Assignment 2: Smalltalk

20 marks

Preparing:

You may want a bit more experience with Smalltalk before diving into the assignment. The BankAccount tutorial, available at http://static.squeak.org/tutorials/BankAccount.html
is a good introductory tutorial that will help. *Note*: if you do complete the tutorial, **do not hand this portion in**. If you want a copy of your work, you can save the project or sources.

The Smalltalk-80 System class reading will also be useful when doing the assignment, and the following terse reference may also help: https://squeak.org/documentation/terse_guide/ . The course TAs will also be able to provide guidance if you get stuck.

Actual Assignment:

(1 mark) Create a new project in Squeak called **Matryoshka Project**.
Using the system browser, define a class category called **Matryoshka**.

(1 mark) Define a class in this category, called **Doll**, that has the following instance variables:

size (the size of the Doll)

nestedDoll (the Doll inside this Doll, or nil if empty)

(2 marks) Provide accessor methods for size and nestedDoll called **size** and **nestedDoll**, respectively.

(2 marks) implement a mutator for size called **size:** that sets the Doll's size to the provided argument value if and only if size is currently nil (unset), otherwise it ignores the message.

(2 marks) Define a constructor (a *class method*) **newSize:** that takes a single argument specifying the Doll's size and returns a new Doll object with that size.

(2 marks) Provide a **nest:** method, which receives another Doll as an argument. If the Doll to nest is smaller in size than this Doll, set the nestedDoll instance variable to refer to the Doll to nest. Ignore whether or not nestedDoll is nil.

(2 marks) Provide a method called **unstack**, which sends a message to nestedDoll telling it to unstack (if not nil), and removes the association with nestedDoll (by setting nestedDoll to nil).

(2 marks) Override the **Object asString** method such that it returns "Doll of size X[, …]" where X is the size of this Doll, and if nestedDoll is not nil it will append a comma, a space, and the result of calling nestedDoll: asString.

Hint: the comma (",") binary message is used to concatenate strings.

(4 marks) Define a *class method* **Doll nest:** that takes a Set of Dolls and nests them according to size, returning a reference to the outermost Doll. If more than one Doll of a given size is encountered during the nesting, the method will select one to nest and discard the other. If any of the Dolls in the Set already contain other Dolls, simply ignore and replace them.

Hint: it may be useful to sort the Set by Doll size before nesting the Dolls. The asSortedCollection: message can be sent to a Smalltalk collection (including a Set) in order to sort it by criteria that you include in a block as a message argument.

(2 marks): include scratch code (from the Workspace) that tests each of your methods. For Doll nest: the following scratch code **must be included** (you may include other test code as well)

```
someDolls := Set new.
aDoll := Doll newSize: 5.
someDolls add: aDoll.
someDolls add: (Doll newSize: 3).
someDolls add: (Doll newSize: 1).
someDolls add: (Doll newSize: 2).
someDolls add: (Doll newSize: 2).
someDolls add: (Doll newSize: 4).
Doll nest: someDolls.
Transcript show: aDoll, asString.
```

This **must output the following** in the Transcript window:
```
Doll of size 5, Doll of size 4, Doll of size 3, Doll of size 2,
Doll of size 1
```

Instructions for submitting your assignment:

Submit your assignment on Brightspace. You have two options:

1. Save the Squeak project and submit that, **or**
2. Save the Matryoshka category as source (fileOut) and copy the Workspace contents into a text file. Submit these as separate files or as a single compressed (or otherwise bundled) file.