Exercises - Session 13

In case you get stuck anywhere, don't be afraid to ask the coaches! They are here to help and will gladly explain everything to you!
Take notes during the exercises. Even if you never look at them again, they will help you memorise things!

The Cat Organiser

In this exercise, we are going to implement a little program that lets you organise your feline friends. For the sake of simplicity, each of the cats that we'll deal with has exactly three properties, no more and no less:

- A name
- The number of lives it has left (out of its 9 lives)
- The type of its coat (e.g. tabby or tuxedo)

1. Let's start by thinking about the data structures we are going to use for our program. What data types should we use for each of the following things? Think of "simple" data types like numbers and strings but also about the more complex ones like arrays and hashes.

    - The cat's name
    - The number of lives
    - The type of its coat
    - The whole cat
    - The collection of all of your cats

    After you have thought about this for a while, talk to a coach to verify your findings. You will need the solution of this exercise for the rest of them!

2. We need define some initial data to play with in order to start programming. Create a new file called "cats-variable.rb". In that file, define a variable called `cats` that contains the data in the following table. Use the data types you thought of in the first exercise to define this variable.

Name	**Number of lives**	**Coat type**
Miffles	9	tabby
Sergeant Fuzzyboots	5	black
Gorbatchev Puff-Puff Thunderhorse	7	cream

3. Now let's do something with those cats! Below is some code to get you started. Take some time to roughly understand what it does. If there's anything unclear about it, tell a coach to explain that part to you!
   Copy and paste the code to a file called "cats.rb". If you run the code like it is (with the command `ruby cats.rb`), you will end up with an error. Why is that?
   Copy and paste the code from your cats-variable.rb file and put it in the right place in cats.rb. Then re-run your code. Does it work?

```
# TODO: define your cats variable here!

def ask(question)
  print "#{question} "
  gets.chomp
end

def list_cats(cats)
  # TODO: implement outputting a list of cats to the console here!
end

def add_a_cat(cats)
  # TODO: implement adding a cat to your cats variable here!
end

def remove_a_cat(cats)
  # TODO: implement removing a cat from your cats variable here!
end

loop do
  list_cats(cats)

  input = ask("What would you like to do?")

  if input == "add"
    add_a_cat(cats)
  elsif input == "remove"
    remove_a_cat(cats)
  end

  if input == "exit"
    break
  end
end
```

4. The user of our cat organiser should see the current list of cats when running the program. The output should look like this:

   ```
   Miffles is a tabby cat that has 9 lives.
   Sergeant Fuzzyboots is a black cat that has 5 lives.
   Gorbatchev Puff-Puff Thunderhorse is a cream cat that has 7 lives.
   ```

```
What would you like to do?
```

What part of the code do you need to change in `cats.rb` to make this happen? Implement that part now!

5. The cat organiser wouldn't be much of an organiser if you couldn't add more cats. So we're going to tackle that part next. Here's what the output of your program should look like. Note: The parts that the user is entering are printed in **bold**.

```
Miffles is a tabby cat that has 9 lives.
Sergeant Fuzzyboots is a black cat that has 5 lives.
Gorbatchev Puff-Puff Thunderhorse is a cream cat that has 7 lives.
What would you like to do? add
What's your cat's name? Felis
How many lives does it have? 3
What type is its coat? tabby
Miffles is a tabby cat that has 9 lives.
Sergeant Fuzzyboots is a black cat that has 5 lives.
Gorbatchev Puff-Puff Thunderhorse is a cream cat that has 7 lives.
Felis is a tabby cat that has 3 lives.
What would you like to do?
```

Again, find the part of the code where this should go, then start implementing it. If you get stuck, ask the coaches!

6. Finally, we'd like to be able to remove cats from the list. Your program should behave like this when run:

```
Miffles is a tabby cat that has 9 lives.
Sergeant Fuzzyboots is a black cat that has 5 lives.
Gorbatchev Puff-Puff Thunderhorse is a cream cat that has 7 lives.
What would you like to do? remove
What's the name of the cat to remove? Mittens
There is no cat called Mittens!
Miffles is a tabby cat that has 9 lives.
Sergeant Fuzzyboots is a black cat that has 5 lives.
Gorbatchev Puff-Puff Thunderhorse is a cream cat that has 7 lives.
What would you like to do? remove
What's the name of the cat to remove? Miffles
Sergeant Fuzzyboots is a black cat that has 5 lives.
Gorbatchev Puff-Puff Thunderhorse is a cream cat that has 7 lives.
What would you like to do?
```

Note that the user has misspelled the name the first time around. As shown, the program should handle this case by telling the user about her mistake. Make sure to also implement this behaviour, but you can start with the case of the correct name.