

Дисципліна: Технології паралельних обчислень

ПІБ студента: Спаських Микола Дмитрович

Група ІІІ-12

Викладач: Дифучин Антон Юрійович

Лабораторна робота 3

Тема: Завдання до комп'ютерного практикуму 3 «Розробка паралельних програм з використанням механізмів синхронізації: синхронізовані методи, локери, спеціальні типи»

Завдання 1: Реалізуйте програмний код, даний у лістингу, та протестуйте його при різних значеннях параметрів. Модифікуйте програму, використовуючи методи управління потоками, так, щоб її робота була завжди коректною. Запропонуйте три різних варіанти управління.

Синхронізований метод:

```
public synchronized void sync_method_transfer(int from, int to, int amount) {  
    accounts[from] -= amount;  
    accounts[to] += amount;  
    ntransacts++;  
    if (ntransacts % NTEST == 0)  
        test();  
}
```

Синхронізований блок:

```
1 usage  
private Object obj = new Object();
```

```
no usages new *  
public void sync_block_transfer(int from, int to, int amount) {  
    synchronized (obj) {  
        accounts[from] -= amount;  
        accounts[to] += amount;  
        ntransacts++;  
        if (ntransacts % NTEST == 0)  
            test();  
    }  
}
```

Рентрантлок:

```
/.../  
import java.util.concurrent.locks.ReentrantLock;  
new *
```

```
2 usages  
ReentrantLock locker = new ReentrantLock();
```

```

no usages new *
public synchronized void sync_reentrantlock_transfer(int from, int to, int amount) {
    locker.lock();
    accounts[from] -= amount;
    accounts[to] += amount;
    ntransacts++;
    if (ntransacts % NTEST == 0)
        test();
    locker.unlock();
}

```

Завдання 2: Реалізуйте приклад Producer-Consumer application

(див. <https://docs.oracle.com/javase/tutorial/essential/concurrency/guardmeth.html>).

Модифікуйте масив даних цієї програми, які читаються, у масив чисел заданого розміру (100, 1000 або 5000) та протестуйте програму. Зробіть висновок про правильність роботи програми.

Код виконання:

```

package p2;

import java.util.Random;

public class Consumer implements Runnable {
    private Drop drop;
    private int N;

    public Consumer(Drop drop, int n) {
        this.drop = drop;
        this.N=n;
    }

    public void run() {
        Random random = new Random();
        for (int message = drop.take(); message!=-1; message = drop.take()) {
            System.out.format("MESSAGE RECEIVED: %d\n", message);
            try {
                Thread.sleep(random.nextInt(5));
            } catch (InterruptedException e) {}
        }
    }
}

package p2;

public class Drop {
    // Повідомлення, відправлене від виробника
    // до споживача.
    private int message;
    // true, якщо споживач повинен чекати,
    // поки виробник надішле повідомлення,
    // false, якщо виробник повинен чекати,
    // поки споживач отримає повідомлення.
    private boolean empty = true;
}

```

```

    public synchronized int take() {
        // Чекаємо, поки повідомлення буде доступне.
        while (empty) {
            try {
                wait();
            } catch (InterruptedException e) {}
        }
        // Змінюємо статус.
        empty = true;
        // Повідомляємо виробника, що
        // статус змінився.
        notifyAll();
        return message;
    }

    public synchronized void put(int message) {
        // Чекаємо, поки повідомлення буде отримане.
        while (!empty) {
            try {
                wait();
            } catch (InterruptedException e) {}
        }
        // Змінюємо статус.
        empty = false;
        // Зберігаємо повідомлення.
        this.message = message;
        // Повідомляємо споживача, що статус
        // змінився.
        notifyAll();
    }
}

package p2;

import java.util.Random;

public class Producer implements Runnable {
    private Drop drop;
    private int[] mas;
    private int N;

    public Producer(Drop drop, int n) {
        this.drop = drop;
        this.N=n;
        mas = new int[N];
        for (int i = 0; i<N; i++){
            mas[i] = i;
        }
    }

    public void run() {
        Random random = new Random();

        for (int i = 0; i < mas.length; i++) {
            drop.put(mas[i]);
            try {
                Thread.sleep(random.nextInt(5));
            } catch (InterruptedException e) {}
        }
        drop.put(-1);
    }
}

package p2;

public class ProducerConsumerExample {

