**Прізвище:** Тарганський

**Ім'я:** Тарас

**Група:** КН-315

**Варіант:** 12

**Дата захисту:**06.03.2019р.

**Кафедра:** САПР

**Дисципліна:** Технології розподілених систем і

паралельних обчислень

**Перевірив:** Фарамага І.В.

**ЗВІТ**

до лабораторної роботи № 1

на тему "*Багатопотоковість*"

**Мета роботи**

Одержати навики побудови багатопотокових алгоритмів обчислювальних

процесів, навчитися складати програми з розгалуженими та взаємодіючими

процесами на мові Java.

**Індивідуальне завдання**

Програма моделює обслуговування одного потоку процесів

одним центральним процесором комп’ютера без черги. Якщо черговий процес

генерується в мить, коли процесор вільний, процес поступає на обробку в

процесор, інакше процес знищується. Визначте відсоток знищених процесів.

**Текст (код) програми**

class Main{

private static int n = 0;

public static void main(String[] args) {

Exercise exercise = new Exercise();

for (int i = 0; i < 10; i++) {

Process process = new Process(exercise, i, (int)(Math.random()\*10));

try {

Thread.sleep(3000);

} catch (InterruptedException e) { e.printStackTrace(); }

}

System.out.println("percent of sucseffuly process = " + (n\*10) + "%");

}

public static void count(){ n++; }

}

public class Process implements Runnable{

private Exercise exercise;

private Thread thread;

private int name;

private int number;

public Process(Exercise exercise, int name, int number) {

this.name = name;

this.number = number;

this.exercise = exercise;

thread = new Thread(this);

thread.start();

}

@Override

public void run() { exercise.doExercise(name, number); }

}

public class Exercise {

private boolean isAviable;

public Exercise() { this.isAviable = true; }

synchronized public void doExercise(int name, int number) {

if (isAviable) {

Main.count();

isAviable = false;

System.out.println("process number " + name + "(" + number + ")" + " start");

for (int i = 0; i < number; i++) {

try {

Thread.sleep(1000);

wait(1);

} catch (InterruptedException e) { e.printStackTrace(); }

}

System.out.println("process number " + name + "(" + number + ")" + " finish");

isAviable = true;

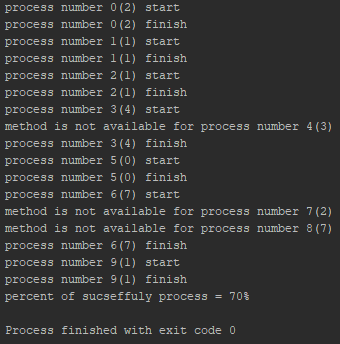
} else

System.out.println("method is not available for process number " + name + "(" + number + ")");

}

}

**Результати виконання програми**

****

**Висновок**

Одержав навики побудови багатопотокових алгоритмів обчислювальних

процесів, навчився складати програми з розгалуженими та взаємодіючими

процесами на мові Java.