

SOLID / Design Patterns Coding Exercise

This exercise is the "infrastructure" of others future exercises. Briefly , there is a console program that opens a prompt and reads commands from the keyboard. The first commands to implement are dedicated to read/write parameters from text files. Those parameters are organized by "namespace", where the namespace is the name of the file.

In the future exercises, we will use those parameters as row data for specific business. For example if a namespace is called "pizza hawaii" and another "pizza rodeo" and inside of it I have parameters as : cheese=300g, tomato=200g, etc ... the namespace is a recipe for a pizza, then we can invent an exercise as "if you type in command line *make 5 hawaii 5 rodeo* " will create an output with 10 lines and use Prototype design pattern ... And we can invent such exercises to pass through all design patterns.

So, please provide in a text file a section of self crafted code (in a language of your choice) with associated unit tests that demonstrates use of SOLID principles and Design Patterns to meeting the following acceptance criteria:

Given I have a command line window open
When the program starts
Then the program set the current namespace to "general"
And displays "Current namespace : general"

Given I have a command line window open
And I type in the command "ns {namespace_name}"
When press return
Then the program set the current namespace to {namespace_name}
And displays "Current namespace : {namespace_name}"

Given I have a command line window open
And I type in the command "set {parameter_name} {parameter_value}"
When press return
Then the program inserts or updates in the current namespace a parameter with the name {parameter_name} and the value {parameter_value}
And displays "{namespace_name} : {parameter_name} = {parameter_value}"

Given I have a command line window open
And I type in a the command "get {parameter_name}"
When press return
Then the program reads the parameter with the name {parameter_name} from the current namespace
And displays "{namespace_name} : {parameter_name} = {parameter_value}"

Given I have a command line window open

And I type in a the command "list"

When press return

Then the program reads all the properties from all namespaces ordering alfabetically the namespaces and the parameters

And displays "{namespace_name} : {parameter_name} = {parameter_value}"

Given I have a command line window open

And I type in a the command "list {namespace_name}"

When press return

Then the program reads the properties of the namespace {namespace_name}

And displays "{namespace_name} : {parameter_name} = {parameter_value}"

Given I have a command line window open

And I type in a the command "load {namespace_name}"

When press return

Then the program reads the properties of a the text file {namespace_name}.properties located in the current directory and stores them in the namespace {namespace_name}

And displays "{namespace_name} : loaded X parameters." , where X is the number of parameters loaded

Given I have a command line window open

And I type in a the command "save {namespace_name}"

When press return

Then the program stores the properties of the namespace {namespace_name} a the text file with the name {namespace_name}.properties located in the current directory

And displays "{namespace_name} : saved X parameters." , where X is the number of parameters saved

Given I have a command line window open

And I type in a the command "load"

When press return

Then the program reads all the files with extension .properties located in the current directory and stores all the properties of a file in the namespace {name_of_the_file}

And displays for each file "{namespace_of_the_file} : loaded X parameters." , where X is the number of parameters loaded

Given I have a command line window open

And I type in a the command "save"

When press return

Then the program stores the properties of each namespace into a file with the name {namespace_name}.properties located in the current directory

And displays for each file "{namespace_name} : saved X parameters." , where X is the number of

parameters saved

Observations:

This example will be used as base for others examples, where the parameters will have meaning.

Other commands will be added in the future

Those other examples will help to implement others design patterns.

1. Format of .properties file:

param1=value1

param2=value2

2. try to demonstrate a TDD approach, but this is not mandatory

hints:

try to use the following design patterns:

- Factory Method (for reading the files)
- Singleton (to store the properties)
- Interpreter (to read command line)
- Interpreter (to parse each line of the file)
- Strategy (ordering the namespaces and the parameters on list

commands)

3. use integration tests and unit tests

4. submit your code on git: //github.com/radu-obreja/tdd.git in a folder named as your git account name

5. Deadline : 24 April 2013