```

```

public static void main(String[] args) {
    int N=100; //1000, 10000
    Drop drop = new Drop();
    (new Thread(new Producer(drop, N))).start();
    (new Thread(new Consumer(drop, N))).start();
}
}

```

Результат виконання (для 1000):

```

Run ProducerConsumerExample x
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:D:\programs\IntelliJ ID
MESSAGE RECEIVED: 0
MESSAGE RECEIVED: 1
MESSAGE RECEIVED: 2
MESSAGE RECEIVED: 3
MESSAGE RECEIVED: 4
MESSAGE RECEIVED: 5
MESSAGE RECEIVED: 6

```

```

MESSAGE RECEIVED: 996
MESSAGE RECEIVED: 997
MESSAGE RECEIVED: 998
MESSAGE RECEIVED: 999

Process finished with exit code 0

```

Програма спрацювала правильно.

Завдання 3: Реалізуйте роботу електронного журналу групи, в якому зберігаються оцінки з однієї дисципліни трьох груп студентів. Кожного тижня лектор і його 3 асистенти виставляють оцінки з дисципліни за 100-бальною шкалою.

Код:

```

package p3;

public class jornal {
    public String [][][] mas = new String [3][20][12][2]; // група, студент,
    тиждень, лектор чи асистент

    public synchronized void put (int group, int student, int week, int lector,
    String grade){
        mas[group][student][week][lector]=grade;
    }
    public void print() {
        for(int gro = 0; gro < 3; gro ++){
            System.out.print("\n Group "+gro+":\n");
            for (int st = 0; st < 20; st++){
                System.out.print("\n Student "+st+":\n");
                for (int l=0; l<2; l++){
                    if (l==0) {

```

```

        System.out.print("As: ");
    }
    else System.out.print("\nLe: ");
    for (int w = 0; w < 12; w++) {
        System.out.print(mas[gro][st][w][1]+" ");
    }
    }
}

package p3;
public class main {
    public static void main(String[] args) throws InterruptedException {
        jornal j = new jornal();
        Thread[] threads = new Thread[4]; // Масив для зберігання посилань на
створені потоки
        Runnable r = () -> {
            threads[0] = new Thread(new producer(j, "Lecturer 1", 1, new
int[]{0,1,2}));
            threads[1] = new Thread(new producer(j, "Assistant 1", 0, new
int[]{0}));
            threads[2] = new Thread(new producer(j, "Assistant 2", 0, new
int[]{1}));
            threads[3] = new Thread(new producer(j, "Assistant 3", 0, new
int[]{2}));

            for (Thread t : threads) {
                t.start(); // Запускаємо кожен потік
            }
        };

        Thread t = new Thread(r);
        t.start();
        t.join(); // Очікуємо завершення основного потоку

        try {
            for (Thread thread : threads) {
                thread.join(); // Очікуємо завершення кожного потоку
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        j.print();
    }
}

package p3;
import java.util.Random;
public class producer implements Runnable{
    private jornal j;
    private String name;
    private int[] groups;
    private int lector;
    public producer(journal j, String who, int lector, int[] g){
        this.j=j;
        name = who;
        groups = g;
        this.lector=lector;
    }
    public void run () {
        Random random = new Random();
        for (int g : groups){

```

```

        for(int stud=0; stud<20; stud++){
            for(int w=0; w<12; w++){
                j.put(g, stud, w, lector, random.nextInt(100) + " (" + name+
");");
            }
        }
    }
}
}
}

