DIGITAL INNOVATION ONE

Objetivos

Copiar caminho

Copiar o caminho dos itens selecionados para a Área de Transferência.

1. Entender o que são operadores relacionais

2. Saber como utilizá-los

DIGITAL INNOVATION ONE

Conceituação

"São símbolos especiais quais são capazes de realizar comparações entre determinados operandos e, em seguida, retornar um resultado"

Tipos:

- Similaridade: igual, diferente
- Tamanho: maior, maior igual, menor, menor igual

DIGITAL INNOVATION ONE

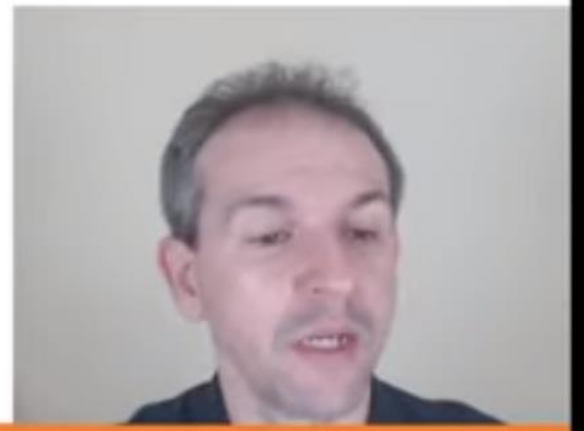
Utilização

Similaridade

- Igualdade: determina se um operando é igual ao outro
- Diferença: determina se um operando não é igual ao outro

Símbologia

- Igualdade: ==
- Diferença: !=



DIGITAL INNOVATION ONE

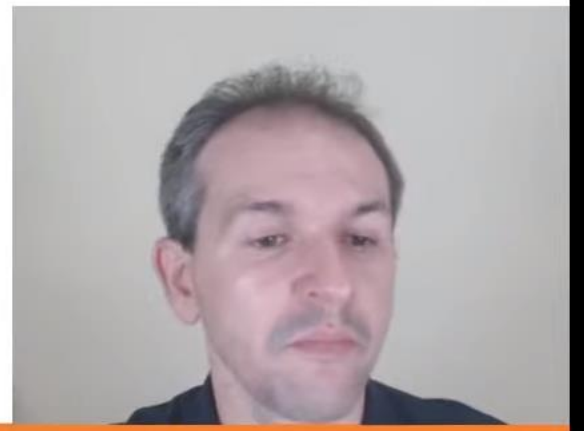
Utilização

Tamanho

- Maior: determina se um operando é maior do que outro
- Maior Igual: determina se um operando é maior ou igual a outro

Símbologia

- Maior: >
- Maior Igual: >=



DIGITAL INNOVATION ONE

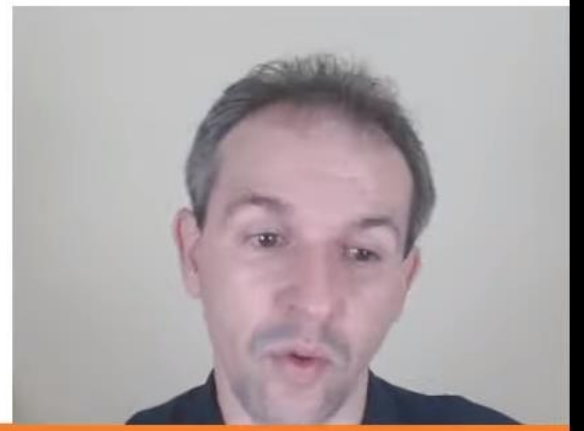
Utilização

Tamanho

- Menor: determina se um operando é menor do que outro
- Menor igual: determina se um operando é menor ou igual a outro

