



POO

Asserções

Prof. Alcides Calsavara

PUCPR

Assertão

- ↻ Indica uma condição que deve ser satisfeita para que a execução prossiga normalmente.
- ↻ Se a condição não for satisfeita, uma exceção é gerada.

↻ Exemplo em Java:

```
assert( x > 0 && x <= 10 );
```

A expressão lógica
x > 0 && x <= 10
deve ser verdadeira para que
a execução prossiga
normalmente.

Exemplo:

```
public class LeituraInteiro {  
    public static void main(String[] args) {  
  
        int k = 10;  
  
        System.out.println("Antes de assert ...");  
  
        assert(k != 10);    Causa uma exceção  
  
        System.out.println("Depois de assert!");  
    }  
}
```

Nunca deveria ser executado!

No entanto ...

```
C:\Users\AiA-Dev\.jdk\openjdk-21.0.2\bin\java.exe "-javaagent:
```

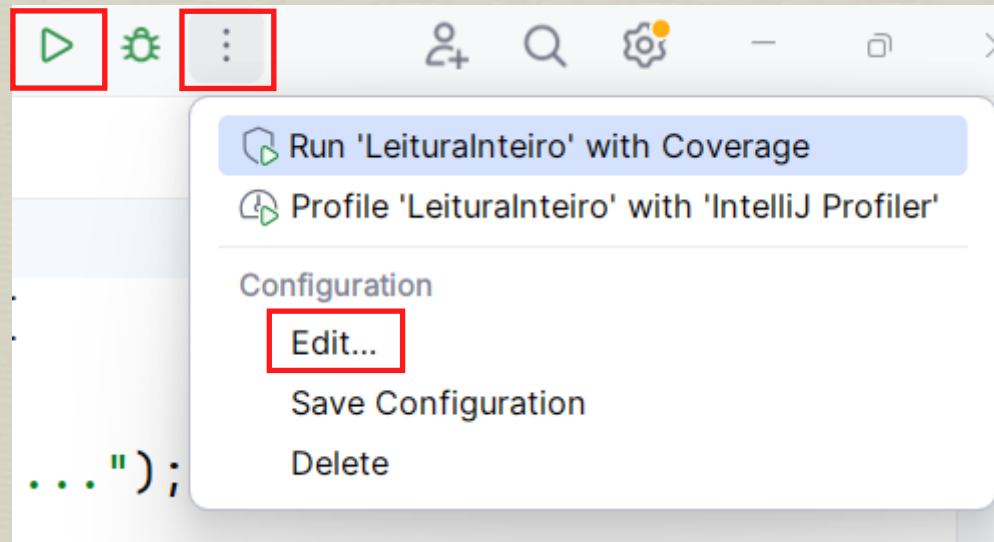
```
Antes de assert ...
```

```
Depois de assert!
```

```
Process finished with exit code 0
```

Motivo: falta habilitar o tratamento de asserções na
Máquina Virtual de Java

Habilitação do tratamento de asserções no IntelliJ





Run/Debug Configurations



Application

LeituraInteiro

Name: LeituraInteiro

☐ Store as project file

Run on: Local machine

[Manage targets...](#)

Run configurations may be executed locally or on a target: for example in a Docker Container or on a remote host using SSH.

Build and run

[Modify options](#) Alt+M

java 21 SDK of 'Assert' LeituraInteiro

Program arguments

Press Alt for field hints

Add Run Options

Operating System

- Allow multiple instances Alt+U
- ✓ Environment variables Alt+E
- Redirect input

Java

- Do not build before run
- Use classpath of module Alt+O
- Modify classpath
- Add dependencies with "provided" scope to classpath
- Unnamed class run configuration
- Shorten command line
- Add VM options Alt+V

Logs

- Specify logs to be shown in console
- Save console output to file

Run/Debug Configurations

Application

LeituraInteiro

Name:

LeituraInteiro

☐ Store as project file

Run on:

Local machine

Manage targets...

Run configurations may be executed locally or on a target: for example in a Docker Container or on a remote host using SSH.

Build and run

Modify options

java 21 SDK of 'Assertion' module

-ea

LeituraInteiro

Program arguments

Executando novamente depois da habilitação ...

```
C:\Users\AiA-Dev\.jdk\openjdk-21.0.2\bin\java.exe -ea "-javaagent:
```

```
Antes de assert ...
```

```
Exception in thread "main" java.lang.AssertionError Create breakpoint  
    at LeituraInteiro.main(LeituraInteiro.java:5)
```

```
Process finished with exit code 1
```

Lê um valor
inteiro diferente
de 10.

Trata exceção
referente a
asserção

```
import java.util.Scanner;

public class LeituraInteiro {
    public static void main(String[] args) {
        try {
            System.out.print("Digite um valor inteiro: ");
            Scanner teclado = new Scanner(System.in);
            int k = teclado.nextInt();

            System.out.println("Antes de assert ...");
            assert(k != 10);
            System.out.println("Depois de assert!");
        }
        catch (AssertionError e) {
            System.out.println("Valor inválido");
        }
        catch (Exception e) {
            System.out.println(e);
        }
        System.out.println("Fim do programa!");
    }
}
```

Lê um valor inteiro maior que 4 e menor que 9.

```
import java.util.InputMismatchException;
import java.util.Scanner;
public class LeituraInteiroIntervalo {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int k = 0;
        boolean sucesso = false;
        while (!sucesso)
            try {
                System.out.print("Digite um valor inteiro: ");
                k = teclado.nextInt();
                assert(k > 4 && k < 9);
                sucesso = true;
            }
            catch (AssertionError e) {
                System.out.println("Valor inválido");
            }
            catch (InputMismatchException e) {
                System.out.println("Entrada em formato inválido");
                teclado.nextLine();
            }
        System.out.println("Fim do programa! k = " + k);
    }
}
```