Sprawozdanie nr 6

Data ćwiczeń: 30.04.2015

Data oddania sprawozdania: 28.05.2015

* Tematyka

Na szóstych zajęciach laboratoryjnych poruszana była nowa problematyka zwana "Problemem filozofów".

* Zadanie

Na zajęciach należało zaimplementować kilka programów. Pierwszym z nich była wersja niedziałająca, w której dochodzi do sytuacji zwanej zakleszczeniem wątków. Na wklejonym output'cie z konsoli widać, że każdy z filozofów trzyma lewy widelec w ręce czekając na dostęp do prawego. Drugie zadanie polegało na naprawie tego problemu kiedy oba widelce podnoszone i odkładane są jednocześnie. W wariancie numer trzy należało zaimplementować rozwiązanie z lokajem. No i na koniec wykonać pomiary dla każdego z tych rozwiązań oraz krótko skomentować.

* Kod zadania 2

**package** lab6.zad2;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** Lab6Zad2 {  
 **public static void** main(String args[]){  
 List<Widelec> widelce = **new** ArrayList<Widelec>();  
 Filozof[] filozofs = **new** Filozof[5];  
  
 Object lockA = **new** Object();  
  
 **for**(**int** i = 0; i < 5; i++){  
 widelce.add(**new** Widelec());  
 filozofs[i]=**new** Filozof(i, widelce, lockA);  
 }  
  
 **for**(**int** i = 0; i < 5; i++) {  
 filozofs[i].start();  
 }  
  
 **for**(**int** i = 0; i < 5; i++){  
 **try** {  
 filozofs[i].join();  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

**package** lab6.zad2;  
  
**import** java.util.List;  
  
**public class** Filozof **extends** Thread {  
 **private int \_licznik** = 0;  
 **private int num**;  
 **private** List<Widelec> **widelce**;  
 **private final** Object **lockObject**;  
  
 **public** Filozof(**int** num,List<Widelec> widelce, Object lock) {  
 **lockObject** = lock;  
 **this**.**num** = num;  
 **this**.**widelce** = widelce;  
 }  
  
 **public void** run() {  
 **long** startTime = System.*currentTimeMillis*();  
 **int** lewyWidelec = ((**num** == 0) ? 4 : **num**-1);  
 **int** prawyWidelec = ((**num** == 4) ? 0 : **num**+1);  
   
 **while**(**true**) {  
  
 **synchronized** (**lockObject**) {  
 **widelce**.get(lewyWidelec).podnies();  
 **widelce**.get(prawyWidelec).podnies();  
 }  
  
 ++**\_licznik**;  
 **widelce**.get(lewyWidelec).odloz();  
 **widelce**.get(prawyWidelec).odloz();  
  
 **if** (**\_licznik** == 1000000) {  
 System.***out***.println(**"Filozof: "** + Thread.*currentThread*() + **"jadlem "** + **\_licznik** + **" razy"** + **" czas:"** + (System.*currentTimeMillis*() - startTime));  
 **break**;  
 }  
  
 }  
 }  
}

**package** lab6.zad2;  
  
**public class** Widelec {  
 **private boolean state**;  
  
 **public** Widelec(){  
 **this**.**state** = **true**;  
 }  
  
 **public synchronized void** podnies() {  
 **while**(!**state**){  
 **try** {  
 wait();  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
  
 **state**=**false**;  
 }  
  
 **public synchronized void** odloz() {  
 **state**=**true**;  
 notifyAll();  
 }  
  
 **public boolean** zajety(){  
 **return state**;  
 }  
}

* Kod zadania 3

**package** lab6.zad3;  
  
*/\*\*  
 \* Created by Patryk on 2015-05-22.  
 \*/***public class** Lab6Zad3 {  
 **public static void** main(String[] args) {  
 Filozof[] f = **new** Filozof[5];  
 Lokaj lokaj = **new** Lokaj();  
  
 **for**(**int** i = 0; i < 5; i++) {  
 f[i] = **new** Filozof(i, lokaj);  
 }  
  
 **for**(**int** i = 0; i < 5; i++) {  
 f[i].start();  
 }  
  
 **for**(**int** i = 0; i < 5; i++){  
 **try** {  
 f[i].join();  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

**package** lab6.zad3;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
*/\*\*  
 \* Created by Patryk on 2015-05-28.  
 \*/***public class** Lokaj {  
 **private** List<Widelec> **stol** = **new** ArrayList<Widelec>();  
 **private int liczbaZajetychWidelcow** = 0;  
  
 **public** Lokaj() {  
 **for** (**int** i = 0; i < 5; i++) {  
 **stol**.add(**new** Widelec());  
 }  
 }  
  
 **public synchronized void** podajWidelce(**int** num1, **int** num2){  
 **while**(!**stol**.get(num1).zajety() || !**stol**.get(num2).zajety()){  
 **try** {  
 wait();  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
  
 **while**(**liczbaZajetychWidelcow** == 4){  
 **try** {  
 wait();  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
  
 **liczbaZajetychWidelcow** += 2;  
 **stol**.get(num1).podnies();  
 **stol**.get(num2).odloz();  
 }  
  
 **public synchronized void** zabierzWidelce(**int** num1, **int** num2){  
 **liczbaZajetychWidelcow** -= 2;  
 **stol**.get(num1).odloz();  
 **stol**.get(num2).odloz();  
 notifyAll();  
 }  
}

**package** lab6.zad3;  
  
**public class** Widelec {  
 **private boolean state**;  
  
 **public** Widelec(){  
 **this**.**state** = **true**;  
 }  
  
 **public void** podnies() {  
 **state**=**false**;  
 }  
  
 **public void** odloz() {  
 **state**=**true**;  
 }  
  
 **public boolean** zajety(){  
 **return state**;  
 }  
}

**package** lab6.zad3;  
  
**public class** Filozof **extends** Thread {  
 **private int \_licznik** = 0;  
 **private int num**;  
 **private** Lokaj **lokaj**;  
  
 **public** Filozof(**int** num, Lokaj lokaj){  
 **this**.**num** = num;  
 **this**.**lokaj** = lokaj;  
 }  
  
 **public void** run() {  
 **long** startTime = System.*currentTimeMillis*();  
  
 **int** lewyWidelec = ((**num** == 0) ? 4 : **num**-1);  
 **int** prawyWidelec = ((**num** == 4) ? 0 : **num**+1);  
  
 **while**(**true**) {  
 **lokaj**.podajWidelce(lewyWidelec, prawyWidelec);  
 ++**\_licznik**;  
 **lokaj**.zabierzWidelce(lewyWidelec, prawyWidelec);  
  
 **if** (**\_licznik** == 1000000) {  
 System.***out***.println(**"Filozof: "** + Thread.*currentThread*() + **"jadlem "** + **\_licznik** + **" razy"** + **" czas:"** + (System.*currentTimeMillis*() - startTime));  
 **break**;  
 }  
 }  
 }  
}