Símbologia

- Menor: <
- Menor Igual: <=



DIGITAL INNOVATION ONE

Exemplos

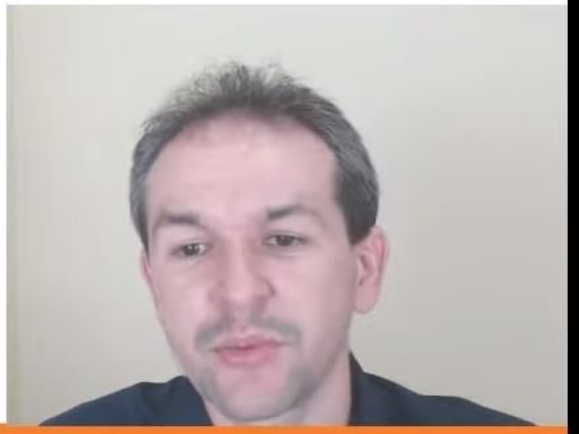
```
int i1 = 10; int i2 = 20; float f1 = 4.5f; float f2 = 3.5f;
char c1 = 'x'; char c2 = 'y'; String s1 = "Fulano"; String s2 =
"Fulano"; boolean b1 = true; boolean b2 = false;

i1 == i2, i1 != i2, i1 > i2, i1 <= i2
f1 == f2, f1 != f2, f1 >= f2, f1 < f2
c1 == c2, c1 != c2, c1 > c2, c1 <= c2
s1 == s2, s1 != s2, s1 >= s2, s1 < s2
b1 == b2, b1 != b2, b1 > b2, b1 <= i1
I2 > f1
```

DIGITAL INNOVATION ONE

Exercitando

Criar um simples projeto no IntelliJ para realizar as comparações dos slide anterior, além de utilizar os demais tipos de dados não apresentados.



DIGITAL INNOVATION ONE

Objetivos

1. Entender o que são operadores lógicos

2. Saber como utilizá-los

DIGITAL INNOVATION ONE

Conceituação

"São símbolos especiais quais são capazes de realizar comparações lógicas entre operandos lógicos ou expressões e, em seguida, retornar um resultado"

Tipos:

- Conjunção
- Disjunção
- Disjunção exclusiva
- Negação

DIGITAL INNOVATION ONE

Conceituação

Tipos:

- Conjunção: operação lógica que só é verdadeira quando ambos os operandos ou expressões envolvidas são verdade

Simbologia:

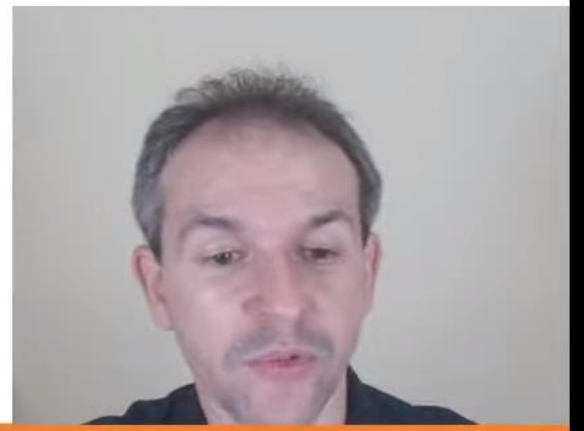
- &&

Terminologia:

- and(e)

O-E	O-E	R
V	V	V
V	F	F
F	V	F
F	F	F

O – Operando E - Expressão R - Resultado



DIGITAL INNOVATION ONE

Conceituação

Tipos:

- Disjunção: operação que só é falsa quando ambos os operandos ou expressões envolvidas são falsos

Simbologia:

