PF2 Laboratory 2: Pokédeck

Liz Schreiner, Léo Gamboa dos Santos

We designed the program by following the user interface flow, meaning the actions of the user lead the program structure more that internal architecture.
We also wanted to be able to save and load decks as one of the main features, so that feature was immediately incorporated into the design from the beginning.

One of the earlier design ideas was to have **PokemonCard**, **TrainerCard** and **EnergyCard** extend a **Card** class (which we later made abstract). While on paper this seems clean, it actually lead to multiple problems:
First GSON couldn't manage serializing to json properly because we were using **Card.class** to serialize, which would delete all card-specific data. The only work-around was to create a custom (de-)serializer, **CardAdapter**.
Next, in **Search** and **CardDeck**, the generalization of all cards as **Card** for filtering and sorting lead to generics problems, which is why most lambdas in both classes use **<? extends Card>** as their generic type.

To be able to centralize the rendering and user-input to facilitate debugging, expanding and adjusting we made the **Display** and **Util** classes which statically handle all the ANSI and terminal shenanigans.

A last problem in the current code is hardcoded visual aspects. We tried to make colors and positions dynamic, but because of the way the visual designs are defined separately in a .txt file, they are disconnected and changing either of them could mess up the entire interface. Apart from that we believe the rest of the visual-related code is clean.

As for tests, unfortunately we couldn't automate testing due to the way our user interface works.