





Sun Certified Programmer for the Java 2 Platform

Fernando Anselmo

fernando.anselmo@dfjug.org

Programa

dia 1 – Fundamentos da Linguagem e Garbage Collection

dia 2 – Operadores e Assinaladores

dia 3 – Declarações e Controle de Acesso

dia 4 – Conversão e Casting

dia 5 – Fluxo de Controle, Exception e Assertions

dia 6 – Overloading, Override, Tipo de Runtime e OO

dia 7 – Threads

dia 8 – Classes fundamentais (Math, String, Wrapper)

dia 9 – Coleções

dia 10 – Questões “Bizarras”

Dia 2

Operators and Assignments

- Determinar o resultado da aplicação de quaisquer operadores, incluindo assinaturas de operadores e instanceof, para operandos de qualquer tipo, classe, escopo, ou acessibilidade e qualquer combinação.
- Determinar resultados de operações lógicas, métodos para os objetos de qualquer combinação entre as classes `java.lang.String`, `java.lang.Boolean` e `java.lang.Object`.
- Expressões envolvendo operadores `&`, `|`, `&&`, `||` e variáveis.

- Conversão de base 10 para base 2

1. Crie uma régua inversa a partir de 1 sempre dobrando o valor (... 128 64 32 16 8 4 2 1)
2. Exemplo vamos converter 53 em binário
3. **53** - 128 (0) = número negativo
53 - 64 (0) = número negativo
53 - 32 (1) = 21
21 - 16 (1) = 5
5 - 8 (0) = número negativo
5 - 4 (1) = 1
1 - 2 (0) = número negativo
1 - 1 (1) = **0**
4. Resultado: 00110101

Evaluation Order

- A ordem é sempre da esquerda para a direita

```
int [] a = { 1, 2 }; int b = 1;  
a[b] = b = 0; ==> resp. 1
```

- Parênteses tem prioridade total, mas depois deles a ordem de precedência se faz entre:

1. Unary	++ -- ! ~ (cast)
2. Arithmetic	* / % + -
3. Shift	<< >> >>>
4. Comparasion	< <= > >= instanceof == !=
5. Bitwise	& ^
6. Short-circuit	&&
7. Conditional	?:
8. Assignment	= "op="

The Unary Operators

- Incrementos e Decrementos (esquerda participa):

Inicial x	Expressão	Final Y	Final X
5	<code>y = x++;</code>	5	6
5	<code>y = ++x;</code>	6	6

- Operador de inversão (`~`)

`i = ~10 ==> -11 ==> fórmula: (-x) - 1`

- Operador de negação (`!`)

`if (!true) ==> false`

- Operador de coerção de tipos

`float a = (float)1.2;`

The Arithmetic Operators

- Multiplicação e Divisão

```
int a = 3, b = 5;
```

```
int c = a * a / b;    ==> 1
```

```
int c = b / a * a;   ==> 5
```

- Operador Módulo (%)

```
17 % 5 => 17 - 5 => 12 - 5 => 7 - 5 => 2
```

```
7.6 % 2.9 => 7.6 - 2.9 => 4.7 - 2.9 => 1.8
```

- Adição e Subtração

```
int a = 1, b = 2;
```

```
System.out.println(a + b + " + a + b);    => 312
```

The Shift Operators

- $x \ll y$ - Desloca os bits de 'x' para a **esquerda** pelo número inteiro 'y', preenche sempre com 0s.

Ex: $52 \ll 2 = 208$;

- $x \gg y$ - Desloca os bits de 'x' para a **direita** pelo número inteiro 'y', Se 'x' for negativo preenche com 1s, caso contrário com 0s

Ex: $52 \gg 3 = 6$;

- $x \ggg y$ - Desloca os bits de 'x' para a **direita** pelo número inteiro 'y', 0s são inseridos a partir da esquerda.

valor 52 = 00110100 valor 6 = 00000110 valor 208 = 11010000

The Comparison Operators

- Operadores matemáticos: $<$ (menor), \leq (menor igual), $>$ (maior) e \geq (maior igual)
- **instanceof** – testa se o objeto pertence a classe em questão.
- $==$ (igual) $!=$ (diferente) utilize para atributos

The Bitwise Operators

- **&** - uma coisa e outra

`0 & 0 = 0; 0 & 1 = 0; 1 & 0 = 0; 1 & 1 = 1`

- **^** - uma coisa ou exclusivamente a outra

`0 ^ 0 = 0; 0 ^ 1 = 1; 1 ^ 0 = 1; 1 ^ 1 = 0`

- **|** - uma coisa ou a outra

`0 | 0 = 0; 0 | 1 = 1; 1 | 0 = 1; 1 | 1 = 1`

The Short-Circuit Logical Operators

- **&&** - uma coisa e outra

F && F = F; T && T = T;

F && T = F; T && F = F;

- **||** - uma coisa ou a outra

F || F = F; T || T = T;

F || T = T; T || F = T;

The Conditional Operator

- (condicao)?true:false

```
int x = 4;
```

```
((x > 4) ? 10.0 : 5); ==> Resposta 5.0
```

The Assignment Operators

- `byte x = 2; x = (byte)(x + 3) ==> 5`
- `int a, b, c; a = b = c = 5;`
- `*=, /=, %=, +=, -=, <<=, >>=, >>>=, &=, ^=, |=`

`byte x = 2; x += 3; ==> 5`

`byte x = 2; x = x + 3; ==> ?`

Time for Exercises

