Sprawozdanie z ćwiczenia laboratoryjnego VII
Refaktoryzacja

Bartłomiej Ankowski

17.05.2015

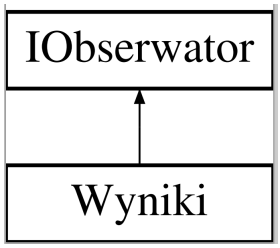
Spis treści

1 Wstęp

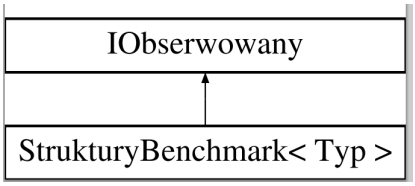
Celem tego laboratorium było zapoznanie się z zasadami projektowania oprogramowania obiektowego. W tym celu zapoznano nas z zasadami SOLID. Naszym zadaniem była refaktoryzacja wcześniej pisanych programów pod kątem wyżej wymienionych zasad.

2 Realizacja Obserwatora

Pierwszym zadaniem było zastosowanie wzorca projektowego jakim jest obserwator. W tym celu został zamodelowany Interfejs dla obiektów, które będą obserwowane i dla obiektow obserwujących. W ramach realizjacji tego podpunktu zostały stworzone dwie nowe klasy w stosunku do poprzedniej wersji programu.Ze starej klasy Benchmarkujacej zostały "wyciete"metody odpowiedzialne za przechowywanie wyników. Stworzono również klasę modelującą stoper.Spełniona została w tym momemcie zasada pojedynczej odpowiedzialności.



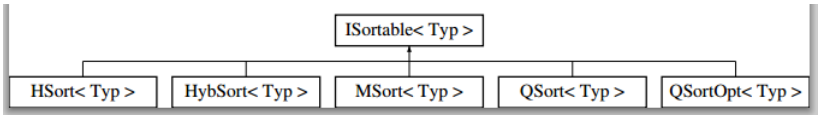
Rysunek 1: Obserwator



Rysunek 2: Obserwowany

3 Realizacja Interfejsu Sortowań

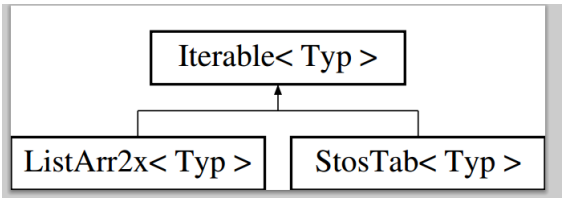
Kolejnym celem było umożliwienie zastosowania wcześniej zaimplementowanych algorytmów sortowań dla dowolnej kontenera danych. Wcześniejsza implementacja łamała zasadę segregowania interfejsów oraz Open/closed. W celu naprawy tego błedu został stworzony interfejs ISortable, który następnie jest dziedziczony do każdego wyodrębnionego algorytmu sortowania. Interfejs zawiera metodę Sort, która jako argument przyjmuje wskaźnik na kontener na którym ma zostać przeprowadzone sortowanie, co umożliwia obsługe dowolnego konteneru.



Rysunek 3: Sortowania

4 Realizacja Interfejsu Iterable

W celu umożliwienia algorytmom sortowania wglądu do pól kontenera, został stworzony interfejs Iterable, ktory jest dziedziczony przez zaimplementowane pojemniki danych. Jego zadaniem jest odczyt wartości jaka kryje sie pod danym indeksem oraz zwrócenie referencji w celu modyfikacji danego pola.



Rysunek 4: Iterable

5 Wnioski

- Zakładany cel został osiągniety.Wprowadzenie dodatkowych interfejsów umożliwiło przeprowadzenie sortowań dla zaimplementowanych kontenerów.Zmiany w kodzie zostały tak przeprowadzone, aby program był zaprojektowany zgodnie z zasadami SOLID.Warty odnotowania jest fakt, że ścisłe związanie algorytmu sortowania z kontenerem wystepuję w bibliotecę STL, w przypadku zaimplementowanej w niej liście. Zatem takie rozwiązanie również posiada swoje plusy, z pewnością jest nim wydajność. Niestety kosztem uniwersalności.

- Należy się również zastanowić nad zaimplementowanym interfejsem Benchmarku. W tej realizacji jest łamana zasada Open/closed, ponieważ w celu obslugi dowolnej struktury danych jest przechowywany wskaźnik na Interfejs jaki dana struktura posiada.Wskaźnik ten każdorazowo wymaga zmiany, jeśli testowana jest nowa struktura.
  Możliwe, że lepszy rozwiązaniem byłoby stworzenie jeszcze jednego interfejsu, np Itest. Posiadałby metodę test, która byłaby implementowana w testowanej strukturze. Wywołanie odbywałoby się poprzez wskaźnik na ten interfejs w Benchmarku.