**OOP PROJECT**

**BY RAHUL THAMBI A059, BTECH IT**

**BRIEF DESCRIPTION:**

The purpose of the code is to implement a simple text-based Pokémon battle game. The game consists of three classes: Health, Pokemon, and Battle. The Health class defines the current and maximum health points of a Pokémon and checks if the Pokémon is fainted or not. The Pokemon class defines the attributes of a Pokémon such as name, type, and health points. The Battle class is where the actual battle between the player's Pokémon and an enemy Pokémon takes place. It includes methods to calculate the damage inflicted by each Pokémon and to check if a Pokémon has fainted. It also has a start() method which runs the game loop and prompts the player to choose an action for their Pokémon. The main() method is in the PokemonGame class and creates instances of the player's Pokémon and an enemy Pokémon. It then creates a Battle object and starts the battle.

**CODE:**

package javaproject;

import java.util.Scanner;

import java.util.Random;

class Health

{

private int maxHealth;

private int currentHealth;

public Health(int maxHealth)

{

this.maxHealth = maxHealth;

this.currentHealth = maxHealth;

}

public int getMaxHealth()

{

return maxHealth;

}

public int getCurrentHealth()

{

return currentHealth;

}

public void setCurrentHealth(int currentHealth)

{

this.currentHealth = currentHealth;

}

public boolean isFainted()

{

return currentHealth <= 0;

}

}

class Pokemon

{

private String name;

private String type;

private Health health;

public Pokemon(String name, String type, int maxHealth)

{

this.name = name;

this.type = type;

this.health = new Health(maxHealth);

}

public String getName()

{

return name;

}

public String getType()

{

return type;

}

public Health getHealth()

{

return health;

}

}

class Battle

{

private Pokemon playerPokemon;

private Pokemon enemyPokemon;

private Random random;

public Battle(Pokemon playerPokemon, Pokemon enemyPokemon)

{

this.playerPokemon = playerPokemon;

this.enemyPokemon = enemyPokemon;

this.random = new Random();

}

private int calculateDamage(Pokemon attacker, Pokemon defender)

{

int attackPower = 10; // Set a default attack power

// Check if the attacker's type is strong against the defender's type

if (attacker.getType().equals("Fire") && defender.getType().equals("Grass"))

{

attackPower \*= 2;

}

else if (attacker.getType().equals("Water") && defender.getType().equals("Fire"))

{

attackPower \*= 2;

}

else if (attacker.getType().equals("Grass") && defender.getType().equals("Water"))

{

attackPower \*= 2;

}

else

{

attackPower=10;

}

// Calculate a random damage value based on the attack power and the defender's health

int damage = random.nextInt(attackPower)+1; //if we dont write - then

// damage = defender.getHealth().getCurrentHealth() - damage;

// Make sure the damage is at least 1

if (damage < 5)

{

System.out.println("not very effective");

}

else

{

System.out.println("Super effective");

}

return damage;

}

public void start()

{

Scanner input=new Scanner(System.in);

System.out.println("A wild " + enemyPokemon.getName() + " appeared!");

// Loop until one of the Pokemon faints

while (!playerPokemon.getHealth().isFainted() && !enemyPokemon.getHealth().isFainted())

{

System.out.println(playerPokemon.getName() + " (HP: " + playerPokemon.getHealth().getCurrentHealth() + ")");

System.out.println(enemyPokemon.getName() + " (HP: " + enemyPokemon.getHealth().getCurrentHealth() + ")");

// Ask the player to choose an action

System.out.println("Choose an action:");

System.out.println("1. Attack");

System.out.println("2. Run");

int choice = input.nextInt();

if (choice == 1)

{

// Player attacks

int damage = calculateDamage(playerPokemon, enemyPokemon);

System.out.println(playerPokemon.getName() + " attacks for " + damage + " damage!");

enemyPokemon.getHealth().setCurrentHealth(enemyPokemon.getHealth().getCurrentHealth() - damage);

if (enemyPokemon.getHealth().isFainted())

