[ITERATOR PATTERN]

Employee Management Application

Nurjahan Akter & Alexandros Gkalkin

Introduction

The main goal of this document is to give an overview of the implementation of the iterator design pattern as well as the pattern itself.

Iterator Pattern

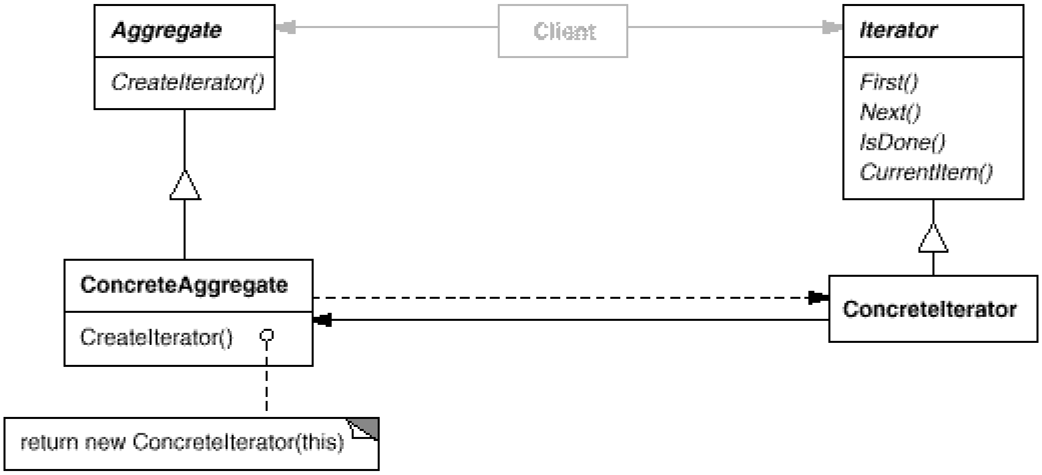


Figure 1 - Iterator pattern UML

The main goal of the iterator pattern to create a mechanism to cycle through a collection (either a list or an array, etc.) of objects. Although in modern day languages there are already implemented mechanism to access and cycle through object collections, using iterator pattern for these purposes makes the collection lighter through reducing additional collection functionalities. Furthermore, it defines a clear separation between the collection and the method how it is iterated.

Reusability

The reusability aspect of the iterator pattern is our application can be considered medium to high. Even though the mechanism of the iterator can be reused with medium to none addition to the code, the iterator is bound at the moment to an individual types of element collection. For example, we have a linked list in two cases and an array in one. Therefore, if we consider reusing the mechanism for the same types of collections then we do not have to alter the code except for the type of the elements.

Extensibility

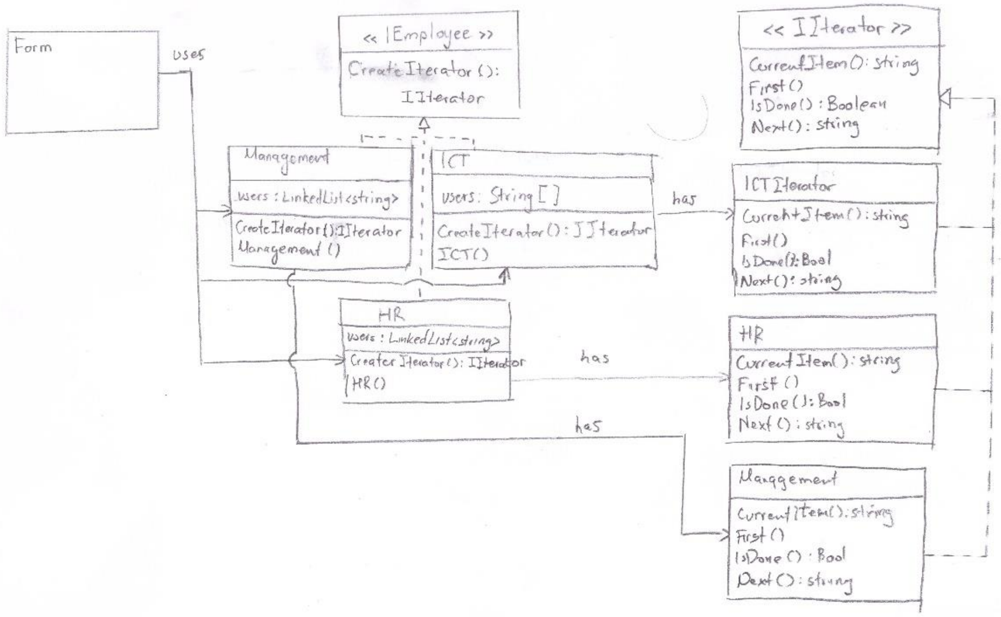
The iterator mechanism itself can be extended very simply by adding new iterators. Furthermore, we could simply extend the mechanism to the most used types of objects. Additionally, we could implement methods that could give use more control over the objects, such as iterate in the first or second half of the collection etc.

Maintainability

As far as maintainability is concerned we could describe it as an easy task. The iterators can use generic types of collections; therefore, we consider that we don’t really have trouble to maintain the current code.

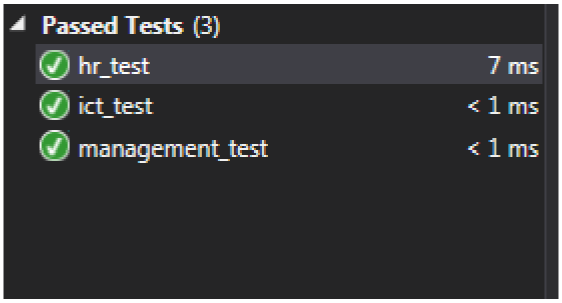
Furthermore, since we do not use if-else statements in the mechanism itself, there are no special cases that we need to worry about.

UML



Test

We have implemented our tests in a way that for every iterator we test if it circles through the objects correctly. We have done so for all the classes.



```
Passed Tests (3)
  hr_test            7 ms
  ict_test          < 1 ms
  management_test   < 1 ms
```

Video: https://youtu.be/UBVfioO1yGI