



Compartilhe o seu link com luciorocha@professores.utfpr.edu.br

- 1) (Prática de programação) (Online) Acesse a atividade prática no link:
<https://codeboard.io/projects/380750>

```
1/**  
  
* TODO1: Crie 2 (duas) novas threads  
  
*  
  
* TODO2: Modifique o tempo de sleep das threads para ate 1 segundo.  
  
*  
  
* TODO3: AThread1: Crie no construtor da classe um identificador unico  
para cada thread.  
  
*           O construtor devera atribuir a variavel de instancia ID.  
  
*  
  
* TODO4: Crie 20 threads que entram em estado de sleep por ate 1  
segundo, mas apenas entrarao em  
  
*           sleep as que tiverem ID impar.  
  
*/  
  
//Threads implementam interface Runnable  
  
public class APrincipal1 {  
  
    public static void main(String[] args) {  
  
        Thread [] lista = new Thread [20];
```

```

        for(int i=0; i<20; i++)

            lista[i] = new Thread (new AThread1 ("t"+i, i));

        for(Thread t: lista)

            t.start();


        //new Thread(new AThread1("t1")).start(); //.start sai do estado 'Novo'
        para o estado 'Executar'

    }

}

```

```

import java.util.Random;


/**
 *
 * @author lucio
 */
public class AThread1 implements Runnable {

    private int tempoDormir;

    private String nomeTarefa;

    private Random r = new Random();

```

```
private int id;

public AThread1(String nomeTarefa, int id) {

    super();

    this.nomeTarefa = nomeTarefa;

    this.id = id;

    tempoDormir = r.nextInt(1000);

} //fim construtor

public void run() {

    try {

        System.out.println("NomeTarefa:" + nomeTarefa +

            " tempoDormir:" + "id " + this.id + tempoDormir);

        Thread.sleep(tempoDormir);

    } catch (Exception e) {

        System.out.println("Excecao na thread: "

            + e.getMessage());

    }

} //fim construtor

}
```

- 2) (Prática de programação) (Online) Acesse a atividade prática no link: <https://codeboard.io/projects/380752>

```
/**  
  
 * DONE: Principal: Crie 2 (duas) novas threads  
  
 *  
  
 * DONE: AThread2: Modifique a classe para que ela estenda da classe  
Thread.  
  
 *  
  
 * TODO3: AThread2: Crie no construtor da classe um identificador unico  
para cada thread.  
  
 *         O construtor devera atribuir a variavel de instancia ID.  
  
 *  
  
 * TODO4: Principal: Crie 20 threads que entram em estado de sleep por ate  
1 segundo, mas apenas entrarao em  
  
 *         sleep as que tiverem ID impar.  
  
 */  
  
//Threads estendem a classe Thread  
public class APrincipal2 {  
  
    public static void main(String[] args) {  
  
        Thread t1 = new Thread(new AThread2("t1")); //.start sai do estado  
'Novo' para o estado 'Executar'  
  
        //TODO1  
  
        Thread t2 = new Thread(new AThread2("t2"));
```

```
Thread t3 = new Thread(new AThread2("t3"));

t1.start();
t2.start();
t3.start();

System.out.println(t1.getId());
System.out.println(t2.getId());
System.out.println(t3.getId());

//TODO4

}
}
```

```
import java.util.Random;

/**
 *
 * @author lucio
 */
public class AThread2 extends Thread {
```

```
private int tempoDormir;

private String nomeTarefa;

private Random r = new Random();

public AThread2(String nomeTarefa) {

    super();

    this.nomeTarefa = nomeTarefa;

    tempoDormir = r.nextInt(5000);

    //TODO2: insira a chamada do metodo start() aqui

} //fim construtor

public void run() {

    //TODO3

    try {

        System.out.println("NomeTarefa:" + nomeTarefa +

            " tempoDormir:"+tempoDormir);

        Thread.sleep(tempoDormir);

    } catch (Exception e) {

        System.out.println("Excecao na thread: "

            + e.getMessage());

    }

} //fim construtor
```

```
}
```



36 | |

```
NomeTarefa:t1 tempoDormir:3724
NomeTarefa:t2 tempoDormir:1172
9
11
13
NomeTarefa:t3 tempoDormir:1391
```

Input to your program (press Enter to send)

User: #anonymous (sign in to save your changes) Role: Project user

- 3) (Prática de programação) (Online) Acesse a atividade prática no link: <https://codeboard.io/projects/380793>

//Soma de vetor nao sincronizado

/**

* DONE1: BPrincipal2: instancie 2 (duas) threads.

*

* DONE2: BPrincipal2: insira o trecho de codigo:

* 't1.thrd.join();'

* no bloco try-catch

*

* DONE3: BThread2: define que a classe implemente a interface Runnable.

```

*
* TODO4: BThread2: insira o trecho de código no construtor:
*     'thrd.start();'
*
*/

```

```

class BPrincipal2 {
    public static void main(String args[]) {

        int a[] = {1, 2, 3, 4, 5};

        BThread2 [] lista = new BThread2[200];
        for(int i=0; i<200; i++)
            lista[i] = new BThread2("Filho #" + i, a);

        BThread2 t1 = new BThread2("Filho #1", a);

        //TODO1
        BThread2 t2 = new BThread2("Filho #2", a);

        try {
            for(BThread2 i: lista){
                i.thrd.join();
            }

            //TODO2
            //t1.thrd.join();