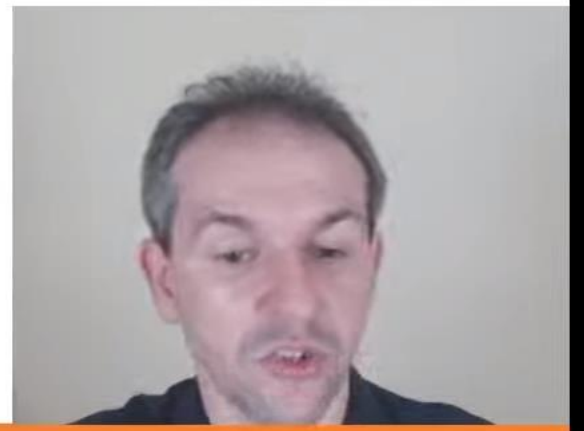
- ||

Terminologia:

- or(ou)

O-E	O-E	R
V	V	V
V	F	V
F	V	V
F	F	F

O – Operando E - Expressão R - Resultado



DIGITAL INNOVATION ONE

Conceituação

Tipos:

- Disjunção exclusiva: operação que só é verdade quando ambos os operandos ou expressões são opostos

Simbologia:

- ^

Terminologia:

- xor

O-E	O-E	R
V	V	F
V	F	V
F	V	V
F	F	F

O – Operando E - Expressão R - Resultado

DIGITAL INNOVATION ONE

Conceituação

Tipos:

- Negação: operação que inverte o valor lógico de um operando ou expressão

Simbologia:

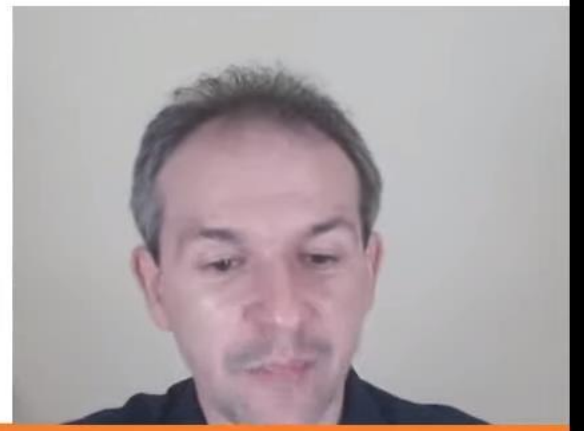
- !

Terminologia:

- inverção

O-E	R
V	F
F	V

O – Operando E - Expressão R - Resultado



DIGITAL INNOVATION ONE

Curiosidades

- Operadores bitwise: & e |
- Operadores shift: ~, >>, >>>, <<

DIGITAL INNOVATION ONE

Exemplos

```
boolean b1 = true; boolean b2 = false;
boolean b3 = true; boolean b4 = false;

b1 && b2, b1 && b3
b2 || b3, b2 || b4
b1 ^ b3, b4 ^ b1
!b1, !b2
(i1 > i2) || (f2 < f1)
((i1 + i2) < (f2 - f1)) && true
```

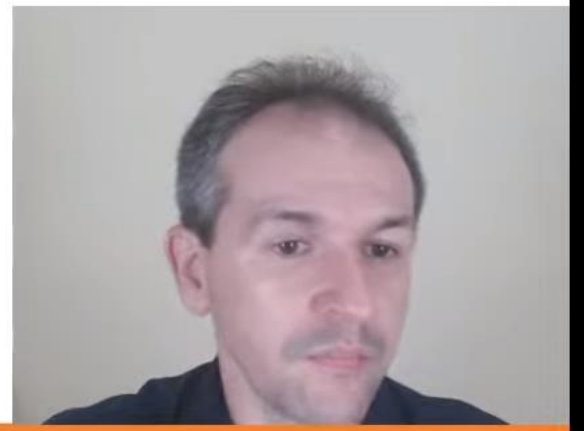
DIGITAL INNOVATION ONE

Boas práticas

- Crie variáveis auxiliares para guardar resultados intermediários

```
(salarioMensal < mediaSalario) && (quantidadeDependentes >=
mediaDependentes)
    pode ser
(salarioBaixo) && (muitosDependentes)

boolean recebeAuxilio = (salarioBaixo)
            && (muitosDependentes);
```



DIGITAL INNOVATION ONE

Exercitando

Criar um simples projeto no IntelliJ e codificar os exemplos dos slides anteriores para compreender as operações lógicas. Utilize operandos e expressões.

