ACIT2515 - coding assignment

Pokemon relationships

```
┌─────────────────────────────┐        ┌────────────────────────────────────────┐
│      Ⓒ  Pokemon             │        │         Ⓒ  Arena                        │
├─────────────────────────────┤        ├────────────────────────────────────────┤
│  o name: str                │        │  □ internal implementation is up to you │
│  o health: int              │────────┤                                         │
│  o level: int               │        │  ● load_from_file(filename: str): None  │
├─────────────────────────────┤        │  ● add(pokemon: Pokemon): None          │
│  ● join(arena: Arena): None │        │  ● active(): Pokemon[]                  │
│  ● level_up(): None         │        │  ● get_pokemons(): Pokemon[]            │
└─────────────────────────────┘        └────────────────────────────────────────┘
```
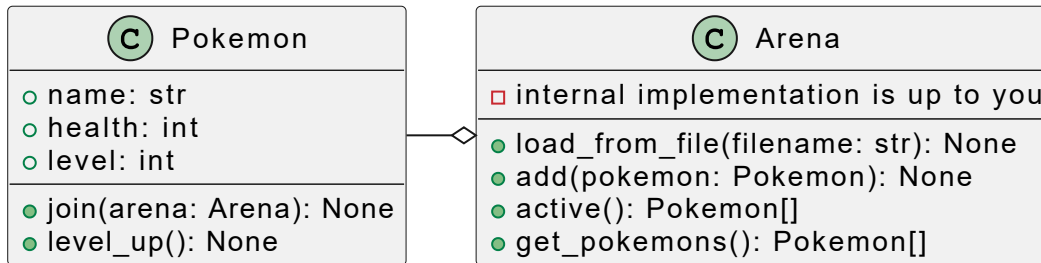
Pokemon

A Pokemon has the following attributes:

- `name` (string)
- `health` (integer)
- `level` (integer)

The constructor ONLY receives `name` and `health`. A new Pokemon will always have its level set to 1.

- the `name` must be a non-empty string
- `health` must be a strictly positive integer
- if any of these conditions is not respected, raise a `ValueError` exception.

You can check your code with the tests in `test_pokemon.py`. Note: the `join` method will be tested later (see below).

Arena

An arena is a place where pokemons get together.

- an arena contains pokemons
- you can add a pokemon to an arena: this uses `add` on the arena instance. Only pokemons or creatures inheriting from pokemons can be added to an arena.
- a pokemon can also join an arena with the `join` method: `pokemon.join(arena)`. Implement this method on the Pokemon class. Do not make any type checks - you only need a single line of code in that method!
- the `active` method returns a list of all **active** pokemons (i.e. whose health is > 0) in the arena
- the "length" of the arena is the number of **active** pokemons in it
- the method `get_pokemons` returns a list of **all** pokemons, sorted by name
- it is possible to load pokemons from a file into an arena with the `load_from_file` method
    - `load_from_file` takes one argument: the name of the file to read from
    - the file is in CSV format and has three columns: name, health and level of the pokemons

Submission and grading

- Make sure all tests in `test_arena.py` and `test_pokemon.py` pass.
- Submit your files to D2L.
- 1 mark per test = **10 marks total**
- Comments, formatting, code quality: **2 marks**