Laboratory: Io (2)

Summary: We continue our exploration of the Io programming language.

Prerequisites: The first Io lab. Section 3.3 of Tate.

Contents:

- Preparation
- Exercises
  - Exercise 1: Slot Names, Revisited
  - Exercise 2: Overriding Operators
  - Exercise 3: Alternate Averages
  - Exercise 4: More Averages
  - Exercise 5: Fun with Operators
- For Those with Extra Time
  - Extra 1: Overriding Division

Preparation

a. Create a directory for the lab.

b. Verify that Io is installed on the workstation you’re using. If not, you can ssh to church, where it’s installed.

Exercises

Exercise 1: Slot Names, Revisited

Io is a bit less friendly than Ruby in its provision of methods. For example, if I wanted to find what slots are available for an integer, I might write

```
Io> 0 slotNames
==> list()
```

As this suggests, the results are not necessarily as helpful as possible. Add a method, slots, that gives more useful results.

Exercise 2: Overriding Operators

Override the + operator so that it does subtraction rather than addition.
Exercise 3: Alternate Averages

The average message averages a list of numbers. But what if we have a heterogeneous list? Write a method, aveNums, that averages all numbers in a list. (You may want to explore the list methods to figure out how to extract parts of a list.)

This problem is based upon a problem from Tate.

Exercise 4: More Averages

Write a program that averages all the numbers in a two dimensional heterogeneous array.

This problem is based upon a problem from Tate.

Exercise 5: Fun with Operators

a. Add three operators, a, b, and c, with places 0, 1, and 2. E.g.,

Io> OperatorTable addOperator("a", 0)

b. Define them as follows.

Io> Number a := method(x, self * 2 + x)
Io> Number b := method(x, self + x * 2)
Io> Number c := method(x, self - x)

c. Predict the values of the following:

- 1 a 1
- 1 b 1
- 1 c 1
- 1 c 1 b 1 a 1

d. Check your answer experimentally.

e. Change the level of a to 2.

f. Predict the value of the following:

- 1 c 1 b 1 a 1

g. Check your answer experimentally.

For Those with Extra Time
Extra 1: Overriding Division

Right now, Io’s division operation returns $\infty$ when the divisor is 0. Rewrite it to return 0 if the divisor is 0.

This problem is taken from Tate. I will admit that I am not quite sure how to solve it.