DIGITAL INNOVATION ONE

Objetivos

1. Entender o que são estruturas de controle de fluxo

2. Saber como usar cada uma

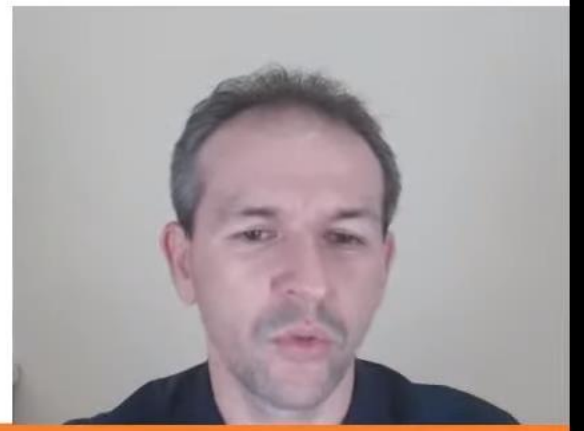
DIGITAL INNOVATION ONE

Conceituação

"São estruturas que tem a capacidade de direcionar o fluxo de execução do código"

Tipos:

- Decisão: if, if-else, if-else-if, switch e operador ternário
- Repetição: for, while, do while
- Interrupção: break, continue e return



DIGITAL INNOVATION ONE

Criação

Tipos:

- Decisão: estrutura que avalia uma condição booleana ou variável para direcionar o fluxo de execução

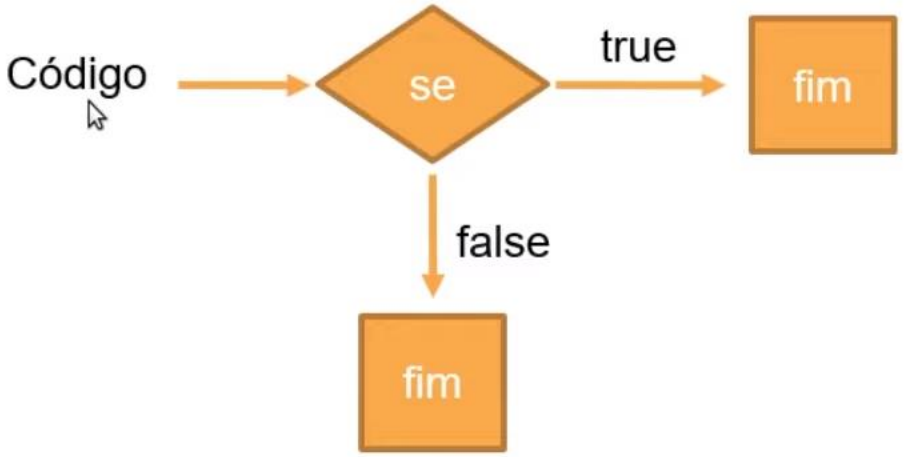
- Opções: if(se), switch(escolha) e operador ternário

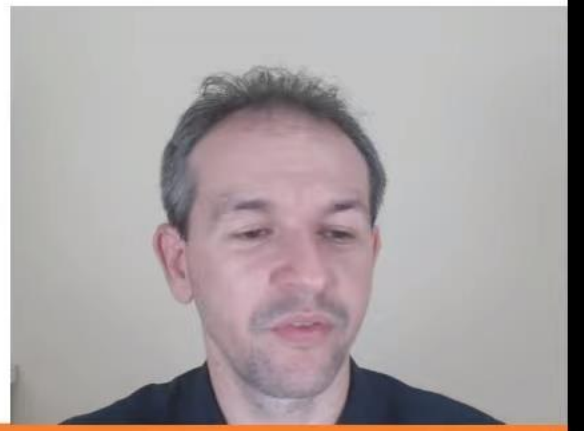
DIGITAL INNOVATION ONE

Criação

Tipos:

