CS 474: Object Oriented Programming Languages and Environments

Fall 2021

First Ruby project

Due time: 11:59 pm on Sunday 10/3/2021

In this project you will explore the *reflection* capability of Ruby (also known as *metaprogramming* in Ruby). In brief, reflection gives you the ability to examine the classes and objects in a running program, for instance, finding out the methods and variables in a class.

Using reflection, you are to create a *class browser* for Rubymine Ruby using command line prompts. Your program works as a continuous read-eval-print loop. The browser will read a user's input command, process the command and print information about Ruby classes and their methods. Initially the browser will print information about class *Object*, which you may view as the root of the class hierarchy. For each class that you visit, the browser will print by default the following information:

1. The name of the class
2. The name of the class's superclass
3. An enumerated list of the class's subclasses, in alphabetical order
4. An enumerated list of the instance methods (public, protected and private) defined in the class, along with their access levels in alphabetical order. The list should not include inherited methods.
5. An enumerated list of the class's instance variables, in alphabetical order.

Next, the browser will wait for one of the following user commands:

- `s` — Display the five items of information above for the superclass of the current class. If the current class has no superclass because it is the root of the class hierarchy, you should display a suitable message and wait for new input. Otherwise, the superclass of the current class becomes the current class.
- `u` n — The *n*-th subclass of the current class becomes the current class. By default display the five items of information above for the new current class. If *n* is out of bounds for the list of subclasses of the current class a suitable error message is displayed and no further action is taken.
- `m` — This command prints a list of the public class methods defined in the current class. Protected methods, private methods, instance methods and inherited methods (of all kinds) are not printed.
- `c` *aString* — The input string must denote a class name. The named class becomes the current class. The four items of information above are displayed for the current class. If there no such named class, a suitable error message is displayed and no further action is taken.
- `q` — This command exits your browser.

In order to keep the method list and subclass list in alphabetical order, you must implement the popular recursive sorting algorithm *MergeSort* in Ruby.

Hints: Class objects will hold all the information that you need about each class. Use the `ObjectSpace` class to go over all the objects in a running Ruby environment. For instance the following expression lists all the direct and indirect subclasses of a given class *C1*:

$$\texttt{ObjectSpace.each\_object(Class).select } \{ \mid \texttt{kl} \mid \texttt{kl} < \texttt{C1} \}$$

Also, look up the API of class *Object* at URL https://ruby-doc.org for additional methods that allow you navigate the class hierarchy.

You must work alone on this project. Submit all your source code files in a zip archive named `your_last_name.zip`. Submit the file using the Blackboard course web site. No late submissions will be accepted, except for documented hardship.