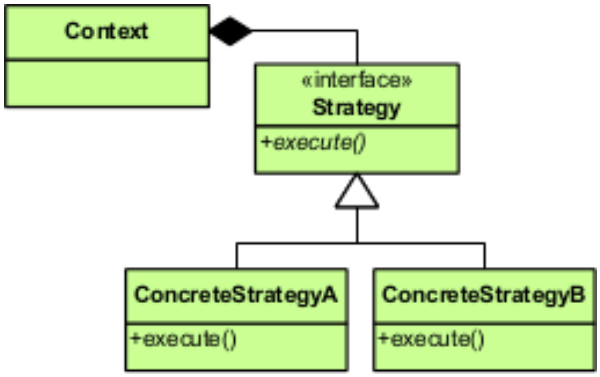
15-214 Homework 3

Name: Xiang Shu
Andrew ID: xshu

Design Pattern

In this program, I implement the strategy design pattern to realize the cryptarithm solving program. As the image illustrated below, there should be an interface to define the abstract permutation function, and one concrete class permutation to implement the generation and access of each permutation result in the recursive process (as with the Heap's algorithm). Whenever reach the base case in the heap's algorithm, the one possible permutation is set in the field, and could be accessed by the concrete class cryptarithm solver class which implements the abstract permutation interface. In this condition, each recursively derived permutation could be assigned to the cryptarithm letters and check the expression.



With the strategy design pattern, the program could save a lot of time of storing all the possible permutation since the permutation with large numbers could generate large numbers N! of results where N is the number of elements. And iteration of all possible permutation again is a waste of time.

Discussion

I have some problem in the access of each possible permutation. Since permutation generator is a general method which could not contains the information of cryptarithm, and the heap's algorithm is recursively implemented, I find it hard to get access of the one permutation, and check instantly. I think if heap's algorithm could be expressed in iterative process, and each permutation could be traced properly, the program would be some easier.