>>> Programação Orientada a Objetos (POO)

. . . Estruturas de controle: Condicionais

Prof: André de Freitas Smaira

[~]$ _

```
 .-------------------------------------------.
|| Closure:                                  ||
||    (A + B) is in K whenever A,B is in K   ||
||    (A * B) is in K whenever A,B is in K   ||
|| Neutral elements:                         ||
||    There is \/, such that A + \/ = A to ev. A in k ||
||    There is /\, such that A + /\ = A to ev. A in k ||
 "==========================================="
```

kOs

Álgebra de Boole e Lógica

>>> Lógica Proposicional - Conectivos Lógicos

São operações comuns em lógica:

- AND (E lógico)
  - Verdadeiro apenas se as duas entradas forem verdadeiras

>>> Lógica Proposicional - Conectivos Lógicos

São operações comuns em lógica:

* AND (E lógico)
    * Verdadeiro apenas se as duas entradas forem verdadeiras
    * Ex: Meu nome é André E tenho 34 anos. (Verdade!)

>>> Lógica Proposicional - Conectivos Lógicos

São operações comuns em lógica:

- AND (E lógico)
  - Verdadeiro apenas se as duas entradas forem verdadeiras
  - Ex: Meu nome é André E tenho 34 anos. (Verdade!)
- OR (OU lógico)
  - Verdadeiro se pelo menos uma das entradas forem verdadeiras

>>> Lógica Proposicional - Conectivos Lógicos

São operações comuns em lógica:

* AND (E lógico)
  * Verdadeiro apenas se as duas entradas forem verdadeiras
  * Ex: Meu nome é André E tenho 34 anos. (Verdade!)
* OR (OU lógico)
  * Verdadeiro se pelo menos uma das entradas forem verdadeiras
  * Ex:
    $1 \neq 1$ OU π é igual 3.1415... (Verdade!)
    $1 = 1$ OU π é igual 3.1415... (Verdade!)
    $1 \neq 1$ OU $\pi^2 = g$ (Falso!)

>>> Lógica Proposicional - Conectivos Lógicos

São operações comuns em lógica:

* AND (E lógico)
  * Verdadeiro apenas se as duas entradas forem verdadeiras
  * Ex: Meu nome é André E tenho 34 anos. (Verdade!)
* OR (OU lógico)
  * Verdadeiro se pelo menos uma das entradas forem verdadeiras
  * Ex:
    $1 \neq 1$ OU π é igual 3.1415... (Verdade!)
    $1 = 1$ OU π é igual 3.1415... (Verdade!)
    $1 \neq 1$ OU $\pi^2 = g$ (Falso!)
* XOR (OU ... OU lógico)
  * Verdadeiro se apenas uma das entradas for verdadeira.

>>> Lógica Proposicional - Conectivos Lógicos

São operações comuns em lógica:

* AND (E lógico)
  * Verdadeiro apenas se as duas entradas forem verdadeiras
  * Ex: Meu nome é André E tenho 34 anos. (Verdade!)
* OR (OU lógico)
  * Verdadeiro se pelo menos uma das entradas forem verdadeiras
  * Ex:
    $1 \neq 1$ OU π é igual 3.1415... (Verdade!)
    $1 = 1$ OU π é igual 3.1415... (Verdade!)
    $1 \neq 1$ OU $\pi^2 = g$ (Falso!)
* XOR (OU ... OU lógico)
  * Verdadeiro se apenas uma das entradas for verdadeira.
  * Ex: OU vou a praia OU vou estudar

>>> Lógica Proposicional - Conectivos Lógicos

São operações comuns em lógica:

* AND (E lógico)
  * Verdadeiro apenas se as duas entradas forem verdadeiras
  * Ex: Meu nome é André E tenho 34 anos. (Verdade!)
* OR (OU lógico)
  * Verdadeiro se pelo menos uma das entradas forem verdadeiras
  * Ex:
    $1 \neq 1$ OU π é igual 3.1415... (Verdade!)
    $1 = 1$ OU π é igual 3.1415... (Verdade!)
    $1 \neq 1$ OU $\pi^2 = g$ (Falso!)
* XOR (OU ... OU lógico)
  * Verdadeiro se apenas uma das entradas for verdadeira.
  * Ex: OU vou a praia OU vou estudar
* NOT (NÃO lógico)
  * Verdadeiro se a entrada for falsa.

>>> Lógica Proposicional - Conectivos Lógicos

São operações comuns em lógica:

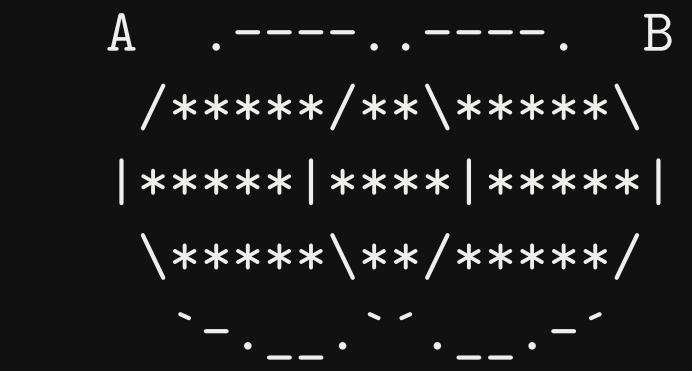
* AND (E lógico)
  * Verdadeiro apenas se as duas entradas forem verdadeiras
  * Ex: Meu nome é André E tenho 34 anos. (Verdade!)
* OR (OU lógico)
  * Verdadeiro se pelo menos uma das entradas forem verdadeiras
  * Ex:
    $1 \neq 1$ OU π é igual 3.1415... (Verdade!)
    $1 = 1$ OU π é igual 3.1415... (Verdade!)
    $1 \neq 1$ OU $\pi^2 = g$ (Falso!)
* XOR (OU ... OU lógico)
  * Verdadeiro se apenas uma das entradas for verdadeira.
  * Ex: OU vou a praia OU vou estudar
* NOT (NÃO lógico)
  * Verdadeiro se a entrada for falsa.
  * Ex: Vou estudar -> NÃO vou estudar