```

Результат виконання:

```

Run main
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:D:\programs\IntelliJ IDEA 2023.3.4\lib\idea_rt.jar=1501:D:\programs\IntelliJ IDEA 2023.3.4\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8

Group 0:

Student 0:
As: 25 (Assistant 1); 24 (Assistant 1); 76 (Assistant 1); 45 (Assistant 1); 57 (Assistant 1); 44 (Assistant 1); 88 (Assistant 1); 88 (Assistant 1); 46 (Assistant 1); 24 (Assistant 1); 39 (Assistant 1); 32 (Assistant 1);
Le: 87 (Lecturer 1); 25 (Lecturer 1); 84 (Lecturer 1); 8 (Lecturer 1); 29 (Lecturer 1); 65 (Lecturer 1); 23 (Lecturer 1); 77 (Lecturer 1); 86 (Lecturer 1); 94 (Lecturer 1); 75 (Lecturer 1); 10 (Lecturer 1);
Student 1:
As: 65 (Assistant 1); 20 (Assistant 1); 94 (Assistant 1); 42 (Assistant 1); 57 (Assistant 1); 90 (Assistant 1); 32 (Assistant 1); 95 (Assistant 1); 69 (Assistant 1); 61 (Assistant 1); 36 (Assistant 1); 43 (Assistant 1);
Le: 59 (Lecturer 1); 92 (Lecturer 1); 35 (Lecturer 1); 60 (Lecturer 1); 21 (Lecturer 1); 94 (Lecturer 1); 73 (Lecturer 1); 33 (Lecturer 1); 51 (Lecturer 1); 47 (Lecturer 1); 75 (Lecturer 1); 47 (Lecturer 1);
Student 2:
As: 8 (Assistant 1); 82 (Assistant 1); 96 (Assistant 1); 86 (Assistant 1); 74 (Assistant 1); 11 (Assistant 1); 36 (Assistant 1); 70 (Assistant 1); 40 (Assistant 1); 74 (Assistant 1); 26 (Assistant 1); 11 (Assistant 1);
Le: 42 (Lecturer 1); 95 (Lecturer 1); 63 (Lecturer 1); 99 (Lecturer 1); 11 (Lecturer 1); 60 (Lecturer 1); 82 (Lecturer 1); 54 (Lecturer 1); 87 (Lecturer 1); 74 (Lecturer 1); 28 (Lecturer 1); 4 (Lecturer 1);
Student 3:
As: 75 (Assistant 3); 51 (Assistant 3); 60 (Assistant 3); 21 (Assistant 3); 40 (Assistant 3); 71 (Assistant 3); 21 (Assistant 3); 40 (Assistant 3); 60 (Assistant 3); 60 (Assistant 3); 60 (Assistant 3); 60 (Assistant 3);
Le: 49 (Lecturer 1); 0 (Lecturer 1); 12 (Lecturer 1); 46 (Lecturer 1); 29 (Lecturer 1); 97 (Lecturer 1); 20 (Lecturer 1); 27 (Lecturer 1); 70 (Lecturer 1); 52 (Lecturer 1); 31 (Lecturer 1); 24 (Lecturer 1);
Student 18:
As: 37 (Assistant 3); 97 (Assistant 3); 45 (Assistant 3); 55 (Assistant 3); 20 (Assistant 3); 71 (Assistant 3); 22 (Assistant 3); 51 (Assistant 3); 11 (Assistant 3); 21 (Assistant 3); 53 (Assistant 3); 64 (Assistant 3);
Le: 53 (Lecturer 1); 88 (Lecturer 1); 71 (Lecturer 1); 87 (Lecturer 1); 5 (Lecturer 1); 48 (Lecturer 1); 32 (Lecturer 1); 22 (Lecturer 1); 8 (Lecturer 1); 60 (Lecturer 1); 8 (Lecturer 1); 30 (Lecturer 1);
Student 19:
As: 13 (Assistant 3); 71 (Assistant 3); 99 (Assistant 3); 15 (Assistant 3); 86 (Assistant 3); 9 (Assistant 3); 31 (Assistant 3); 22 (Assistant 3); 33 (Assistant 3); 90 (Assistant 3); 37 (Assistant 3); 93 (Assistant 3);
Le: 56 (Lecturer 1); 25 (Lecturer 1); 75 (Lecturer 1); 45 (Lecturer 1); 86 (Lecturer 1); 75 (Lecturer 1); 49 (Lecturer 1); 22 (Lecturer 1); 75 (Lecturer 1); 85 (Lecturer 1); 84 (Lecturer 1); 43 (Lecturer 1);
Process finished with exit code 0

```