Requirements

- A new world must start with a number of trainers, at least 2, each of them starts with at least one code-a-mon. You can also decide to have trainers join in time if you like.
- The trainer can acquire more code-a-mons (max of 6) throughout their adventure. How they acquire them is up to you. An example could be they win one as a reward for every X amount of battle victories or they can catch them.
- Code-a-mons will compete 1v1 with another trainers code-a-mons.
- The simulation should run on cycles. A cycle is considered to be of 2 parts - 1 day time and 1 night time.
- Each cycle should have it's own weather event (these are some examples to give you ideas, use your creativity!):
  - Day 1 - Sunny
  - Night 1 - Clear
  - Day 2 - Rainy
- Weather events should benefit certain types of code-a-mon's stats while being a disadvantage to others (these are some examples to give you ideas, use your creativity!):
  - Sunny: Fire type gains 25% increase in stats and decreases water type by 25%.
  - Clear: Neutral.
  - Rainy: Water type gains 25% increase in stats and decreases fire type by 25%.
- Code-a-mons should be of different types and gain advantages or disadvantages based on their opponent's type (these are some examples to give you ideas, use your creativity!):
  - Water > Fire
  - Fire > Grass
  - Or, you can get creative and change the types from elements to coding languages

    C++>Java

    Java>Python
- Battles with other trainers (or wild code-a-mons are done during the day time). Battles with trainers earn money and experience points while battles with wild code-a-mons only earns experience points.
- During the night these things are possible (choose one or all):

  - Code-a-mons can heal (based on whatever you come up with, maybe potions maybe a percentage, be creative)
  - Can attempt to catch a new code-a-mon (if you chose to catch them) or if they reached the win threshold to receive one, they would get it at this time.
  - Evolutions of code-a-mons can occur.
  - Items can be purchased from the store with money. Items could be potions, items to catch wild code-a-mons with, stat boosters, etc.
- Code-a-mons should have at minimum:
  - Stats: Attack, Defense, Health (Others like speed and so forth can be added if you would like to implement).
  - They should have 1-4 different attacks
  - Each attack has a specific type which gains bonus damage if it matches the type of the user.
  - Attacks should have a chance to critical strike (Double damage).
  - Attacks should have a chance to miss.
  - Attacks should deal a minimum of 1 damage unless they use an attack that deals 0 base damage.
- Code-a-mons gain experience points from winning battles and can level up after earning enough points. Evolutions can occur after reaching certain levels.
- Only one battle can take place at a time. A battle is always between two trainers and each using one code-a-mon. OPTIONAL: You can also decide to have one main trainer and everyone battles against that main trainer instead of all trainers battling with each other.
- During battles with trainers:
  - Each trainer has 1 code-a-mon on the field at a time.
  - Attacks should be performed in a turn-based manner, one code-a-mon attacks while the other defends, then vice-versa until one faints.
  - When a code-a-mon's health reaches 0 or less, they faint, and this particular fight is over. Code-a-mons can heal during the night and be awake again the next day, maybe with less strength or depending on potions etc.
  - When a trainer has no code-a-mon left (no code-a-mons awake during the day), they leave the battlefield
  - Experience points can be handled in two ways: when a code-a-mon defeats another or when the battle is over, it can be given either to the entire team or to just the code-a-mon that won.
  - For a trainer's turn, they can either attack or use an item such as a potion, stat booster, etc.

It is all purposefully pretty wide open to give you more options and have you think about things more. You can be creative on this; if you like a different idea better, go for it. It is more about having a rough framework in which you want to work and doing something fun with implementing Design Patterns than the particular requirements.

Some HINTS (not requirements):

- You could use decorator pattern for evolutions.
- To build new code-a-mon or trainers you could use the factory pattern.
- The simulation should be tick-based. (Mediator pattern). Each tick something should happen, the trainer battles, catches new code-a-mon, purchases items, etc.