>>> Visualizando com conjuntos

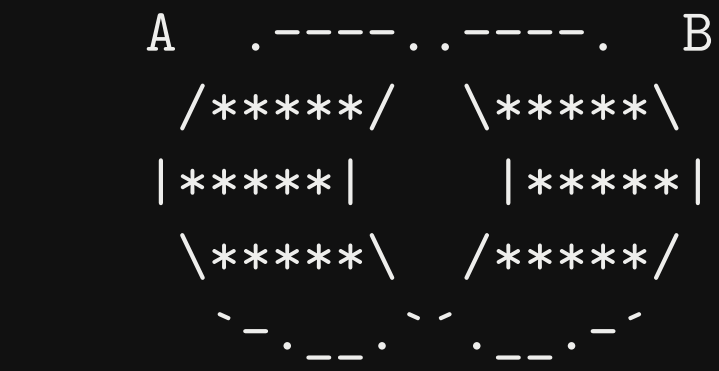
AND

NOT A

OR



XOR



Fazendo computadores tomarem decisões!

>>> Condicionais em C

```c
if (condicao1) {
    // Caso a condicao1 seja verdadeira
    // Executa o código que estiver aqui dentro
}
else if (condicao2) {
    // Executa o código que estiver aqui dentro
}
...
else if (condicaoN) {
    // Executa o código que estiver aqui dentro
}
else {
    // Se todas as condições falharem...
    // faça o que está aqui dentro
}
```

>>> Condicionais em C: Sintaxe

```
#include <iostream>
int main()
{
    char classe;
    std::cout << "Você é um robô? [s - sim, n - não]\n";
    std::cin >> classe;
    // Note que para apenas um comando dentro do if
    // não é necessário usar chaves para delimitar o bloco
    if (classe == 's') // Este "==" significa equivalência
        std::cout << "Olá, amigo!\n";
    else if (classe == 'n')
        std::cout << "Perigo!!! Humano detectado!!!\n";
    else
        std::cout << "Erro Fatal, entrada não esperada!\n";
    return 0;
}
```

>>> Condicionais em C: Cuidado!

Esse código imprime o que?

```
#include <iostream>
int main() {
    int n=1, m=10;
    if(n==1)
        if(m==1)
            std::cout << "m vale 1\n";
    else
        std::cout << "n nao vale 1\n";
    return 0;
}
```

>>> Condicionais em C: Cuidado!

Esse código imprime o que?

```
#include <iostream>
int main() {
    int n=1, m=10;
    if(n==1)
        if(m==1)
            std::cout << "m vale 1\n";
    else
        std::cout << "n nao vale 1\n";
    return 0;
}
```

"n nao vale 1"

>>> Comparações lógicas em C

```c
if (ca == fe) {
    //Bloco de comandos
}
if (ca != fe) {
    //Bloco de comandos
}
if (ca > fe) {
    //Bloco de comandos
}
if (ca >= fe) {
    //Bloco de comandos
}
if (ca < fe) {
    //Bloco de comandos
}
if (ca <= fe) {
    //Bloco de comandos
}
```

>>> Operações lógicas em C: Sintaxe

```c
if (p && q) {  // python: if p and q:
    //Bloco de comandos
}
if (p || q) { // python: if p or q:
    //Bloco de comandos
}
if (!p || q) { // python: if not p and q:
    //Bloco de comandos
}
if ( !(p || q) ) { // python: if not (p or q):
    //Bloco de comandos
}
```

```c
/* Exemplo usando conectivos */
#include <stdio.h>
int main()
{
    int ca, fe;
    scanf("%d %d", &ca, &fe);
    if (ca || fe)
        printf("Ou ca ou fe ou ambos são verdadeiros\n");
    else
        printf("Ambos são falsos\n");
    if (ca && fe)
        printf("Ambos são verdadeiros\n");
    else
        printf("Ou ca ou fe ou ambos são falsos\n");
    if ((ca || fe) && !(ca && fe)) // XOR xD
        printf("Ou ca ou fe é verdadeiro\n");
    else
        printf("Ambos são iguais\n");
}
```

>>> Referências e leitura recomendada

* Aulas do Grupo Maratona IFSC (Ian Giestas Pauli e eu)

---

[1] Capítulo 2 - Algebra de Boole

>>> Referências e leitura recomendada

* Aulas do Grupo Maratona IFSC (Ian Giestas Pauli e eu)
* Álgebra Booleana e aplicações [1]
  http://www.vision.ime.usp.br/~jb/boolean%20algebra/aulas_mac0329.pdf
* Edward V. Huntington, Sets of Independent Postulates for the Algebra of Logic
  https://www.jstor.org/stable/1986459

[1] Capítulo 2 - Algebra de Boole