

Engenharia de Software

Software Concorrente e Distribuído

Universidade Federal de Goiás

Marcus Mitra Muniz Inácio - 201508741

Atividade “Lab Threads”:

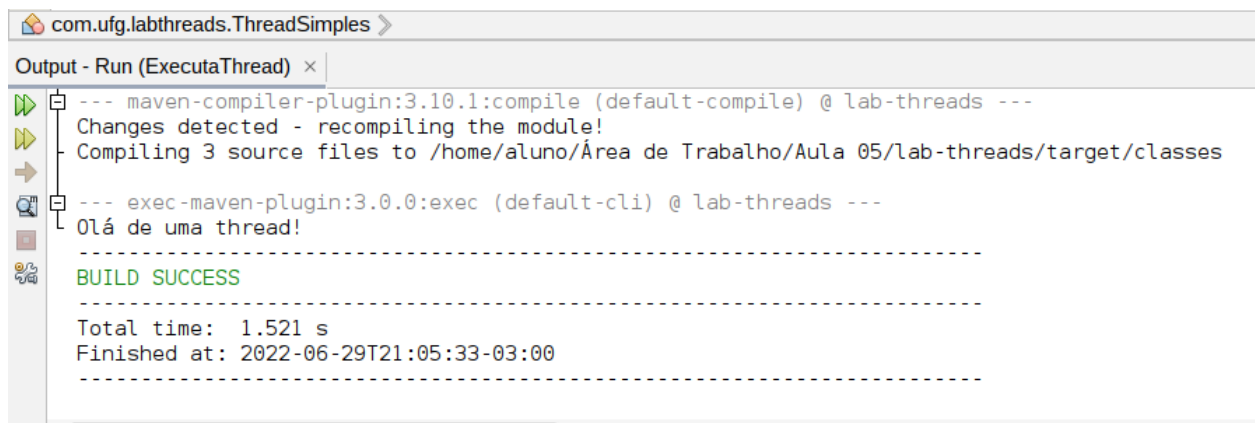
Exercício 01:



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
4   */
5   package com.ufg.labthreads;
6
7   /**
8    *
9    * @author aluno
10   */
11   public class ExecutaThread {
12   public static void main(String args[]) {
13       ThreadSimples simples = new ThreadSimples();
14       Thread thread = new Thread(simples);
15       thread.start();
16   }
17   }
18   }
```



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.ja
4   */
5   package com.ufg.labthreads;
6
7   /**
8    *
9    * @author aluno
10   */
11  public class ThreadSimples implements Runnable {
12      public void run() {
13          System.out.println("Olá de uma thread!");
14      }
15  }
```



```
com.ufg.labthreads.ThreadSimples >
Output - Run (ExecutaThread) x
--- maven-compiler-plugin:3.10.1:compile (default-compile) @ lab-threads ---
Changes detected - recompiling the module!
Compiling 3 source files to /home/aluno/Área de Trabalho/Aula 05/lab-threads/target/classes
--- exec-maven-plugin:3.0.0:exec (default-cli) @ lab-threads ---
Olá de uma thread!
-----
BUILD SUCCESS
-----
Total time: 1.521 s
Finished at: 2022-06-29T21:05:33-03:00
-----
```

Exercício 02:

```
LabThreadsApplication.java x ThreadSimples.java x ExecutaThread.java x
Source History
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-d
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
4   */
5   package com.ufg.labthreads;
6
7   /**
8    *
9    * @author aluno
10   */
11  public class ExecutaThread {
12      public static void main(String args[]) {
13          ThreadSimples simples = new ThreadSimples();
14          simples.start();
15      }
16  }
17
```

```
LabThreadsApplication.java x ThreadSimples.java x ExecutaThread.java x
Source History
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/lice
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class
4   */
5   package com.ufg.labthreads;
6
7   /**
8    *
9    * @author aluno
10   */
11  public class ThreadSimples extends Thread {
12      public void run() {
13          System.out.println("Hello from a thread!");
14      }
15  }
16
```

```
Output - Run (ExecutaThread) x
--- maven-compiler-plugin:3.10.1:compile (default-compile) @ lab-threads ---
Changes detected - recompiling the module!
Compiling 3 source files to /home/aluno/Área de Trabalho/Aula 05/lab-threads/target/classes
--- exec-maven-plugin:3.0.0:exec (default-cli) @ lab-threads ---
Hello from a thread!
-----
BUILD SUCCESS
-----
Total time: 1.570 s
Finished at: 2022-06-29T21:12:09-03:00
-----
```

Exercício 03:

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to cha
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this tem
4   */
5   package com.ufg.labthreads;
6
7   /**
8    *
9    * @author aluno
10   */
11  public class ThreadSimples {
12      public static void main(String args[]) throws InterruptedException {
13          String info[] = {
14              "Java",
15              "é uma boa linguagem.",
16              "Com threads",
17              "é melhor ainda."
18          };
19          for (int i = 0; i < info.length; i++) {
20              Thread.sleep(4000);
21              System.out.println(info[i]);
22              System.out.println("Nome da Thread Atual: " + Thread.currentThread().getName());
23          }
24      }
25  }
26
```

Exercício 04:

(i):

```
Output - Run (ThreadSimples) x
- Compiling 2 source files to /home/aluno/Área de Trabalho/Aula 05/lab-threads/target/classes
--- exec-maven-plugin:3.0.0:exec (default-cli) @ lab-threads ---
main Iniciando a thread Loop
main Esperando que a thread Loop termine
main Ainda esperando...
main Ainda esperando...
main Ainda esperando...
main Ainda esperando...
Thread-0 Java
main Ainda esperando...
main Ainda esperando...
main Ainda esperando...
main Ainda esperando...
Thread-0 é uma boa linguagem.
main Ainda esperando...
main Ainda esperando...
main Ainda esperando...
main Ainda esperando...
Thread-0 Com threads,
main Ainda esperando...
main Ainda esperando...
main Ainda esperando...
main Ainda esperando...
Thread-0 é melhor ainda.
main Finalmente!
-----
BUILD SUCCESS
-----
Total time: 17.486 s
Finished at: 2022-06-29T21:22:45-03:00
-----
|
```

(ii):

(iii):

A classe tem a função de executar uma Thread Loop enquanto a main espera ela finalizar, assim, a Thread Loop espera 4 segundos para cada uma de suas mensagens. Enquanto isso, a Main deve esperar e notificar que está esperando o término da Loop.

Caso a Loop seja interrompida durante o “sleep”, uma mensagem será passada: “Nada feito!”