Algorithms and Data Structures / List No. 2

Wroclaw University of Science and Technology - Faculty of Computer Science and Management
Marcin Stachowiak (marcin.stachowiak@pwr.edu.pl)

Keywords—lista jednokierunkowa, lista dwukierunkowa, tablica, węzeł, lista tablicowa, lista powiązana, iterator, realokacja pamięci

I. Zadanie 1 (Podstawowe struktury danych)

Zaimplementuj następujące struktury danych i wykonywane na nich operacje: **lista powiązana** i **lista tablicowa** [1] [2]. Zaimplementowane struktury przetestuj za pomocą testów jednostkowych.

- Klasa reprezentująca listę tablicową powinna nazywać się *CustomArrayList*, a listę powiązaną *CustomLinkedList*. Puste implementacje obu klas są już dostarczone w pakiecie *com.aisd.foundation.datastructure.list*. Klasy te implementują inferfejs *List*. Zaimplementować należy tylko wymagane metody, czyli te, które w klasie *AbstractCustomListAdapter* są abstrakcyjne.
- Należy zwrócić uwagę na sposób w jaki przechowywane będą dane. Do wykonania tego zadania nie należy posługiwać się gotowymi implementracjami interfejsu *List* dostarczonymi przez Jave. W zamian, w przypadku listy tablicowej należy użyć tablicy, a w przypadku listy powiązanej zbudować rekurencyjną konstrukcję z wykorzystaniem obiektu *com.asid.foundation.datastructure.list.Node*.
- Lista tablicowa powinna mieć wbudowany mechanizm realokacji, który zwiększy długość tablicy, gdy będzie ona zapełniona w 90% lub zmniejszy w przypadku zapełnienia poniżej 60%. Mechanizm relokacji powinien być uruchamiany na początku metod dokonujących zmian na tablicy. Wielkość o jaką lista będzie zwiększana i zmniejszana może być dowolna.
- Listę tablicową można utworzyć z długością przekazaną w konstruktorze. W przypadku jej braku powinna zostać użyta długość domyślna.
- Zarówno lista tablicowa jak i lista powiązana powinny mieć możliwość przechowywania obiektów o dowolnych typach, co można uzyskać dzięki zastosowaniu typów generycznych.
- Do obu struktur danych należy zaimplementować ich iteratory. Puste klasy interatorów zostały już przygotowane jako klasy prywatne należące do konkretnej listy.
- Podczas implementacji należy zadbać o odpowiednią obsługę warunków brzegowych np. usunięcie obiektu z pustej listy powinno zakończyć się błędem.
- Dla zaimplementowanych struktur danych należy przygotować odpowiednie testy jednostkowe. Przykład testu znajduje się w klasie *com.asid.foundation.datastructure.list.CustomArrayListTest*.

II. Pytania

1) Kiedy warto zastosować listę powiązaną a kiedy tablicową?
2) Co to jest iterator?
3) Jaka jest różnica pomiędzy kolejką FIFO, a stosem?
4) Algorytm udostępniający kolejne liczby Fibbonacciego mniejsze od podanego N.
5) Kolejki FIFO, LIFO oraz ich zastosowania.
6) Co to jest wartownik (w kontekście struktur danych)?

Literatura

[1] Robert Sedgewick and Kevin Wayne. *Algorithms, 4th Edition. [in Polish: Algorytmy]*. 2011. 1
[2] James Ross Simon Harris. *Algorytmy. Od podstaw*. 2006. 1