Modeling and implementation of a Pokedeck

The purpose of this TP is to implement a Pokedeck, i. e. software to manage your pokemon cards, for the card game as described in this booklet : http://assets19.pokemon.com/assets/cms2/pdf/trading-card-game/rulebook/xy1-rulebook-en.pdf

Depot :

GitHub : https://github.com/sckraa/JAVA

Pokedeck :

The source code of the project is in English. Before starting to create the classes of this program and declare the methods, I first focused on structuring data for this project (analyzing the data that will be processed in this project and the project domain) and finding a computer representation that will allow me to progress faster on the project development. I started by making a class diagram by noting the different classes and adding attributes and methods.

At first I only had two classes, Deck and Card, and as the project progressed, I had to add more classes. A good practice to consider when developing a program is to first create a Test class in our project in which we will test how the program works (test instants, methods of a class, etc.). So after creating the Card class, I instantiated Card objects to check the proper functioning of the class.

After declaring the manufacturers of each class, I developed the basic features of this program: add a description of a new map and delete a map. To add a description for a map, I added an attribute for the Card class that will store the description of the map and to add or modify the description, I added a setter for this attribute. To make a mini presentation of the Deck class is a deck of cards and the cards I store them in a card type ArrayList. So to add or remove cards in the package, I used the available methods of this object (add() and remove()). To display a map, I used the following data: map id, map name, map type, if it is an energy or trainer type map, I display the corresponding type and if it is a pokemon type map, I display the life points of the pokemon and also its evolution level and in addition a description.

Then I started by adding the functionality to act with the program directly in the terminal to add a card, delete a card or display all the cards in the package. For that I had to add a main menu and a sub-menu but also the possibility to modify a map, consult its collection and search for a map on different criteria. After implementing these features I also added the possibility to save the package state after leaving the program. To implement this feature I used the Serializable class.

Conclusion :

The Pokedeck project is an interesting project that has allowed me to improve my knowledge of JAVA language, to better understand how to structure data but also good practices to take into account before starting to code it.

During this project I had to encounter problems concerning the architecture of the program when creating the menu and sub-menu but also when implementing the card types. Another problem I had to encounter was the creation of graphical interfaces with swing by reusing the code that works as a user interface (terminal).

The program architecture is presented in the UML folder.