- Decisão: if, if-else, if-else-if, if-else-if-else





DIGITAL INNOVATION ONE

Criação

Tipos:

- Decisão: if, if-else, if-else-if

```
if (condição) {
}
```

```
if (condição) {
} else {
}
```

```
if (condição) {
} else if (condição) {
} else {
}
```

DIGITAL INNOVATION ONE

Criação

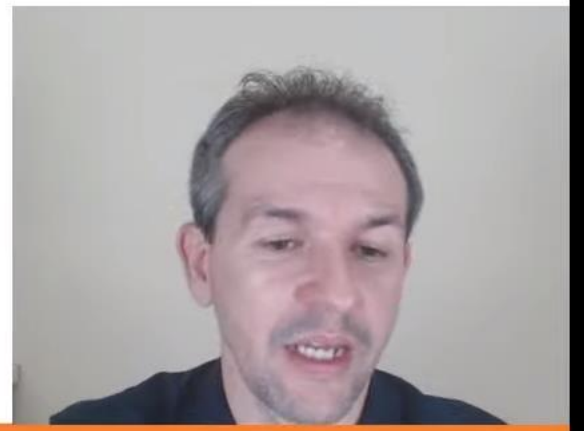
Tipos:

- Decisão

```
if (idade > 18) {
}
```

```
if (aprovado) {
} else {
}
```

```
if (casado && temFilhos) {
} else if (casado && semFilhos) {
} else {
}
```

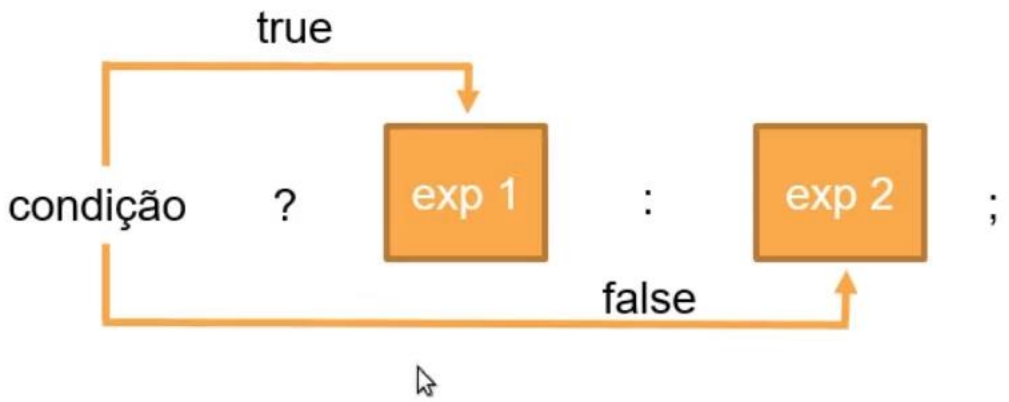


DIGITAL INNOVATION ONE

Criação

Tipos:

- Decisão: operador ternário



DIGITAL INNOVATION ONE

Criação

Tipos:

- Decisão: operador ternário

condição ? true : false; condição ? true : null;

