**POKEMON GAME**

**Core Classes:**

* **Pokémon:**
* **Attributes:** Name, type, health, attack, defence, special attack, special defence, speed, level, experience, moves (list of Move objects), is wild.
* **Methods:**
* Attack (opponent): Calculates damage based on types and stats.
* Take Damage (damage): Reduces health and handles potential faints.
* Gain experience (exp): Increases experience and checks for level up.
* Level up (): Increases stats and potentially learns new moves.
* Learn move (move): Adds a new move to the Pokémon’s repertoire.
* Get Stat (stat): Accesses a specific stat value.
* **Move:**
* **Attributes:** Name, type, power, accuracy, category (physical, special), effect (optional).
* **Methods:**
* Apply effect (target): Executes the move's effect (if any).
* **Trainer:**
* **Attributes:** Name, team (list of Pokémon objects), inventory (list of Item objects).
* **Methods:**
* Catch Pokémon (wild Pokémon): Attempts to catch a wild Pokémon with a Pokeball.
* Use item (item, target): Applies the item's effect to a chosen Pokémon.
* Battle (opponent): Engages in a turn-based battle with another trainer.
* **Item:**
* **Attributes:** Name, type, effect.
* **Methods:**
* Use (target): Applies the item's effect to a chosen Pokémon.
* **Inheritance:**
* **Pokémon subclasses:** Similar to Java, create subclasses for different types with unique stats and moves (Fire Pokémon, Water Pokémon, etc.).
* **Move subclasses:** Consider subclasses for different categories (Physical Move, Special Move) or effects (Status Move).
* **Gameplay Mechanics:**
* **Battle system:**
* Implement turn-based combat with move selection and damage calculation.
* Utilize random number generation for accuracy checks.
* Track and apply status effects (poison, paralysis, etc.).
* **Pokémon catching**:
* Use a Pokeball class with different catch rates based on factors like Pokémon health and level.
* Introduce Pokeball types with varying effectiveness.
* **Additional Features:**
* **Evolution system:** Implement evolution mechanisms for certain Pokémon based on level or specific items.
* **Note:**
* Start with core functionalities and build progressively.
* Test code thoroughly at every stage.
* Keep the game design engaging and balanced.
* Utilize OOP principles effectively for code organization and reusability.
* You are allowed to use Gemini/ChatGPT for your reference, but please be honest with yourself, code the entire game yourself.

**Good luck!**