{

System.out.println(enemyPokemon.getName() + " fainted!");

break;

}

// Enemy attacks

damage = calculateDamage(enemyPokemon, playerPokemon);

System.out.println(enemyPokemon.getName() + " attacks for " + damage + " damage!");

playerPokemon.getHealth().setCurrentHealth(playerPokemon.getHealth().getCurrentHealth() - damage);

if (playerPokemon.getHealth().isFainted())

{

System.out.println(playerPokemon.getName() + " fainted!");

break;

}

}

else if (choice == 2)

{

// Player attempts to run away

if (random.nextInt(2) == 0) //that is chooses either 0/1

{

System.out.println("You ran away!");

break;

}

else

{

System.out.println("You couldn't escape!");

// Enemy attacks

int damage = calculateDamage(enemyPokemon, playerPokemon);

System.out.println(enemyPokemon.getName() + " attacks for " + damage + " damage!");

playerPokemon.getHealth().setCurrentHealth(playerPokemon.getHealth().getCurrentHealth() - damage);

if (playerPokemon.getHealth().isFainted())

{

System.out.println(playerPokemon.getName() + " fainted!");

break;

}

}

}

}

}

}

public class PokemonGame

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

// Create player's Pokemon

System.out.println("Create your Pokemon!");

System.out.print("Enter Pokemon name: ");

String playerName = scanner.nextLine();

System.out.print("Enter Pokemon type: ");

String playerType = scanner.nextLine();

System.out.print("Enter Pokemon max health: ");

int playerMaxHealth = scanner.nextInt();

Pokemon playerPokemon = new Pokemon(playerName, playerType, playerMaxHealth);

// Create enemy Pokemon

Pokemon enemyPokemon = new Pokemon("charmander", "Fire", 100);

// Start battle

Battle battle = new Battle(playerPokemon, enemyPokemon);

battle.start();

}

}

**OUTPUT:**

**Case where the player was able to run away!**

**Text, letter

Description automatically generated**

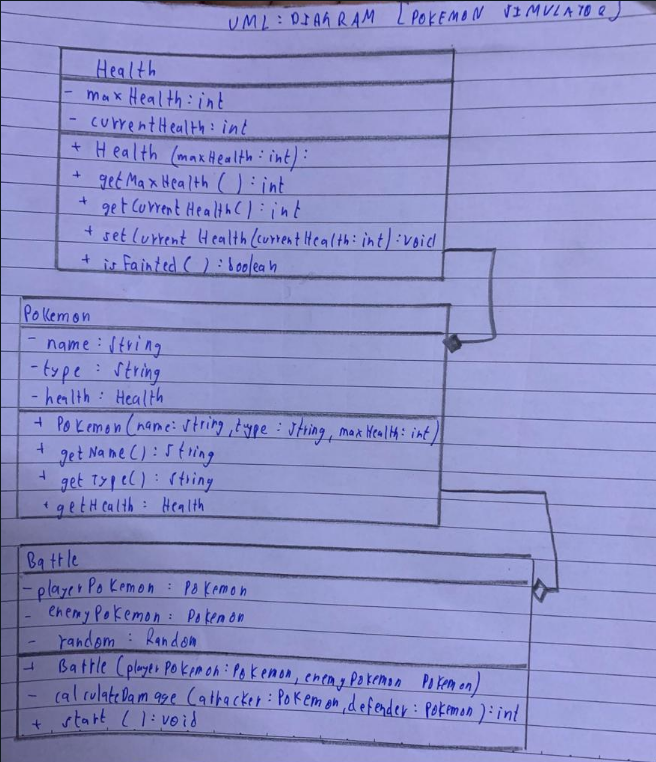
**Case where the player was unable to run away!**

**Text

Description automatically generated Text

Description automatically generated**

**UML DIAGRAM:**

****