

Questões

4.10 Compare e contraste a instrução de seleção única `if` e a instrução de repetição `while`. Qual é a semelhança dessas duas instruções? Qual é a diferença?

Ambos os tipos de instruções avaliam expressões lógicas para a execução de uma ou mais linhas de código, quando as expressões lógicas contêm resultado falso, tanto a instrução de seleção quanto a instrução de repetição, não executam seus trechos de código. A diferença se apresenta quando a expressão lógica apresenta resultado verdadeiro, na instrução de seleção o bloco de código é apenas executado uma vez, já na instrução de repetição, o trecho de código pode ser executado mais vezes, caso o resultado da expressão lógica continue sendo verdadeiro.

4.11 Explique o que acontece quando um programa Java tenta dividir um inteiro por outro. O que acontece para a parte fracionária do cálculo? Como você pode evitar esse resultado?

O Java entende que o resultado também é inteiro, por isso a parte fracionária é perdida. Para evitar esse problema é necessário usar o operador unário de coerção (`double`) antes da divisão, assim é salvo temporariamente o resultado como ponto flutuante, para assim poder ser salvo em uma variável posteriormente.

4.12 Descreva as duas maneiras como as instruções de controle podem ser combinadas.

Pode-se aninhar instruções de seleção, assim será feita outra seleção quando o resultado da primeira for verdadeiro. Também é possível aninhar instruções de seleção e de repetição ou vice-versa, para executar uma repetição caso o resultado da seleção for verdadeiro ou repetir uma seleção caso a expressão lógica da repetição seja verdadeira.

4.13 Que tipo de repetição seria apropriado para calcular a soma dos primeiros 100 inteiros positivos? Que tipo seria apropriado para calcular a soma de um número arbitrário de inteiros positivos? Descreva brevemente como cada uma dessas tarefas poderia ser realizada.

No caso da soma dos 100 primeiros inteiros positivos, seria interessante uma repetição controlada por contador, pois se sabe o número de vezes em que o laço terá que se repetir, assim seria necessário criar uma variável contadora e ir incrementando cada vez que o laço fosse executado. Já no segundo caso, não se sabe quantos números serão somados, assim seria interessante criar uma variável sentinela e controlar a execução do laço por ela, perguntando ao usuário se ele deseja continuar a operação, caso sim, não modificar a variável da sentinela, caso não, modificar o valor da variável sentinela de forma que assim possa sair do laço. Note que neste caso é necessário usar uma instrução de seleção para alterar ou não o valor da variável sentinela.

4.14 Qual é a diferença entre pré-incrementar e pós-incrementar uma variável?

Ao se usar o pré-incremento em uma variável, seu valor é modificado antes do seu uso, já no pós-incremento o valor da variável é modificado após o seu uso. Quando é feito isoladamente, tanto o pré quanto o pós incremento tem o mesmo efeito, porém é necessário ter cuidado ao usá-los em expressões aritméticas, pois podem causar um resultado inesperado.

4.15 Identifique e corrija os erros em cada um dos seguintes fragmentos de código. [Observação: pode haver mais de um erro em cada trecho de código.]

a) `if (age >= 65);`
 `System.out.println("Age is greater than or equal to 65");`
 `else`
 `System.out.println("Age is less than 65");`

`if(age >= 65)`
 `System.out.println("Age is greater than or equal to 65");`
`else`
 `System.out.println("Age is less than 65");`

b) `int x = 1, total;`
 `while (x <= 10)`
 {
 `total += x;`
 `++x;`
 }

`int x = 1, total = 0;`
`while (x <= 10)`
{
 `total += x;`
 `++x;`
}

c) `while (x <= 100)`
 `total += x;`
 `++x;`

`int x = 1, total = 0;`
`while (x <= 100)`
{
 `total += x;`
 `++x;`
}

d) `while (y > 0)`
{
 `System.out.println(y);`
 `++y;`

`int y = 10;`
`while (y > 0)`
{
 `System.out.println(y);`
 `--y;`
}