

# C206 - POO

Arthur Openheimer

# Informações Gerais

- Atendimento → Terça-feira 17:30-19:10, prédio 1, sala 19
- Email → arthur.openheimer@ges.inatel.br
- Github → <https://github.com/ArthurOpenheimer/C206-Monitoria>

# Threads

Thread é uma forma como uma tarefa de um programa é dividido em duas ou mais tarefas que podem ser executadas simultaneamente

Uma thread permite que duas ou mais partes do programa sejam executadas ao mesmo tempo. Em equipamentos que possuem apenas uma CPU, cada thread é processada de forma aparentemente simultânea, pois a mudança entre uma thread e outra é feita de forma tão rápida que para o utilizador, processando de forma concorrente. Em equipamentos com múltiplos CPUs, multi-cores, as threads são realizadas realmente de forma simultânea.

# Criando uma Thread

Implementando uma Thread herdando a classe Thread:

```
public class ThreadExemplo extends Thread{  
  
    @Override  
    public void run(){  
  
    }  
  
}
```

```
public class Main {  
    Run | Debug  
    public static void main(String[] args) {  
        ThreadExemplo t1 = new ThreadExemplo();  
        t1.start();  
    }  
}
```

# Criando uma Thread

Implementando uma Thread implementando a interface Runnable:

```
public class ThreadExemplo implements Runnable{  
  
    @Override  
    public void run() {  
  
    }  
}
```

```
public class Main {  
    Run | Debug  
    public static void main(String[] args) {  
        ThreadExemplo t1 = new ThreadExemplo();  
        new Thread(t1).start();  
    }  
}
```

# Criando uma Thread

Exemplo prático com uma thread que printa um número de 0 a 4 a cada 3 segundos e uma thread que printa a string “Tocando música” a cada 5 segundos

# Criando uma Thread

```
import java.util.concurrent.TimeUnit;
public class ThreadMusica extends Thread{

    @Override
    public void run() {
        for (int i = 0; i < 3; i++) {
            System.out.println("Tocando música ");

            try {
                TimeUnit.SECONDS.sleep(5);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
import java.util.concurrent.TimeUnit;
public class ThreadTempo extends Thread{

    @Override
    public void run() {
        for (int i = 0; i < 5; i++) {
            System.out.println(i);

            try {
                TimeUnit.SECONDS.sleep(3);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

# Exercício

Crie 3 Threads e as execute na Main:

1. Uma thread deve printar um número de 1 a 10 a cada 2 segundos;
2. Uma thread deve printar um número de 2 a 6 a cada 4 segundos;
3. Uma thread deve calcular e printar o quadrado dos números de 1 a 5 num intervalo de 3 segundos.