UF UNIVERSITY of FLORIDA
Herbert Wertheim
College of Engineering

COP 3502

Pokedex Project

Overview

This project will help students understand OOP concepts. By completing this project, students will know how to create classes and objects. Students will also know how to interact objects with other objects.

Structure

Pretend that you are Ash Ketchum, or some other Pokemon trainer. In your everyday life, you encounter different Pokemon. Your job is to record the info of each Pokemon that you encounter to a Pokedex.

You must have three classes for your program: Pokemon, Pokedex, and Main.

The Pokemon class will contain the blueprint for the different Pokemon objects that you will create. Each Pokemon object will have the following data types: `species, attack, defense,` and `speed`. `species` will be the name of the Pokemon, such as Pikachu or Bulbasaur. `attack`, `defense`, and `speed` are integer data types, and they will be initialized using these formulas:

```
attack = species.length() * 4 + 2
defense = species.length() * 2 + 7
speed = species.length() * 3 + 5
```

Note: All the data types will be initialized in a constructor. You will have to create getter and setter methods for each data type in the Pokemon class. If you do not know what is meant by getter and setter methods, the headers of the methods are shown below:

```
/*
 * When called, the getAttack method will return the attack
 * variable.
 */
public int getAttack(){ … }


/*
 * When called, the setAttack method will set the attack
 * variable to whatever value is passed to the method.
 */
public void setAttack(int newAttack){ … }
```

```
/*
 * When called, the getDefense method will return the defense
 * variable.
 */
public int getDefense(){ … }


/*
 * When called, the setDefense method will set the defense
 * variable to whatever value is passed to the method.
 */
public void setDefense(int newDefense){ … }


/*
 * When called, the getSpeed method will return the speed
 * variable.
 */
public int getSpeed(){ … }


/*
 * When called, the setSpeed method will set the speed variable
 * to whatever value is passed to the method.
 */
public void setSpeed(int newSpeed){ … }


/*
 * When called, the getSpecies method will return the species
 * variable.
 */
public String getSpecies(){ … }


/*
 * When called, the setSpecies method will set the species variable
 * to whatever value is passed to the method.
 */
public void setSpecies(String newSpecies){ … }
```

You will also have to implement the evolve method:

```
/*
 * When called, the evolve method will double the speed stat,
 * triple the attack stat, and multiply the defense stat by 5.
 */
public void evolve(){ … }
```

For the evolve method, make sure you use the getter and setter methods that you created for the Pokemon class.

The Pokedex class will contain all the Pokemon that you will encounter. The Pokedex class will contain an array of Pokemon objects. The Pokedex class should implement the following the methods in addition to a constructor:

```
/*
 * Return all the names of the Pokemon species in the
 * Pokedex
 */
public String[] listPokemon(){ … }


/*
 * Add a Pokemon to the Pokedex and return true if it can
 * actually be added to the Pokedex. If not, return false.
 */
public boolean addPokemon(String species) { … }


/*
 * Return the stats of a certain Pokemon that you are
 * searching for.
 */
public int[] checkStats(String species){ … }


/*
 * Sort Pokedex in lexical order according to Java string ordering.
 */
public void sortPokedex(){ … }


/*
 * Evolve a certain Pokemon that you are searching for in the
 * Pokedex and return true if the Pokemon is actually in the
 * the Pokedex. If not, return false.
 */
public boolean evolvePokemon(String species){ … }
```

One thing to note about the array of Pokemon objects: It is a static array. The Pokedex will contain a number of Pokemon based on what the user inputs at the beginning of the program. The number of Pokemon species will never grow beyond what the user inputs. When the Pokedex is full and a user tries to add a Pokemon, output should be "Max". If there is already a Pokemon with the same species name (ignoring casing), output should be "Duplicate"

The Main class will be used to run your program. Your Main class will utilize the Pokedex class. The Main class should have user input for the number of Pokemon located in the area. A menu should then be printed out to the user to show what he can do with the Pokedex. The menu should have following options:

1. List Pokemon (List all the Pokemon species in the Pokedex)
2. Add Pokemon (Adds a Pokemon to the Pokedex)
3. Check Pokemon Stats (Checks the stats of a certain Pokemon)
4. Evolve Pokemon (Evolves a certain Pokemon)

5. Sort Pokemon (Sorts Pokemon in alphabetical order)
6. Exit (Exits the program)

Note: Make sure that your program validates user’s input.

Submissions

*Output needs to be formatted exactly like the provided sample output.

Name your files Pokemon.java, Pokedex.java, and Main.java. If you choose to disobey these instructions, then you will receive a ZERO.

Submit your zipped src folder at the end of this project. Do not submit any other files.

***NOTE: Make sure you implement all the methods that were mentioned above. Please, make sure that your method headers match to what is shown on the instructions. We are going to use a testing suite to grade your projects. If the method headers do not match to what is shown on the instructions, then you will not receive a high grade.

Sample Output

```
Welcome to your new PokeDex!
How many Pokemon are in your region: 30

Your new Pokedex can hold 30 Pokemon. Let’s start using it!

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 2

Please enter the Pokemon’s Species: Pikachu

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 1
1. Pikachu

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 2

Please enter the Pokemon’s Species: Raichu

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
```

```
5. Sort Pokemon
6. Exit

What would you like to do? 3

Please enter the Pokemon of interest: Abra
Missing

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 2

Please enter the Pokemon’s Species: Charmander

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 3

Please enter the Pokemon of interest: Charmander

The stats for Charmander are:
Attack: 42
Defense: 27
Speed: 35

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 3

Please enter the Pokemon of interest: Pikachu

The stats for Pikachu are:
Attack: 30
Defense: 21
Speed: 26

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 4

Please enter the Pokemon of interest: Charmander
Charmander has evolved!

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 4
```

```
Please enter the Pokemon of interest: Squirtle
Missing

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 3

Please enter the Pokemon of interest: Charmander

The stats for Charmander are:
Attack: 126
Defense: 135
Speed: 70

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? -2

That is not a valid choice. Try again.

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 5

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 1
1. Charmander
2. Pikachu
3. Raichu

1. List Pokemon
2. Add Pokemon
3. Check a Pokemon’s Stats
4. Evolve Pokemon
5. Sort Pokemon
6. Exit

What would you like to do? 6
```