UFRPE
Universidade
Federal Rural
de Pernambuco

Paradigmas de Programação

Estruturas de Controle

Kellyton Brito
kellyton.professor@gmail.com
21/11/2017

Introdução

- Linguagens imperativas: baseadas em atribuições de valores a variáveis

- Possuem dois mecanismos adicionais:
  - Controles de fluxo alternativos
  - Execução repetida
- Chamados de Instruções de Controle

- Afetam bastante as capacidades de leitura e escrita

Três tipos de estruturas

- **Instruções de Seleção**
- Instruções Iterativas
- Ramificações Não Condicionais

Instruções de Seleção

- Mecanismo de escolha entre dois ou mais caminhos de execução
- Dois tipos de seleção
  - *two-way* (dois caminhos);
  - *multiple-way ou n-way* (multiplos caminhos)

- Forma básica

If (expressao_controle)
then clausula
else clausula

Instruções de Seleção: *two-way* Selection

- Forma da expressão de controle:
  - Normalmente entre parênteses ou uso do ***then***
- Forma da Clausula:
  - Alguma linguagens usam palavras reservadas (begin, end)
  - Normalmente chaves delimitam inicio e fim
  - Delimitadores evitam ambiguidade

```
if (sum == 0)
  if (count == 0)
    result = 0;
else
    result = 1;
```

```
if (sum == 0) {
  if (count == 0)
    result = 0;
}
else
  result = 1;
```

Instruções de Seleção: *n-way* Selection

- Permite a seleção de uma entre um grupo de opções
  - Evita muitas expressões de seleção aninhadas
- Questões de design:
  - Forma e tipo da expressão de controle
  - Como os segmentos são especificados
  - O que fazer quando nenhuma opção foi escolhida?

```
switch(i){
    case 1:
        a++;
        break;
    case 2:
        b++;
        break;
    case 3:
        c++;
    case 4:
        d++;
        break;
    default:
        e++;
}
```

Instruções de Seleção: *n-way* Selection

O que acontece quando i recebe 1 e 5, no lugar da ???

```
public void switchCase(){
    int i = ???;

    switch (i){
        case 1:
            System.out.println("1");
        case 2:
            System.out.println("2");
            break;
        case 3:
            System.out.println("3");
        case 4:
            System.out.println("4");
        default:
            System.out.println("Padrao");
    }
```

A) Dá erro de compilação nos dois casos
B) Dá erro de compilação em um dos dois casos
C) Dá erro durante a execução nos dois casos
D) Dá erro durante a execução em um dos dois casos
E) Imprime *1 e Padrao,* respectivamente quando recebe 1 e 5.
F) Nenhuma das anteriores

Imprime 1 e 2, e Padrao, respectivamente

Três tipos de estruturas

- Instruções de Seleção
- **Instruções Iterativas**
- Ramificações Não Condicionais

Instruções Iterativas

- *Loops:* executam repetidamente
- Inicialmente utilizados para iteração sobre os arrays

- Questões de design:
  - Como a iteração é controlada
  - Onde o mecanismo de controle aparece? (início ou fim)

Instruções Interativas

- Loops controlados por contadores: FOR

```
for (inicializacao; controle; incremento){
  ...
}
```

```
For (int i = 0; i < 5; i++){
    ...
}
```

Instruções Interativas

- Loops controlados por contadores: FOR
- Versão “moderna”

for (Tipo objeto: lista){

...

}

```
for (Aluno a: alunos){
        a.addPresenca("27/12/2013");
}
```

Instruções Iterativas

- Loops logicamente controlados: WHILE, DO WHILE
  - O *do-while* sempre executa pelo menos uma vez

```
while (controle){
    ...
}
```

```
do {
    ...
} while (controle);
```

Instruções Interativas

- Mecanismos de controle inseridos pelo usuário
  - Break: encerra o loop

```
while (sum < 1000) {
   getnext(value);
   if (value < 0) break;
   sum += value;
}
```

Pára tudo

  - Continue: encerra aquela iteração

```
while (sum < 1000) {
   getnext(value);
   if (value < 0) continue;
   sum += value;
}
```

