

INSTRUKCJA DO LAB 5.

To ćwiczenie bazuje na przykładach znajdujących się w 21 rozdziale książki „Thinking in Java wydanie IV” oraz w rozdziale 8 książki „Sztuka programowania wieloprocessorowego”.

Celem tych zadań jest zapoznanie się z konstrukcją monitora, stworzenie własnej wersji monitora i wykorzystanie go w rozwiązaniu konkretnego problemu.

Do tej instrukcji dołączone są dwa pliki z klasami:

LockedQueue.java, MutexEvenGenerator.java

ZADANIE 1

Blokady jawne – klasy implementujące interfejs Lock

1. Do projektu zawierającego klasę IntGenerator oraz EvenChecker dodać klasę MutexEvenGenerator (kod w dołączonym pliku).

Uruchomić testowanie MutexEvenGenerator przy pomocy EvenChecker.

Zdjąć komentarze z linii odwołujących się do klasy Lock.

(Klikając lpm+ctrl na nazwie Lock przyjrzeć się opisowi interfejsu).

Analogicznie przyjrzeć się interfejsowi Condition *(typ zwracanego obiektu ostatniej metody interfejsu Lock – zmienna warunkowa)*

ZADANIE 2.

1. Utworzyć projekt, który będzie zawierał klasę LockedQueue (kod w dołączonym pliku).

Zidentyfikować klasę LockedQueue i przeanalizować jej funkcjonalność.

Dopisać część kodu, który wykorzysta i przedstawi w komunikatach działanie powyższej klasy.

2. Wykorzystać kod LockedQueue do stworzenia klasy, która będzie monitorem zawierającym i kontrolującym dostęp do licznika kolejnych liczb naturalnych.

3. Kolejne liczby generowane przez monitor z poprzedniego punktu mają być przekazywane pracującym równolegle wątkom, sprawdzającym, czy dana liczba jest liczbą pierwszą.

Wyjaśnienie:

W zadaniu 2 funkcja generująca kolejne liczby naturalne ma być potraktowana jako „sekcja krytyczna”.