Sprawozdanie nr 4

Data ćwiczeń: 16.04.2015

Data oddania sprawozdania: 23.04.2015

* Tematyka

Na czwartych zajęciach laboratoryjnych ponownie poruszana była problematyka związana z zadaniem producentów i konsumentów.

* Zadanie

Na zajęciach należało zaimplementować jedno zadanie, które było modyfikacją zadania producenci i konsumenci ale z pewnymi narzuconymi odgórnie warunkami, które nieco zmieniały charakterystykę rozwiązania. Zadanie polegało na tym, aby każdy producent i każdy konsument wkładał, bądź wyciągał z/do bufora losową ilość danych. Przy rozwiązaniu należało skorzystać z Java Concurrency Utilities.

* Kod zadania 1

*/\*\*  
 \* Created by Patryk on 2015-04-16.  
 \*/***public class** Lab4Zad1 {  
 **public static void** main(String[] args) {  
 **int** M = 100;  
 **int** m = 1;  
 **int** n = 1;  
 Buffer buf = **new** Buffer(M);  
 ExecutorService service = Executors.*newFixedThreadPool*(m + n);  
  
 **for**(**int** i=1; i<=m; i++) {  
 service.submit(**new** Producer(buf));  
 }  
 **for**(**int** i=1; i<=n; i++) {  
 service.submit(**new** Consumer(buf));  
 }  
 service.shutdown();  
 }  
}

**public class** Producer **extends** Thread {  
 **private** Buffer **\_buf**;  
  
 **public** Producer(Buffer buf) {  
 **this**.**\_buf** = buf;  
 }  
  
 **public void** run() {  
 **int** a[] = **new int**[(**int**) (Math.*random*() \* 10 + 1)];  
 **for**(**int** i=0; i<a.**length**; i++) {  
 a[i] = (**int**)(Math.*random*() \* 10) + 1;  
 }  
  
 **while**(**true**) {  
 **\_buf**.put(a);  
 **try** {  
 *sleep*((**int**) (Math.*random*() \* 100));  
 } **catch** (InterruptedException e) {  
 System.***out***.println(e.getMessage());  
 }  
 }  
 }  
}

**public class** Consumer **extends** Thread {  
 **private** Buffer **\_buf**;  
 **public** Consumer(Buffer buf) {  
 **this**.**\_buf** = buf;  
 }  
 **public void** run() {  
 **while**(**true**) {  
 **\_buf**.get((**int**) (Math.*random*() \* 10 + 1));  
 **try** {  
 *sleep*((**int**) (Math.*random*() \* 100));  
 } **catch** (InterruptedException e) {  
 System.***out***.println(e.getMessage());  
 }  
 }  
 }  
}

**public class** Buffer {  
 **private** List<Integer> **buf** = **new** ArrayList<Integer>();  
 **private int M**;  
 **public** Buffer(**int** m) {  
 **this**.**M** = 2\*m;  
 }  
 **public synchronized void** put(**int**[] a) {  
 **while**(**M** - **buf**.size() < a.**length**) {  
 **try** {  
 System.***out***.println(**"\tProducer waits."**);  
 wait();  
 } **catch**(InterruptedException iexp) {  
 System.***out***.println(iexp.getMessage());  
 }  
 }  
 **for**(**int** j=0; j<a.**length**; j++) {  
 **buf**.add(a[j]);  
 System.***out***.println(**"Producer puts "** + a[j]);  
 }  
 notify();  
 }  
 **public synchronized int** get(**int** a) {  
 **while**(**buf**.size() < a) {  
 **try** {  
 System.***out***.println(**"\tConsumer waits."**);  
 wait();  
 } **catch**(InterruptedException iexp) {  
 System.***out***.println(iexp.getMessage());  
 }  
 }  
  
 **int** returnVal = 0;  
 **for**(**int** i = 0; i < a; i++) {  
 **int** index = **new** Random().nextInt(**buf**.size());  
 returnVal = **buf**.get(index);  
 **buf**.remove(index);  
 notify();  
 System.***out***.println(**"Consumer gets "** + returnVal);  
 }  
 **return** returnVal;  
 }  
}