Apenas não soma

Três tipos de estruturas

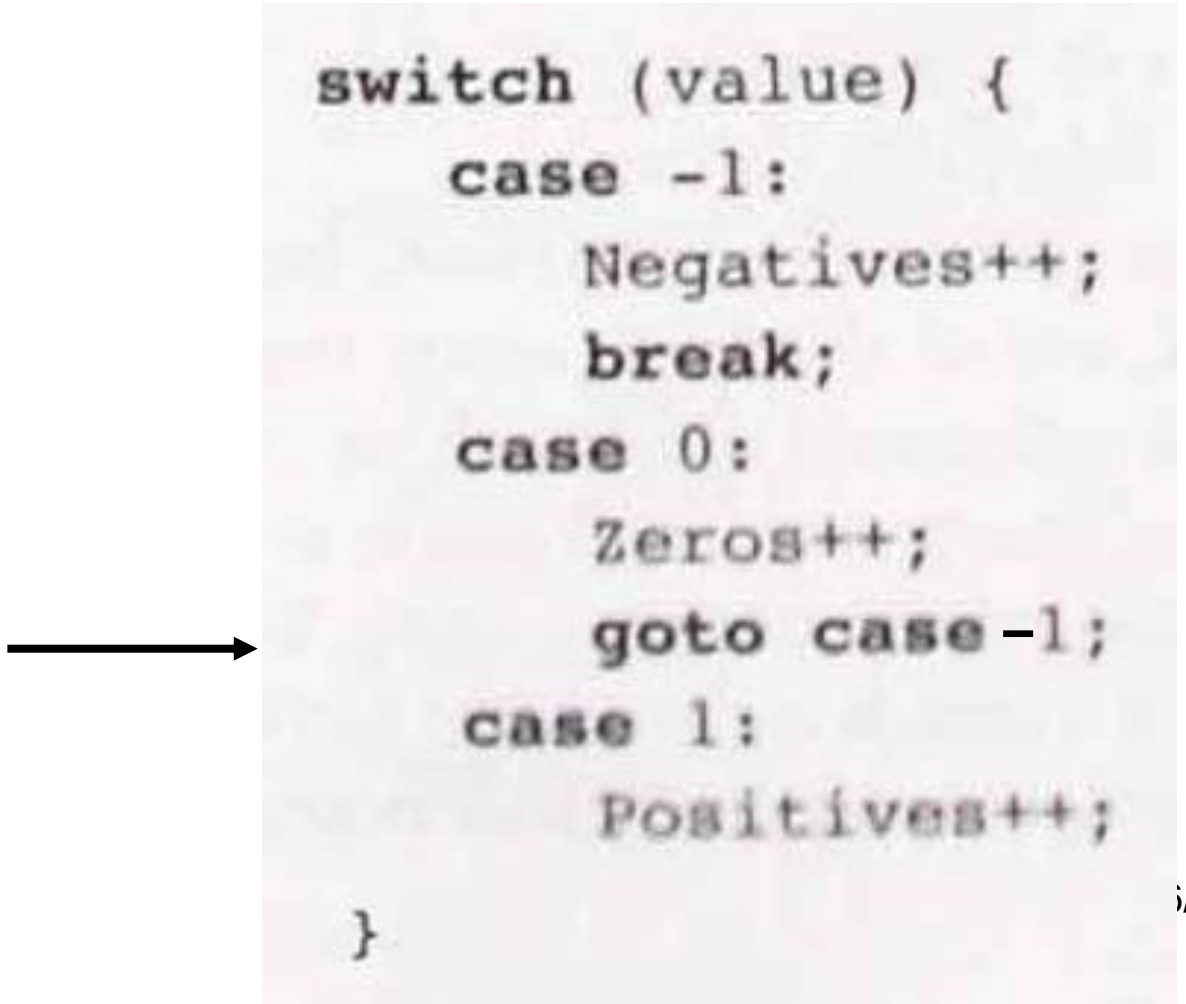
- Instruções de Seleção
- Instruções Iterativas
- **Ramificações Não Condicionais**

Ramificações Não Condicionais

- Transferem a execução do programa para uma localização específica: *goto*
- Dificulta leitura e escrita
- Uso praticamente banido
- Substituído pela combinação de seleções e iterações

Ramificações Não Condicionais

- Uso do goto em C#
  - switch/case:



```
switch (value) {
    case -1:
        Negatives++;
        break;
    case 0:
        Zeros++;
        goto case -1;
    case 1:
        Positives++;
}
```

UFRPE
Universidade Federal Rural de Pernambuco

Dúvidas quanto a estruturas de controle?

Sebesta – Conceitos de Linguagens de Programação. Até o capítulo 8