**Report**

**Pokémon Center Management Application**

The Pokémon Center Management Application was developed to manage trainers and Pokémons. It provides an interface to register, view, and manipulate Pokémon data.

**Classes:**

1. **Trainer** associates each Pokémon with a trainer.  
   Represents a Pokémon trainer with attributes like firstName, lastName, and a unique number.
2. **Pokemon (Abstract Class)**  
   Encapsulates common properties of all Pokémon, such as id, type, level, and their associated Trainer. Implements the Printable interface (returnInfoString method).
3. **Specific Pokémon Classes** inherit from Pokemon and implement their unique attributes:

**CommonPokemon** includes an equippedItems list.

**SpecialPokemon** includes an additionalCombatPower attribute.

**LegendaryPokemon** includes specialAttack and hiddenAbility attributes.

1. **PokemonCenter**  
   A management class for storing, retrieving, and manipulating Pokémons.

pokemons: A list to store registered Pokémon.

max: Maximum capacity (100 Pokémon).

**Printable Interface**: defines the returnInfoString() method, which ensures that every class implementing it can provide detailed information about its attributes.

* Class Pkemon() implements Printable interface (override returnInfoString() function) to return a string with the Pokémon's basic details (id, type) also its trainer and level.
* Each specific Pokémon class extends the base method to include the specific information about its unique attributes.

**Program Structure.**

**Main:**

* Acts as the central control loop for the application.
* Displays an interactive menu ( using menu() function). The user can choose from several options to interact with the system.
* Use a while loop to present the menu until the user selects the exit option (7).

**Features**

The program includes the following features:

1. registerPokemon(pokemon: Pokemon) - Registers a Pokémon if capacity allows.
2. listPokemon() - Returns a list of strings containing details of all registered Pokémon.
3. pokemonInfo(id: String) - Retrieves detailed information about a Pokémon by its ID.
4. increaseLevel(id: String, quantity: Int) - Increases a Pokémon's level by a specified quantity.
5. decreaseLevel(id: String, quantity: Int) - Decreases a Pokémon's level, ensuring it does not fall below
6. checkLevel(id: String): Returns the current level of a Pokémon.

**Validation**

The program includes input validation for user-provided data and provides error messages for invalid inputs:

* IDs are validated using regular expressions to ensure the required **PXXX** format.
* Assign the value to integer variables if the user enters valid integer, or default value 0 in case of invalid input or null.
* Category validation to permit only valid category( common, special or legendary).

In PokémonCenter class:

* checkId(id: String): Ensures that a given ID is not already registered.

Next methods are used to check storage capacity and register Pokémon up to the maximum allowed number:

* getCurrentSize(): Returns the number of currently registered Pokémon
* getMax(): Returns the maximum capacity of the center.