Work on project. Stage 4/6: Long-term memory

Project: Guess the Animal

Long-term memory

Hard 10 minutes ? **88** users solved this stage. Latest completion was **11 days ago**.

Description

Our program is growing and learning so quickly, but just as quickly it forgets everything as soon as we finish the game session. Let's save the precious information: in this stage, you need to implement saving the accumulated knowledge. Now, when the user starts the program, the computer will load the knowledge tree saved in the memory from the previous session.

Theory

We will save the knowledge base in one of the following formats: JSON, XML, YAML. For this, we will use the library Jackson.

First of all, we must add the required dependencies to our *build.gradle* file:

```
dependencies {
    compile group: 'com.fasterxml.jackson.core', name: 'jackson-
databind', version: '2.11.2'
    compile group: 'com.fasterxml.jackson.dataformat', name: 'jackson-
dataformat-xml', version: '2.11.2'
    compile group: 'com.fasterxml.jackson.dataformat', name: 'jackson-
dataformat-yaml', version: '2.11.2'
}
```

27 / 27 Prerequisites

- ✓ Overloading 1
- ✓ Multiple constructors 1
- ✓ The keyword super 1
- ✓ Hierarchy of exceptions 1
- ✓ Exception handling 1

Show all

Join a study group for the project Guess the Animal

Discuss your current project with fellow learners and help each other.

In order to map a Java object to the file, we should create an instance of the corresponding ObjectMapper class. For JSON, it is `JsonMapper()`, for XML — `XmlMapper()`, and for YAML file format it is `YAMLMapper()`.

Consider an example. Say we have a Java class to describe a tree node:

```
public class TreeNode {
    private String data;
    private TreeNode yes;
    private TreeNode no;

    public TreeNode() {
    }
    // Public constructor is required to map java class
    // We skip the public Setters and Getters ...
}
```

In our program, we define the tree root as `TreeNode root` and build a knowledge tree during game sessions. Now we would like to save our tree to the disk in JSON format. To accomplish this task, we should define the file name and create an instance of ObjectMapper:

```
    String fileName = "animals.json";
    ObjectMapper objectMapper = new JsonMapper();
```

Then, we can save our knowledge tree to JSON file using the statement:

```
    objectMapper
            .writerWithDefaultPrettyPrinter()
            .writeValue(new File(fileName), root);
```

To load our knowledge tree, we can use the following command:

```
    root = objectMapper.readValue(new File(fileName), TreeNode.class);
```

For both cases, you should try/catch possible exceptions.

To avoid storing null values in the JSON file and skip parsing additional methods that can be presented in our TreeNode class, we can use Jackson Annotations:

```
@JsonInclude(JsonInclude.Include.NON_NULL)
public class TreeNode {
    private String data;
    private TreeNode yes;
    private TreeNode no;

    public TreeNode() {
    }

    @JsonIgnore
    public boolean isLeaf() {
        return no == null && yes == null;
    }

    // public Setters and Getters ...
}
```

Here is an example of a JSON file:

```
{
  "data" : "It is a mammal",
  "yes" : {
    "data" : "It is living in the forest",
    "yes" : {
      "data" : "It has a long bushy tail",
      "yes" : {
        "data" : "a fox"
      },
      "no" : {
        "data" : "It is a shy animal",
        "yes" : {
          "data" : "a hare"
        },
        "no" : {
          "data" : "a wolf"
        }
      }
    },
    "no" : {
      "data" : "It can climb trees",
      "yes" : {
        "data" : "a cat"
      },
      "no" : {
        "data" : "a dog"
      }
    }
  },
  "no" : {
    "data" : "a shark"
  }
}
```

Objectives

When starting the program, the user can specify the parameter "-type" with one of the following options: "json", "xml", or "yaml". If the parameter is not specified, the program should select the default format "json".

You should polish the output before saving it to the disk. This will allow you to easily review the file in any text editor and make changes to it if necessary.

At the start, the program must search for the file with the database. If the file isn't found, the computer must ask the user about their favorite animal and create a new knowledge tree.

Examples

The greater-than symbol followed by a space `>` represents the user input. Note that it's not part of the input.

Example 1

```
Good morning!

I want to learn about animals.
Which animal do you like most?
> cat
Wonderful!
I've learned so much about animals!
Let's play a game!
You think of an animal, and I guess it.
Press enter when you're ready.
> Is it a cat?
> No
I give up. What animal do you have in mind?
> a dog
Enter a statement which can help me distinguish a cat from a dog.
> It can climb trees.
Is that fact correct for a dog?
> No
I remember the following facts about animals:
- The cat can climb trees.
- The dog can't climb trees.

I've learned so much about animals!
Would you like to play again?
> No

Have a nice day!
```

Example 2

If a knowledge base already exists, the program doesn't ask the user about the favorite animal: it offers to play a game instead.

```
Hi Night Owl!

I know a lot about animals.
Let's play a game!
You think of an animal, and I guess it.
Press enter when you're ready.
>
Can it climb trees?
> Sure!
Come on, yes or no?
> Yeah
Is it a cat?
> Nope
I give up. What animal do you have in mind?
> Lynx
Enter a statement that can help me distinguish a cat from a lynx.
> It has tassels on its ears
Is the fact correct for a lynx?
> yea
I remember the following facts about animals:
- The cat hasn't tassels on its ears.
- The lynx has tassels on its ears.

I've learned so much about animals!
Would you like to play again?
> No

Good night!
```

Report a typo

HINT by BastiHz Viewed hints

For the TreeNode class it is important to either make the fields public or have public getters and setters for them. Otherwise they can't be serialized.

This is the last hint available for this problem! Please post your own hint after completing the problem, future learners will thank you.

Write a program

Code Editor IDE

CONNECTION STATUS

IDE / Checking the plugin's status

Solve in IDE

Look up solution (100)

Comments (16) Hints (4) Useful links (1) Solutions (20) Show discussion