**Nick Domenico**

**Pokémon TCG Program**

**Features of my Program:**

1. Allows 2 users to play a basic version of the Pokémon Trading Card Game against each other
2. It implements parts of the 2022 Pokémon TCG Battle Academy:
   * A beginner friendly version of the game that comes with 3 decks whose cards all have simple effects for new players to learn how to play the game.
   * The 3 decks can be found here:
     + <https://bulbapedia.bulbagarden.net/wiki/Battle_Academy_2022_(TCG)>
   * I implemented 2 of the 3 decks included:
     + Cinderace V
     + Pikachu V
   * I left out the Eevee V deck to save time, but it could be added later
3. All methods in my program are documented with Javadoc style comments

**Notes:**

1. A collection of cards with numbers

   Description automatically generatedOriginally, I wanted to implement the 2 decks that I actually play with competitively, but I quickly realized that the effects of the cards would be too complicated to program, but regardless, here are my deck lists if you are interested:

A collection of cards with numbers

Description automatically generated

1. The class in my program named “Player” was originally the “CardGame” class that you coded in class with us, but I renamed it because I felt that each Player should have their own deck, hand, etc., instead of CardGame
2. The way that I handled drawing cards from the deck is the same way that you showed us in class (picking a random index to draw from the deck instead of drawing from the top of the deck). This works for the purposes of the decks in this program, but for certain cards in my decks that I didn’t implement, this wouldn’t work; namely Iono, whose effect shuffles both players hands and puts them on the bottom of their decks, then they draw a card for their remaining prizes. By putting your hand on the bottom of your deck, you won’t be able to draw any of those cards with the 2nd part of the effect, but this wouldn’t work with implementing drawing by drawing from random indices.
3. I did not implement the effects of Weaknesses, as none of the cards in either deck that I programmed are able to hit each other for Weakness. I do still have a weakness instance variable in the Pokémon class, but it is only for displaying the text on the card.
4. I did not implement the effects of Resistances, as none of the cards in either deck have Resistances. I do still have a resistance instance variable in the Pokémon class, but it is only for displaying the text on the card.
5. When implementing the check for Energy Cost of attacks, I only looked for the correct number of Energies attached rather than their types, as each deck only uses 1 type of Energy, and the attacks of every Pokémon in each deck only use the deck’s respective Energy Type, or Colorless.
6. All of the user prompts in my program don’t have exceptions to handle unexpected inputs, this was done to save time but results in the game crashing when a user inputs a value that is not expected.
7. I probably could’ve created a method for user prompts in my program that way I wouldn’t have had to type out: “System.out.println(“prompt”); userInput = in.nextInt();” every single time, but I realized this when I was around halfway done, and didn’t want to spend the time rewriting everything that I had already written. I also wanted to be consistent across my whole program, so I didn’t make a prompt method halfway through and have some parts use it and others not.