System.arraycopy`

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED.

Collaboration

Only discussion of the assignment at a conceptual high level is allowed. You can discuss course concepts and assignments broadly; that is, at a conceptual level to increase your understanding. If you find yourself dropping to a level where specific Java code is being discussed, that is going too far. Those discussions should be reserved for the instructor and TAs. To be clear, you should never exchange code related to an assignment with anyone other than the instructor and TAs.

You **MAY NOT** use code generation tools to complete this assignment. This includes generative AI tools such as ChatGPT.

Important Notes (Don't Skip)

- Non-compiling files will receive a 0 for all associated rubric items.
- Do not submit `.class` files.
- Test your code in addition to the basic checks on Gradescope.
- Submit every file each time you resubmit.
- Read the "Allowed Imports" and "Restricted Features" to avoid losing points.
- **Check on Ed Discussion for a note containing all official clarifications and sample outputs.**

It is expected that everyone will follow the Student-Faculty Expectations document and the Student Code of Conduct. The professor expects a **positive, respectful, and engaged academic environment** inside the classroom, outside the classroom, in all electronic communications, on all file submissions, and on any document submitted throughout the duration of the course. **No inappropriate language is to be used, and any assignment deemed by the professor to contain inappropriate offensive language or threats will get a zero.** You are to use professionalism in your work. Violations of this conduct policy will be turned over to the Office of Student Integrity for misconduct.

©CS1331

THIS GRADED ASSESSMENT IS NOT FOR DISTRIBUTION.
ANY DUPLICATION OUTSIDE OF GEORGIA TECH'S LMS IS UNAUTHORIZED