



PARADIGMAS E LINGUAGENS DE PROGRAMAÇÃO

ENGENHARIA DA COMPUTAÇÃO – UFC/SOBRAL

Prof. Danilo Alves

`danilo.alves@alu.ufc.br`

ESTRUTURAS DE CONTROLE NO NÍVEL DE SENTENÇA



UNIVERSIDADE
FEDERAL DO CEARÁ

- Introdução
- Sentenças de seleção
- Sentenças de iteração

ESTRUTURA DE CONTROLE



UNIVERSIDADE
FEDERAL DO CEARÁ

- Uma estrutura de controle é uma (sentença de controle) e a {coleção de sentenças} cuja execução ela controla
- Em uma linguagem há basicamente duas estruturas de controle:
 - de seleção
 - de repetição

SENTENÇA DE SELEÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Uma sentença de seleção fornece os meios para escolher entre dois ou mais caminhos de execução em um programa
- Duas categorias gerais:
 - Dois caminhos
 - Seleção múltipla, ou n caminhos

DOIS CAMINHOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- Forma geral:

if expressão_de_controle

cláusula então

cláusula senão

- É possível considerar uma estrutura de seleção simples

- if expressão_de_controle

- cláusula então

- Questões de projeto:

- Qual é a forma e o tipo da expressão que controla a seleção?

- Como são especificadas as cláusulas então e senão?

A EXPRESSÃO DE CONTROLE



UNIVERSIDADE
FEDERAL DO CEARÁ

- Expressões de controle são especificadas entre parênteses se a palavra reservada `then` (ou algum outro marcador sintático) não for usada para introduzir a cláusula então
 - Em Python, utiliza-se o marcador `:` dois-pontos para indicar a cláusula então
 - Nessa linguagem, o sinal de dois-pontos faz a função da palavra `then`
- A expressão de controle **pode ser aritmética**
 - **As expressões válidas possuem valores lógicos equivalentes, quando necessário**

FORMA DE CLÁUSULA



UNIVERSIDADE
FEDERAL DO CEARÁ

- Em muitas linguagens contemporâneas, as cláusulas **então** e **senão** aparecem ou como sentenças simples ou como sentenças compostas
- Variação:
 - Em Perl, todas as cláusulas **então** e **senão** devem ser sentenças compostas
 - Sentenças devem ser envolvidas por chaves SEMPRE
 - Em Fortran 95, Ada e Ruby, as cláusulas **então** e **senão** são sequências de sentenças
 - A construção de seleção completa é terminada com uma palavra reservada. Ex.: **end**
- Python usa indentação para especificar sentenças compostas

```
if x > y :  
    x = y  
    print "case 1"
```

ANINHANDO SELETORES



UNIVERSIDADE
FEDERAL DO CEARÁ

- Exemplo em Java

```
if (soma == 0)
```

```
    if (cont == 0)
```

```
        resultado = 0;
```

```
else resultado = 1;
```

- Para qual if a cláusula else está associada?
- Regra de semântica estática de C/C++ e Java: **else** sempre casa com a **if** mais próxima

ANINHANDO SELETORES



UNIVERSIDADE
FEDERAL DO CEARÁ

- Para forçar a semântica alternativa em Java, o **if** interno é colocado em uma sentença composta, como em:

```
if (soma == 0) {  
    if (cont == 0)  
        resultado = 0;  
  
} else  
    resultado = 1;
```

- A solução acima é usada em C, C++ e C#

ANINHANDO SELETORES



UNIVERSIDADE
FEDERAL DO CEARÁ

- Python

```
if soma == 0:
    if cont == 0:
        resultado = 0
    else:
        resultado = 1
```

CONSTRUÇÕES DE SELEÇÃO MÚLTIPLA



UNIVERSIDADE
FEDERAL DO CEARÁ

- Permite a seleção de uma dentre qualquer número de sentenças ou de grupos de sentenças
- Questões de projeto:
 1. Qual é a forma e o tipo da expressão que controla a seleção?
 2. Como são especificados os segmentos selecionáveis?
 3. O fluxo de execução por meio da estrutura pode incluir apenas um único segmento selecionável?

