Lab: SuperSorter (continued)

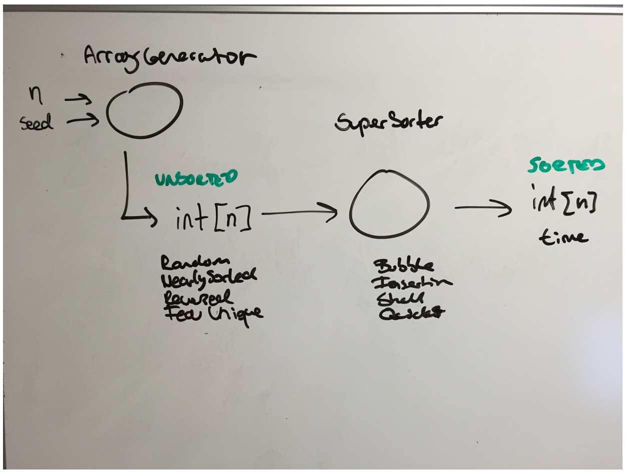
With these new creational patterns in your toolbox, go back to your solution for the SuperSorter from the last lesson and try to redesign it

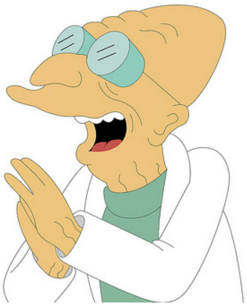
- Where does the factory patterns fit in (if the do)?

- Do they solve one or more problems you had in your previous solution?

Lab Exercise for SWD lecture 9: GoF Template Method & Strategy

Good news everyone! In this assignment you must help the professor with his crazy math experiments





In-between researching the Paradox-Free Time Travel Machine and the Parallel Universe Box Prof. Farnsworth has taken an interest to sorting stuff with various algorithms. He needs some software to help him run some experiments, and his software design skills are not up to scratch – this is where you come into the picture:

1) Create an ArrayGenerator which will take a size **n** (int) and a **seed** (int) and generate an array of randomized integers with values between 0 and **n**.

2) Create a SuperSorter which can take an **unsorted array** of integers and return a **sorted array** using one of the following example algorithms (feel free to pick your own selection): Bubble Sort, Insertion Sort, Shell Sort and Quick Sort*. With the professor, you know you can expect to have to include more algorithms at a later point. He would also like to get a measure of the **time** elapsed in milliseconds.

3) Realizing that the algortihms must be tested under different cases (best -> worst) the ArrayGenerator should be extended so that it can output the randomized array as either (again, maybe more cases in the future):

a) completely random: as in task 1
b) nearly sorted (from smallest to largest, with a few items out of order)
c) reverse order
d) with only a few unique values but in random order

4) Finally it would be nice to have some way of running comparative experiments, where he can just input a size **n** and a **seed** value, and the program will run all combinations of algorithms and sorting orders showing the performance in terms of time.

Hint: Use the GoF Strategy and/or GoF Template Method patterns to help professor F. with his problem

*) Algorithms widely available online, e.g.: http://anh.cs.luc.edu/170/notes/CSharpHtml/sorting.html