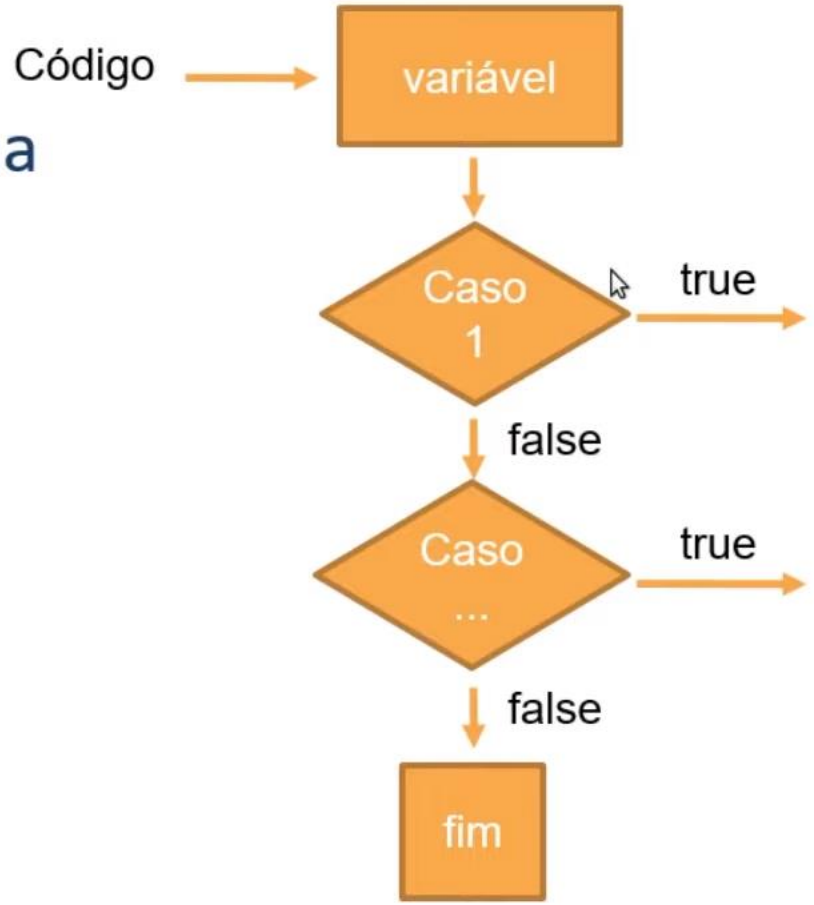
ligado ? desligar : ligar; emMovimento ? freia : null;

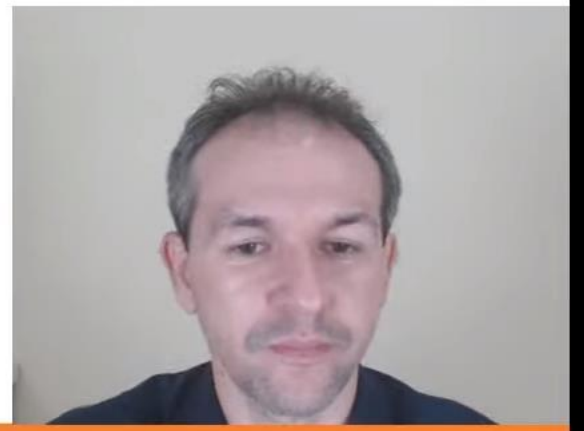
DIGITAL INNOVATION ONE

Criação

Tipos:

- Decisão: escolha





DIGITAL INNOVATION ONE

Criação

Tipos:

- Decisão: switch

Variável:

byte
short
char
int
Enum
String

```
switch (variável) {
  case 1 :
   break;
  case … :
   break;
  default:
   break;
}
```

DIGITAL INNOVATION ONE

Criação

Tipos:

- Decisão

```
switch (olhos) {
  case "AZUIS" :
   break;
  case "VERDES" :
   break;
  case "CASTANHOS" :
   break;
  default:
   break;
}
```

DIGITAL INNOVATION ONE

Boas práticas

- Switch é para valores exatos e if para expressões booleanas
- Evitar usar o default do switch para "cases genéricos"
- Evitar o efeito "flecha" dos if's
- Evitar muitos if's aninhados
- Usar a boa prática da aula 2 para diminir o tamanho if