EXEMPLO DE SELETORES MÚLTIPLOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- C, C++ e Java

```
– switch (expressão) {  
    case expressão_constante_1: sentença_1;  
    ...  
    case constante_n: sentença_n;  
    [default: sentença_n+1]  
}
```

EXEMPLO DE SELETORES MÚLTIPLOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- Escolhas de projeto para o switch de C
 1. A expressão de controle pode ser apenas do tipo inteiro
 2. As sentenças selecionáveis podem ser sequências de sentenças ou blocos
 3. Qualquer número de segmentos pode ser executado em uma execução da construção
 4. O segmento opcional default é usado para valores não representados (se o valor da expressão de controle não é representado e nenhum segmento padrão está presente, a construção não faz nada)

EXEMPLO DE SELETORES MÚLTIPLOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- C#
 - Se difere do C ao ter uma regra de semântica estática que proíbe a execução implícita de mais de um segmento
 - Cada segmento selecionável deve terminar com uma sentença de desvio incondicional explícita (goto ou break)

EXEMPLO DE SELETORES MÚLTIPLOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- Ruby tem duas formas de construções de seleção múltipla
 1. Uma forma se assemelha a uma lista de sentenças if aninhadas

```
leap = case
  when year % 400 == 0 then true
  when year % 100 == 0 then false
  else year % 4 == 0
end
```

2. A outra se assemelha a um switch

```
case in_val
  when -1 then neg_count++
  when 0 then zero_count++
  when 1 then pos_count++
  else puts "Error - in_val is out of range"
end
```

SELEÇÃO MÚLTIPLA USANDO IF



UNIVERSIDADE
FEDERAL DO CEARÁ

- Seletores múltiplos podem aparecer diretamente como extensões de seletores de dois caminhos, usando cláusulas else-if

```
if (cont < 10) {  
    bag1 = 1;  
} else {  
    if (cont < 100) {  
        bag2 = 1;  
    } else {  
        if (cont < 1000) {  
            bag3 = 1;  
        } else {  
            bag4 = 1;  
        }  
    }  
}
```

```
if (cont < 10) {  
    bag1 = 1;  
} else if (cont < 100) {  
    bag2 = 1;  
} else if (cont < 1000) {  
    bag3 = 1;  
} else {  
    bag4 = 1;  
}
```


SELEÇÃO MÚLTIPLA USANDO IF



UNIVERSIDADE
FEDERAL DO CEARÁ

- Por exemplo, em Python:

```
if cont < 10 :  
    bag1 = True  
elif cont < 100 :  
    bag2 = True  
elif cont < 1000 :  
    bag3 = True  
else:  
    bag4 = True
```

SENTENÇA DE ITERAÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Sentenças que fazem com que uma sentença ou uma coleção de sentenças seja executada zero, uma ou mais vezes. Uma construção de iteração é frequentemente chamada de um laço
- Questões de projeto:
 - Como a iteração é controlada?
 - Onde o mecanismo de controle deve aparecer na construção de laço?

LAÇOS CONTROLADOS POR CONTADOR



UNIVERSIDADE
FEDERAL DO CEARÁ

- Uma sentença de controle iterativa de contagem tem uma variável de laço, que inclui os valores inicial e final e o tamanho do passo
- Questões de projeto:
 - Qual é o tipo e o escopo da variável de laço?
 - Deve ser legal para a variável ou para os parâmetros de laço serem modificados nele, e, se isso for possível, essa mudança afeta o controle do laço?
 - Os parâmetros de laço devem ser avaliados apenas uma vez ou uma vez para cada iteração?

EXEMPLOS DE SENTENÇAS DE ITERAÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Sintaxe de FORTRAN 95

DO rótulo variável = inicial, final, [tamanho do passo]

comandos

rótulo continue

- Tamanho do passo pode ser qualquer valor, menos zero

- Parâmetros podem ser expressões

- Questões de projeto:

- A variável de laço deve ser do tipo INTEGER

- A variável de laço não pode ser mudada no laço, mas os parâmetros podem; porque eles são avaliados apenas uma vez, isso não afeta o controle do laço

- Parâmetros do laço são avaliados apenas uma vez

EXEMPLOS DE SENTENÇAS DE ITERAÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Linguagens baseadas em C

for ([expr_1] ; [expr_2] ; [expr_3]) corpo

- O corpo do laço pode ser uma única sentença, uma sentença composta ou uma sentença nula
- Se a segunda expressão está ausente, é um laço infinito
- As três partes que compõem o laço não são obrigatórias

- Escolhas de projeto:

- Não há nenhuma variável de laço explícita
- Tudo pode ser mudado no laço
- A primeira expressão é avaliada uma vez, mas as outras duas são avaliadas com cada iteração

EXEMPLOS DE SENTENÇAS DE ITERAÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- C++, Java e C# se diferem de C de duas maneiras:
 - A expressão de controle pode ser booleana
 - A primeira expressão pode incluir definições de variáveis (o escopo de uma variável definida na sentença for é a partir de sua definição até o final do corpo do laço)

EXEMPLOS DE SENTENÇAS DE ITERAÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Python

```
for variável_do_laço in objeto:  
    corpo do laço  
[else:  
    cláusula senão]
```

- O objeto é frequentemente uma faixa, uma lista de valores em colchetes ([2, 4, 6]), ou uma chamada à função range (range(5), que retorna [0, 1, 2, 3, 4])
- A variável de laço assume os valores especificados na faixa dada, um para cada iteração
- A cláusula else, que é opcional, é executada se o laço termina normalmente

SENTENÇAS DE ITERAÇÃO: LAÇOS CONTROLADOS LOGICAMENTE



UNIVERSIDADE
FEDERAL DO CEARÁ

- Controle de repetição é baseado em uma expressão booleana
- Questões de projeto:
 - O controle deve ser de pré ou pós-teste?
 - O laço controlado logicamente deve ser uma forma especial de um laço de contagem ou uma sentença separada?