```



```
//t2.thrd.join();
```

```
    } catch(InterruptedException exc) {
        System.out.println("Thread principal interrompida.");
    }
}
}
```

```
class BThread2 implements Runnable {
    Thread thrd;

    static BVetorSoma2 sa = new BVetorSoma2();

    int a[];

    int resposta;

    BThread2(String name, int nums[]) {
        thrd = new Thread(this, name); //Thread recebe parametro desta
        classe que implementa Runnable

        a = nums; //Thread inicia com acesso ao recurso compartilhado

        //TODO4

        thrd.start();
    }

    public void run() {
        int sum;
```

```

        System.out.println(thrd.getName() + " iniciou.");

        //Acesso ah regioao critica

        resposta = sa.somarVetor(a);

        System.out.println("Soma para " + thrd.getName() + " eh " + resposta);

        System.out.println(thrd.getName() + " terminando.");

    }

}

class BVetorSoma2 {

    private int sum;

    int somarVetor(int nums[]) {

        sum = 0;

        for(int i=0; i<nums.length; i++) {

            sum += nums[i];

            System.out.println("Total          da          "          +
Thread.currentThread().getName() + " eh " + sum);

            /*try {

                Thread.sleep(10);

            }

            catch(InterruptedException exc) {

                System.out.println("Thread interrompida.");

            }*/

        }

        return sum;

    }
}

```

```

Filho #149 iniciou.
Total da Filho #149 eh 1
Total da Filho #149 eh 3
Total da Filho #149 eh 6
Total da Filho #149 eh 10
Total da Filho #149 eh 15
Soma para Filho #149 eh 15
Filho #149 terminando.
Filho #148 iniciou.
Total da Filho #148 eh 1
Total da Filho #148 eh 3

```

Your program output has more than 32000 characters. That's quite a lot.
For this reason, Codeboard has terminated your program.

- 4) (Prática de programação) (Online) Acesse a atividade prática no link:
<https://codeboard.io/projects/380811>

```
/**
```

```
* DONE1: CPrincipal3.java: instancie 2 (duas) threads.
```

```
*
```

```
* DONE2: CPrincipal3.java: Adicione o seguinte trecho de código:
```

```
*     't1.thrd.join();'
```

```
*
```

```
* TODO3: CThread3.java: adicione a sincronizacao da regiao critica.
```

```
*     'synchronized(sa) //Acesso a regiao critica'
```

```
*
```

```
*/
```

```
//Soma de vetor sincronizado
```

```
class CPrincipal3 {
```

```
    public static void main(String args[]) {
```

```

int a[] = {1, 2, 3, 4, 5};

int TAM = 50;

CThread3 [] lista = new CThread3[TAM];

for(int i=0; i < TAM ; i++)

    lista[i] = new CThread3 ("Filho #" + i, a);


//CThread3 t1 = new CThread3("Filho #1", a);
//CThread3 t2 = new CThread3("Filho #2", a);

try {

    //TODO2

    //    t1.thrd.join();
    //    t2.thrd.join();

    for (CThread3 i : lista)

        i.thrd.join();

} catch (InterruptedException exc) {

    System.out.println("Thread principal interrompida.");

}

}

}

```

```

class CThread3 implements Runnable {

    Thread thrd;

    static CVetorSoma3 sa = new CVetorSoma3();

    int a[];

    int resposta;

    CThread3(String name, int nums[]) {

        thrd = new Thread(this, name);

        a = nums; //Thread inicia com acesso ao recurso compartilhado

        thrd.start();

    }

    public void run() {

        int sum;

        System.out.println(thrd.getName() + " iniciou.");

        //TODO2

        synchronized(sa){

            resposta = sa.somarVetor(a);

        }

        System.out.println("Soma para " + thrd.getName() + " eh " + resposta);

        System.out.println(thrd.getName() + " terminando.");

    }

}

class CVetorSoma3 {

```

```
private int sum;

int somarVetor(int nums[]) {
    sum = 0;

    for(int i=0; i<nums.length; i++) {
        sum += nums[i];

        System.out.println("Total da " + Thread.currentThread().getName() + "
eh " + sum);

        try {
            Thread.sleep(10);
        }
        catch(InterruptedException exc) {
            System.out.println("Thread interrompida.");
        }
    }
    return sum;
}
```

```

Filho #1 terminando.
Total da Filho #2 eh 1
Total da Filho #2 eh 3
Total da Filho #2 eh 6
Total da Filho #2 eh 10
Total da Filho #2 eh 15
Soma para Filho #2 eh 15
Filho #2 terminando.

```

```

Input to your program (press Enter to send)

```

- 5) (Prática de programação) (Online) Acesse a atividade prática no link: <https://codeboard.io/projects/397356>

```

/**
 * TODO 1: Crie 20 novas instancias de threads.
 * TODO 2: Execute as instancias no pool de threads.
 *
 */

import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class Principal {

    public class AThread implements Runnable {

        public void run(){
            System.out.println("Ola Mundo");
        }
    }

    public Principal(){
        ExecutorService pool = Executors.newCachedThreadPool();

        // AThread t1 = new AThread();
        // Thread t = new Thread(t1);
        // TAM = 50
        Thread [] lista = new Thread[50];
    }
}

```

```
for(int i = 0; i < 50; i++)  
    lista[i] = new Thread(new AThread());  
  
for (Thread t : lista)  
    pool.execute(t);  
  
pool.shutdown();  
}  
  
public static void main(String[] args) {  
    new Principal();  
}  
}
```

```
Ola Mundo  
Ola Mundo  
Ola Mundo  
Ola Mundo  
Ola Mundo  
Ola Mundo  
Ola Mundo  
Ola Mundo
```

```
Input to your program (press Enter to send)
```