EXERCISES TO COMPLETE

Exercise 1 - Guessing Game

You are required to write a class modelling a simple guessing game. The informal specification for this class is given below.

Specification

The guessing game begins by choosing an integer between 1 and 100, which the player has to guess. The player wins the game by guessing the number correctly within 10 guesses.

The player can:

- Get the number of guesses remaining.
- See whether or not the game is over.
- See whether or not the game has been won.
- Make a guess. This reduces the number of guesses remaining by 1, and prints whether the guess is correct, too high, or too low. It should also update any other fields as appropriate. A guess can only be made if the game is not over.
- Print the correct number — but only if the game is over.

The constructor and methods specified above must be `public`; all fields of the class must be `private`. It is not necessary to write any additional methods beyond those specified.

Exercise 2 - Tamagotchi

Use the informal specification for a Tamagotchi given below to model and implement the class.

Your implementation should include:

- Methods for each action, e.g. feed the Tamagotchi, take it for a walk…
- A method to get the Tamagotchi's current mood, e.g. happy, hungry, tired…
- A method to "pass time", modelling what happens when you leave the Tamagotchi alone for a short while.

Your implementation should reflect how the internal state of the Tamagotchi is represented, i.e. the fields, which should include how hungry it is, how tired it is, and so on. These are represented as numbers within the range 0–10.

Implementation

- Write a class named `Tamagotchi` according to the given specification. It must have the fields and methods as a bases for your implementation.

- Write a constructor which takes no parameters, and initialises the Tamagotchi object with a sensible initial state.

  The class, constructor, and the methods you specified must be `public`; the fields must be `private`. If you write any additional methods beyond those required by your specification, they must be `private`.

Informal Tamagotchi Specification

A Tamagotchi has hungriness, happiness, cleanness and tiredness. These are integers in the range 0-10. Make these accessible via getter methods.

You can interact with the Tamagotchi by:

- Feed the Tamagotchi
  - It gets less hungry (its hungriness goes down by 5)
- Take the Tamagotchi for a walk
  - It gets happier by (3) and more tired by (2)
- Clean the Tamagotchi
  - It gets cleaner (goes up by 10)
- Pet the Tamagotchi
  - It gets happier (by 5)
- Get the Tamagotchi's current mood, as String
  - If tiredness is 10, then the Tamagotchi is "asleep"
  - If tiredness is at least 8, then the Tamagotchi is "tired"
  - If hungriness is at least 7, then the Tamagotchi is "hungry"
  - If cleanness is less than 3, the Tamagotchi is "dirty"
  - If happiness is at least 8, the Tamagotchi is "happy"
  - If happiness is at least 4, the Tamagotchi is "Ok"
  - Otherwise the Tamagotchi is "sad"
- Pass time
  - Hungriness goes up by 1
  - Cleanness goes down by 1
  - Tiredness goes up by 1, unless the tiredness is 10, in which case the Tamagotchi wakes up with a tiredness of 0
  - If Tamagotchi hungriness is at least 7, decrease happiness by 1
  - If Tamagotchi cleanness is less than 3, decrease happiness by 1

None of these arguments takes a parameter. If any of these methods reduce one of the integers to below 0, then that integer is set to 0.

If any of these methods raise one of the integers to above 10, then that integer is set to 10.