SENTENÇAS DE ITERAÇÃO: LAÇOS CONTROLADOS LOGICAMENTE



UNIVERSIDADE
FEDERAL DO CEARÁ

- C e C++ incluem tanto laços controlados logicamente com pré-teste quanto com pós-teste:

```
while (ctrl_expr)
    corpo do laço
```

```
do
    corpo do laço
while (ctrl_expr)
```

- Ada tem um laço lógico com pré-teste, mas nenhuma versão pós-teste
- FORTRAN 95 não tem um laço lógico, nem com pré-teste, nem com pós-teste
- Perl e Ruby têm dois laços lógicos com pré-teste: while e until. Perl também tem dois laços com pós-teste

MECANISMOS DE CONTROLE DE LAÇOS POSICIONADOS PELO USUÁRIO



UNIVERSIDADE
FEDERAL DO CEARÁ

- C , C++, Python, Ruby e C# têm saídas não rotuladas incondicionais (break)
- Java e Perl têm saídas incondicionais rotuladas (break em Java, last em Perl)
- C, C++ e Python incluem uma sentença de controle não rotulada, continue, que transfere o controle para o mecanismo de controle do menor laço que o envolve
- Java e Perl têm sentenças similares ao continue

ITERAÇÃO BASEADA EM ESTRUTURAS DE DADOS



UNIVERSIDADE
FEDERAL DO CEARÁ

■ PHP

- current aponta para o elemento que foi acessado pela última vez pelo iterador
- next move current para o próximo elemento na matriz
- reset move current para o primeiro elemento

```
$alunos = ["José", "João", "Anna"];  
echo current($alunos); // retorna José  
echo next($alunos); // retorna João  
echo next($alunos); // retorna Anna  
echo reset($alunos); // retorna José
```

ITERAÇÃO BASEADA EM ESTRUTURAS DE DADOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- Java
 - Para qualquer coleção que implementa a interface `Iterator`
 - `next` move o ponteiro para a coleção
 - `hasNext` é um predicado
 - `remove` deleta um elemento
- Perl tem um iterador embutido, `foreach`

ITERAÇÃO BASEADA EM ESTRUTURAS DE DADOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- Java 5.0 (usa for, embora chamado de foreach)
 - Para matrizes ou outra classe que implemente a interface Iterable, por exemplo, ArrayList
for (String myElement : myList) { ... }
- Sentença foreach de C# itera nos elementos de matrizes e de outras coleções:
Strings[] = strList = {"Bob", "Carol", "Ted"};
foreach (Strings name in strList)
 Console.WriteLine ("Name: {0}", name);
 - A notação {0} indica a posição na cadeia a ser mostrada

ITERAÇÃO BASEADA EM ESTRUTURAS DE DADOS



UNIVERSIDADE
FEDERAL DO CEARÁ

■ Python

```
alunos = [{  
    "nome": "José", "nota": 8.9  
}, {  
    "nome": "João", "nota": 6.5,  
}, {  
    "nome": "Anna", "nota": 9.9  
}]  
  
for indice, aluno in enumerate(alunos):  
    print("Aluno {0}: nome: {1}, nota: {2}".format(  
        indice+1, aluno['nome'], aluno['nota']  
    ))
```

DESVIO INCONDICIONAL



UNIVERSIDADE
FEDERAL DO CEARÁ

- Transfere o controle da execução para uma posição especificada no programa
- Maior preocupação: legibilidade
- Algumas linguagens não têm suporte para goto (por exemplo, Java)
- C# oferece goto (pode ser usado em sentenças switch)

ATIVIDADE



UNIVERSIDADE
FEDERAL DO CEARÁ

- 1) Explique a diferença entre palavras reservadas e palavras-chave
- 2) Defina vinculação estática e vinculação dinâmica
- 3) Quais as vantagens e desvantagens de declarações implícitas?
- 4) Explique quais são as questões de projetos relativas aos tipos de cadeia de caracteres
- 5) Defina o que são tipos ordinais, enumeração e subfaixa
- 6) Quais são as vantagens dos tipos de enumeração definidos pelo usuário?
- 7) Explique os tipos de matrizes quanto a sua alocação.
- 8) Apresente um código para criar uma matriz irregular (Escolha uma linguagem)
- 9) Defina união livre e união discriminada
- 10) Quais são os argumentos favoráveis e contrários à utilização de variáveis de referência em Java e de ponteiros em C