Table of Contents

Introduction

This challenge assignment is optional and meant to replace certain assignment scores if you think that the usual assignment is below your level.

Summary

For this game, there will be a player HP (Hit points) and enemy HP. They both start at 100.

The game will continue looping until either the player HP or the enemy HP reaches 0.

Each time through the loop:

- Display the player HP and enemy HP
- Ask the player if they want to Attack, Defend, or Heal.
- Attack: Generate a random attack value, subtract this amount from the Enemy's HP
- Defend: Generate a random defence value. If the enemy attacks, subtract the damage done by this defence value.
- Heal: Generate a random heal amount. Add this amount to the player's HP.
- Then, have the enemy randomly choose between Attack, Defend, and Heal. Have the same actions execute (but vice versa) for the enemy choices.
- Note that HP should not go above 100, and damage should not be negative.
- After both player and enemy have chosen their turn, execute their commands (attack, defend, or heal) and display the result (e.g., "Player hits enemy for 5 damage!")
- Once one of their HP has reached 0, stop looping, and display who is the winner.

Details

Create a class named **RPGBattleProgram**. You will add methods within to create the game.

Member Variables

This program will have the following **private member variables**, which can be used throughout the program:

- playerHP, integer
- enemyHP, integer
- rand, a **Random** object (java.util.Random)

setup() method

This method will initialize the game and get it ready to go. This method might be called after the game is over, if the player wants to play another round.

Initialize playerHP and enemyHP to 100 (to represent 100%).

main() method

This is where the main game loop will be stored.

1. Create a **boolean** variable called **done** and initialize it to **false. done** will be true once playerHP or enemyHP reaches 0.
2. While **done** is false, loop:

    <u>PART 1 CHOOSE ACTION</u>

    1. Display the **playerHP** and **enemyHP** to the user.
    2. Display 3 options to the player: 1. Attack, 2. Defend, 3. Heal.
    3. Ask the user to enter their choice. Store it in an integer variable called **playerChoice**.
    4. Randomly generate the enemy's choice. Store it in an integer variable called **enemyChoice**.

PART 2 CALCULATE RESULTS

5. Display the player's choice and enemy's choice...
   "You ATTACK!", "Enemy HEALS!"
6. Create new variables:
   1. **playerAtk** and **enemyAtk**, initialize these to 0.
   2. **playerHeal** and **enemyHeal**, initialize these to 0.
   3. **playerDef** and **enemyDef**, initialize these to 0.
7. If the player chose to attack, set **playerAtk** to a random value between 3 and 6.
8. If the player chose to defend, set **playerDef** to a random value between 1 and 3.
9. If the player chose to heal, set **playerHeal** to a random value between 5 and 10.
10. (Do the same as above, but for the enemy actions.)

PART 3 ADJUST STATS

11. Once everything has been calculated, you will adjust the player and enemy HP.
12. Healing: **playerHP = playerHP + playerHeal**.
13. Damage: **playerHP = playerHP – enemyAtk + playerDef**
    You'll be attacked by the enemy, but the damage will be nullified by the defense count.
14. ...And vice versa for the enemy.
15. Make sure that nobody's HP is above 100. If their HP goes above 100, just set it back to 100.

PART 4 CHECK HP

16. If the **playerHP** or **enemyHP** is 0 or below, then display a message: "You win!" or "You lose!". Set the **done** variable to **true** so that the loop will end and the program will close.

Once the loop has finished, display the message "Good bye!". Once the end of the method is reached, the program will end.