CS 343 – Structure of Programming Languages
Winter 2011, 3/29/11

Programming Assignment in Ruby
Due Date: Friday, April 8, 2011

Objective
- Use the Reflection API in Ruby to inspect types defined in a Ruby source file

Project Description

Write a program in Ruby that loads the contents of a Ruby source file and uses the Reflection API to discover the types (classes and modules) defined in that file. The program will display the names of types found in the source file. The user can then request to see additional information about a given type. You can assume that an input source file contains only class and module definitions.

The program should display the following information on a type:

- Name
- Super class (only if a type is a Class)
- Ancestors
- Included modules
- Constants
- Class variables
- Class methods
- Public instance methods
- Protected instance methods
- Private instance methods

Sample Execution

`ruby Project5.rb`

```
Enter a ruby source filename to inspect: Test.rb
Load Error: Could not load the file Test.rb. Try again..
Enter a ruby source filename to inspect: MyTestClass.rb
The file MyTestClass.rb contains the following 7 types:
   1. FirstClass
   2. Student
   3. GradStudent
   4. Binary
   5. Base64
   6. Dice
   7. Game
Do you want to examine a type further (y or n)? y
Enter the number of the type to examine further: 8
Not a valid number. Try again.
Do you want to examine a type further (y or n)? y
Enter the number of the type to examine further: 1
   Type: FirstClass
   Super class: Object
   Ancestors: [FirstClass, Comparable, Object, Kernel, BasicObject]
   Included modules: [Comparable, Kernel]
   Constants: [:Constant1, :Constant2]
   Class variables: [:@@cvar1, :@@cvar2, :@@alphabet]
   Class methods: [:method1]
   Public instance methods: [:method2, :method3]
   Protected instance methods: [:method4, :method5]
   Private instance methods: [:method6, :method7]
Do you want to examine a type further (y or n)? y
Enter the number of the type to examine further: 2
   Type: Student
   Super class: Object
   Ancestors: [Student, Object, Kernel, BasicObject]
   Included modules: [Kernel]
   Constants: []
   Class variables: []
   Class methods: []
   Public instance methods: [:name, :name=, :gnbr, :gnbr=, :age, :age=, :to_s]
   Protected instance methods: []
   Private instance methods: [:initialize]
Do you want to examine a type further (y or n)? y
Enter the number of the type to examine further: 3
   Type: GradStudent
   Super class: Student
   Ancestors: [GradStudent, Student, Object, Kernel, BasicObject]
   Included modules: [Kernel]
   Constants: []
   Class variables: []
   Class methods: []
   Public instance methods: [:specialty, :specialty=, :advisor, :advisor=]
   Protected instance methods: []
   Private instance methods: [:initialize]
Do you want to examine a type further (y or n)? y
Enter the number of the type to examine further: 4
   Type: Binary
   Ancestors: [Binary]
   Included modules: []
   Constants: []
   Class variables: []
   Class methods: [:to_bin]
   Public instance methods: [:to_bin]
   Protected instance methods: []
   Private instance methods: []
Do you want to examine a type further (y or n)? y
Enter the number of the type to examine further: 5
   Type: Base64
   Ancestors: [Base64]
   Included modules: []
   Constants: []
   Class variables: []
   Class methods: [:encode, :decode]
   Public instance methods: []
   Protected instance methods: []
   Private instance methods: []
Do you want to examine a type further (y or n)? y
Enter the number of the type to examine further: 6
   Type: Dice
   Ancestors: [Dice]
   Included modules: []
   Constants: []
   Class variables: []
   Class methods: []
   Public instance methods: [:roll]
   Protected instance methods: []
   Private instance methods: []
Do you want to examine a type further (y or n)? y
Enter the number of the type to examine further: 7
   Type: Game
   Super class: Object
   Ancestors: [Game, Dice, Object, Kernel, BasicObject]
   Included modules: [Dice, Kernel]
   Constants: []
   Class variables: []
   Class methods: []
   Public instance methods: []
   Protected instance methods: []
   Private instance methods: []
Do you want to examine a type further (y or n)? n
Do you want to inspect another ruby source file (y or n)? y
Enter a ruby source filename to inspect: Reflection.rb
The file Reflection.rb contains the following 1 types:
   1. TestClass
Do you want to examine a type further (y or n)? y
Enter the number of the type to examine further: 1
   Type: TestClass
   Super class: Object
   Ancestors: [TestClass, Comparable, Object, Kernel, BasicObject]
   Included modules: [Comparable, Kernel]
   Constants: [:Constant1, :Constant2]
   Class variables: [:@@cvar1, :@@cvar2, :@@alphabet]
   Class methods: [:method1]
   Public instance methods: [:method2, :method3]
   Protected instance methods: [:method4, :method5]
   Private instance methods: [:method6, :method7]
Do you want to examine a type further (y or n)? n
Do you want to inspect another ruby source file (y or n)? n
Bye for now from Ruby!
```

Deliverables

1. Upload the file **Project5.rb** on Blackboard.
   - I will use the submission date/time on Blackboard as your official submission date/time.
   - It is your responsibility to make sure the submission on Blackboard went through successfully.
2. Late submission: if you submit it on Monday, April 11th, you will lose a total of 10 points for the delayed submission.