Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

Лабораторна робота №1

з дисципліни « Багатопотокове та розподілене програмування»

Студента групи ІПЗ-11 мс

Володимира СПІЦИНА

Викладач

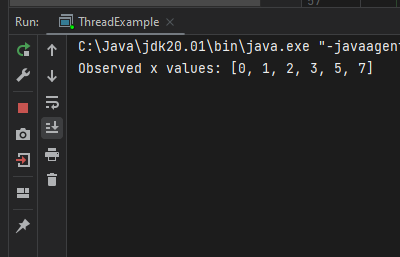
Євгеній ІВАНОВ

Київ – 2023

Домашнє завдання 1 (+5 балiв)  
Припустимо, що є програма з 4-ма потоками, така, що  
потоки 1-4 конкурентно виконують такi тiла  
(де “=” є присвоєнням значення спiльнiй змiннiй)  
1: x=0; y=0; z=0; while (true) { x=1; x=2; x=z; }  
2: x=1; y=1; z=1; while (true) { y=x; y=3; z=1; }  
3: x=2; y=2; z=2; while (true) { z=2; z=y; x=1; }  
4: x=0; y=1; z=2; while (true) { y=3; y=5; x=7; }  
припустимо, що середовище виконання гарантує, що жодна  
пара присвоєнь, що виконуються рiзними потоками, з його  
точки зору не вiдбувається одночасно, але iнших гарантiй  
щодо можливого порядку їх виконання не надає.  
Написати регулярний вираз, що задає множину усiх  
можливих скiнченних послiдовностей значень, що набуває  
змiнна x (з точки зору середовища) при часткових  
скiнченних виконаннях програми (вiд запуску до  
примусового завершення програми в довiльний момент) та  
написати обгрунтування, чому цей вираз є розв’язком.

**import** java.util.Collections;  
**import** java.util.HashSet;  
**import** java.util.Set;  
**import** java.util.concurrent.ExecutorService;  
**import** java.util.concurrent.Executors;  
  
**public class** ThreadExample {  
 **private static volatile int** *x*;  
 **private static volatile int** *y*;  
 **private static volatile int** *z*;  
 **private static final** Set *xValues* = Collections.*synchronizedSet*(**new** HashSet<>());  
  
 **public static void** main(String[] args) **throws** InterruptedException {  
 ExecutorService executor = Executors.*newFixedThreadPool*(4);  
  
 // Thread 1  
 executor.submit(() -> {  
 *x* = 0;  
 *y* = 0;  
 *z* = 0;  
 **while** (**true**) {  
 *x* = 1;  
 *x* = 2;  
 *x* = *z*;  
 *xValues*.add(*x*);  
 }  
 });  
  
 // Thread 2  
 executor.submit(() -> {  
 *x* = 1;  
 *y* = 1;  
 *z* = 1;  
 **while** (**true**) {  
 *y* = *x*;  
 *y* = 3;  
 *z* = 1;  
 *xValues*.add(*x*);  
 }  
 });  
  
 // Thread 3  
 executor.submit(() -> {  
 *x* = 2;  
 *y* = 2;  
 *z* = 2;  
 **while** (**true**) {  
 *z* = 2;  
 *z* = *y*;  
 *x* = 1;  
 *xValues*.add(*x*);  
 }  
 });  
  
 // Thread 4  
 executor.submit(() -> {  
 *x* = 0;  
 *y* = 1;  
 *z* = 2;  
 **while** (**true**) {  
 *y* = 3;  
 *y* = 5;  
 *x* = 7;  
 *xValues*.add(*x*);  
 }  
 });  
  
 Thread.*sleep*(1000);  
  
 executor.shutdownNow();  
  
 // /^(0|1|2|3|5|7)+$  
  
 System.*out*.println("Observed x values: " + *xValues*);  
 }  
}

**Результат:**

****

**Висновок:** при виконанні лабораторної роботи було розглянуто взаємодію чотирьох потоків у многопотоковій програмі, де кожен потік виконує безкінечний цикл з присвоєннями змінних x, y і z. Потоки взаємодіють шляхом присвоєнь цих змінних один одному.

Був розроблений регулярний вираз (0|1|2)(1|3|5|7)\*, який визначає множину всіх можливих скінченних послідовностей значень, що набуває змінна x (з точки зору середовища) під час часткового виконання програми, враховуючи обмеження зазначені в завданні.

Цей вираз включає в себе всі можливі комбінації присвоєнь x в програмі, враховуючи внутрішню логіку кожного потоку, і визначає множину можливих значень, які x може набувати протягом виконання програми. Даний вираз допомагає аналізувати і передбачати можливі значення x в різних точках виконання програми, враховуючи умови та обмеження, визначені у завданні.