DIGITAL INNOVATION ONE

Exercitando

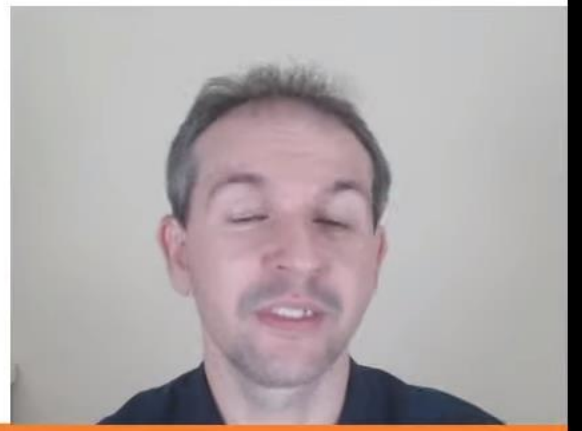
Criar um simples projeto no IntelliJ e criar variáveis e expressões para usar nas estruturas **if** e **switch**.

Com if, exiba o nome do mês do ano de acordo com seu numero. Evite o efeito "flecha". Faça também outro if que verifique se o mês é julho, dezembro ou janeiro, para assim exibir o texto "Férias".

DIGITAL INNOVATION ONE

Exercitando

Com switch use String para a partir do dia da semana, exibir seu número. Ainda no switch, faça outro exemplo onde, se uma variável inteira for entre 1 e 3 exibir o texto "Certo". Se for 4 exibir "Errado", se for 5 "Talvez". Pra demais valores exibir "Valor indefinido".



DIGITAL INNOVATION ONE

Objetivos

1. Entender o que são blocos

2. Tipos de blocos

3. Saber como usar blocos

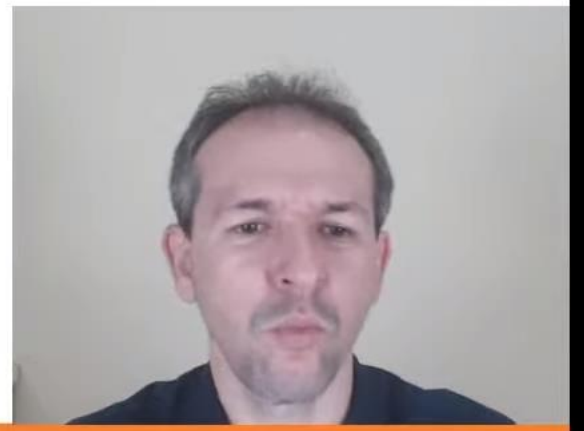
DIGITAL INNOVATION ONE

Conceituação

"É um grupo de 0 ou mais códigos quais trabalham em conjunto para executar uma operação"

Tipos:

- Locais: dentro de métodos
- Estáticos: dentro de classes
- Instância: dentro de classes



DIGITAL INNOVATION ONE

Criação

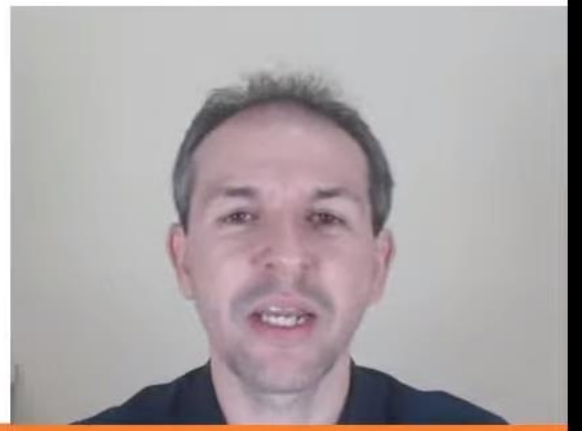
Tipos:

- Locais

{

...

}



DIGITAL INNOVATION ONE

Exemplos

```
if (autorizado) {
    CarregarPerfil
    DirecionarPáginaPrincipal
}

if (menorIdade)
    DirecionarPáginaProibido
```

```
if (autorizado) {
    ...
} else {
    ...
}
```