laboratorium: zadanie 11 dodatkowe *termin: 22, 24 stycznia 2013 r.*

KURS PROGRAMOWANIA W JAVIE

LISTA CYKLICZNA

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

Zadanie 1.

Zdefiniuj klasę sparametryzowaną do pamiętania zbioru dynamicznego w postaci jednokierunkowej listy cyklicznej `CyclicList<T>`. Klasa ta ma być opakowaniem dla homogenicznej struktury listowej tworzonej wewnątrz na obiektach typu `Node`. Twoja klasa powinna implementować operacje stosowe (dodawać element na początku listy `push` i usuwać element z początku listy `pop`) i kolejkowe (dodawać element na końcu listy `append` i usuwać element z początku listy `pop`).

```
public class CyclicList<T> implements MyStack<T>, MyQueue<t>
{
    private class Node { /*...*/ }

    private Node tail;

    public CyclicList () { /*...*/ }

    // ... metody push, pop, append i inne
    public String toString () { /*...*/ }
}
```

Przy próbie włożenia do listy wartości `null` należy zgłosić wyjątek `NullPointerException`. Dopisz też metody podające długość listy `size()` i usuwającej wszystkie elementy z listy `clear()`.

Zdefiniowana przez Ciebie klasa `CyclicList<T>` powinna prawidłowo działać w warunkach pracy współbieżnej, gdy wiele wątków jednocześnie pracuje na takiej liście. Jeśli jakiś wątek chce pobrać element z pustej listy należy go uśpić (metoda `wait()`). Wątek wkładający nowy element do listy ma za zadanie obudzić wątek czekający na jakiś element (metoda `notify()`).

Zadanie 2.

Do klasy reprezentującej listę cylkiczną dopisz metodę `iterator()`, która będzie tworzyła i zwracała iterator związany z daną listą. Zdefiniuj też klasę iteratora dedykowaną dla Twojej listy i implementującą interfejs `Iterator<T>`. Twój iterator powinien być wrażliwy na wszelkie zmiany w liście, po której iteruje — jeśli w trakcie iteracji po liście zostanie na niej dokonana jakakolwiek zmiana, to próba użycia nieaktualnego już iteratora powinna skutkować zgłoszeniem wyjątku `IllegalStateException`.

Zadanie 3.

Napisz program konsolowy, testujący działanie zdefiniowanej listy cyklicznej. Program ten ma symulować działanie *producentów i konsumentów* (uruchom kilkanaście/kilkadziesiąt wątków dla producentów i konsumentów). Prodeucenci i konsumenci mają wykonywać swoje działania w losowych odstępach czasu. Wszystkie produkty (na przykład liczby całkowite) mają być umieszczane w albo pobierane z jednej listy cyklicznej.

Każdy producent i konsument ma wyświetlić zawatość listy po wykonanej przez siebie pracy za pomocą niesynchronizowanej metody `toString()`, która korzysta z iteratora.