

Engenharia de Software

Software Concorrente e Distribuído

Universidade Federal de Goiás

Marcus Mitra Muniz Inácio - 201508741

Exercício 03:

Será possível se a thread atualizar ser parada exatamente após executar sua primeira linha(quando ela pega o valor do saldo da conta e faz o cálculo mas ainda não atribui ao saldo da conta) e antes de executar a segunda linha(quando ela atribui o valor calculado ao saldo da conta).

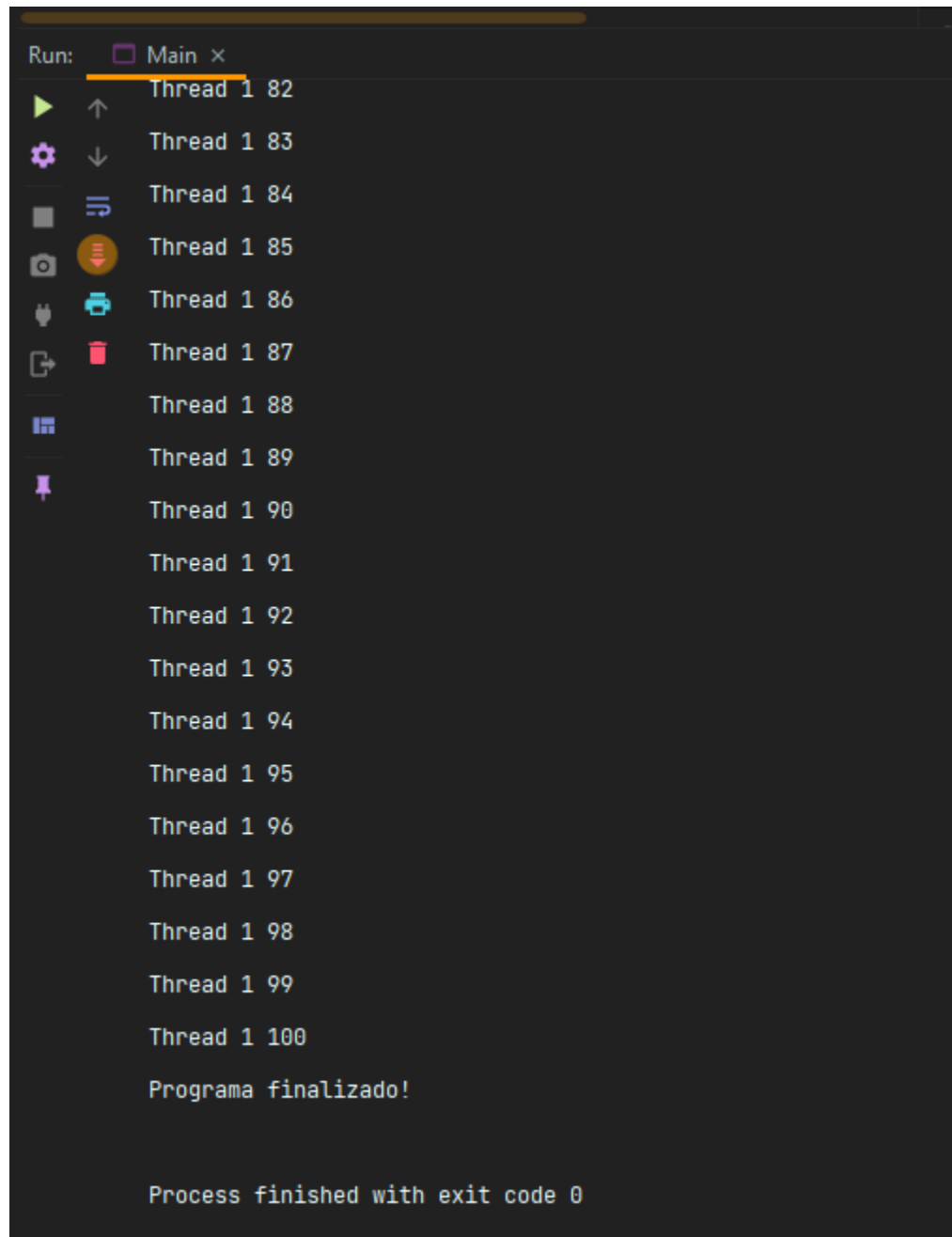
A sugestão seria utilizar semáforos ou algum outro tipo de controlador para garantir a execução completa de uma thread antes de permitir que outra tome seu lugar.

Irei incluir “prints” do funcionamento e dos códigos dos Exercícios 01 e 02 além das pastas de projeto:

Exercício 01:

```
1 package com.ufg;
2
3 public class ThreadOne extends Thread{
4     private String threadName;
5
6     public ThreadOne(String threadName){
7         this.threadName = threadName;
8     }
9
10    public void run(){
11        for (int i=1; i ≤ 100; i++){
12            System.out.println(threadName + " " + i);
13        }
14        System.out.println("Programa finalizado!");
15    }
16 }
17
```

```
1 package com.ufg;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         ThreadOne threadOne = new ThreadOne( threadName: "Thread 1");
7         threadOne.start();
8     }
9 }
10
```



Exercício 02:

```
Main.java x ThreadFruit.java x
1 package com.ufg;
2
3 import ...
4
5
6 public class Main {
7
8     private static final List<ThreadFruit> threadFruits = new ArrayList<>();
9
10    public static void runThreads() throws InterruptedException {
11        threadFruits.forEach(Thread::start);
12
13        Thread.sleep( millis: 3000);
14
15        threadFruits.forEach(Thread::interrupt);
16
17        System.out.println(threadFruits.size() + " threads stopped");
18    }
19
20    public static void addNewThread(String fruitName) {
21        threadFruits.add(new ThreadFruit(fruitName));
22    }
23
24    public static void main(String[] args) throws InterruptedException {
25        addNewThread( fruitName: "Maçã");
26        addNewThread( fruitName: "Banana");
27        addNewThread( fruitName: "Pera");
28        addNewThread( fruitName: "Abacate");
29        addNewThread( fruitName: "Kiwi");
30
31        runThreads();
32    }
33
34 }
35
```

```
1 package com.ufg;
2
3 public class ThreadFruit extends Thread{
4     private String fruitName;
5
6     public ThreadFruit(String fruitName) { this.fruitName = fruitName; }
7
8     @Override
9     public void run() {
10         while (!Thread.interrupted()){
11             System.out.println(this.fruitName);
12
13             try {
14                 Thread.sleep( millis: 500);
15             } catch (Exception e) {
16                 break;
17             }
18         }
19     }
20 }
```

