

Programação Orientada a Objetos

Alexandre César Muniz de Oliveira



Dados e Operadores

Parte V



Representação

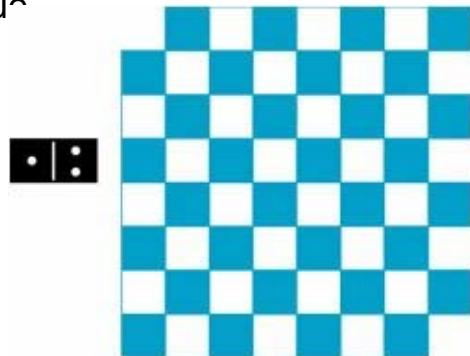


- Programação
 - Solução de problemas em duas fases:
 - Representação (modelagem de dados)
 - Ação (modelagem de código)

Representação



- Um tabuleiro de xadrez, sem os cantos superior-esquerdo e inferior-direito, pode ser completamente ocupado por dominós que cobrem duas casas por vez?





Variáveis Lógicas

- if (boolean expression) statement;

```
boolean isDone = false;
isDone = true;
if (isDone)
    Classe.fazAlgumaCoisa();
else
    Classe.fazOutraCoisa()
```



Variáveis Lógicas

x	y	x && y	x y	x Δ y	!x
true	true	TRue	TRue	false	false
true	false	false	TRue	true	false
false	TRue	false	TRue	true	true
false	false	false	false	false	true

Variáveis Lógicas



`(true || true) && false ==> true && false ==> false`
`true || (true && false) ==> true || false ==> true`

Variáveis Lógicas



Precedência	Operador	Operação
1	()	Parênteses
2	!	NOT
3	\wedge	EXCLUSIVE-OR
4	&&	AND

Variáveis Numéricas



Type	Bits	Range of Values
byte	.8	-128 to +127
short	16	-32768 to 32767
int	32	-2147483648 to 2147483647
long	64	-2^{63} to $2^{63} - 1$
float	32	-3.40292347E + 38 to +3.40292347E + 38
double	64	-1.79769313486231570E + 308 to +1.79769313486231570E + 308

Variáveis Numéricas



- Tamanhos dos tipos de variáveis
 - Java: independe de plataformas
 - C/C++: depende do compilador



Operadores

Operação	Operador	Exemplo
Adição	+	$x + 2$
Subtração	-	$m - 2$
Multiplicação	*	$m * 2$
Divisão	/	x/y
Módulo	%	$x\%y$



Operadores

Expressão	Operação	Interpretação
$j = ++k$	Pré-incremento	$k = k + 1; j = k;$
$j = k ++$	Pós-incremento	$j = k; k = k + 1;$
$j = --k$	Pré-decremento	$k = k - 1; j = k;$
$j = k --$	Pós-decremento	$j = k; k = k - 1;$

Operadores



Operador	Operação	Exemplo	Interpr.
=	Atribuição	m = n;	m = n;
+=	Adição	m += 3;	m = m + 3;
-=	Subtração	m -= 3;	m = m - 3;
*=	Multiplicação	m *= 3;	m = m * 3;
/=	Divisão	m /= 3;	m = m/3;
%=	Resto	m %= 3;	m = m%3;

Operadores



Precedência	Operador	Operação
1	()	Parênteses
2	++ --	Incremento, decremento
3	* / %	Multiplicação, divisão, módulo
4	+ -	Adição, subtração
5	< > <= >=	Relacional
6	= = !=	Igualdade

java.lang.Math



Método	Descrição	Exemplos
int abs(int x)	Absoluto	if $x \geq 0$ abs(x) is x
long abs(long x)		if $x < 0$ abs(x) is x
float abs(float x)		
int ceil(double x)	Teto	ceil(8.3) is 9 ceil(8.3) is 8
int floor(double x)	Piso	floor (8.9) is 8 floor(8.9) is -9

java.lang.Math



Método	Descrição	Exemplos
double log(double x)	Log. Natural	log (2.718282) is 1.0
double pow(double x, double y)	Potência	pow(3, 4) is 81.0 pow(16.0, 0.5) is 4.0
double random()	Número aleatório	random() is 0.5551 random() is 0.8712
long round(double x)	Arredonda	round(26.51) is 27 round (26.499) is 26
double sqrt(double x)	Raiz quadrada	sqrt(4.0) is 2.0

java.lang.Math



```
public final class Math
{ private Math() {}
...
    public static native double sqrt(double a)
        throws ArithmeticException;
...
}
```

- *Public final*
 - Pode ser acessada mas não pode ser especializada (subclasse) por outra
- *Construtor private*
 - Não pode ser instanciada

java.text.NumberFormat



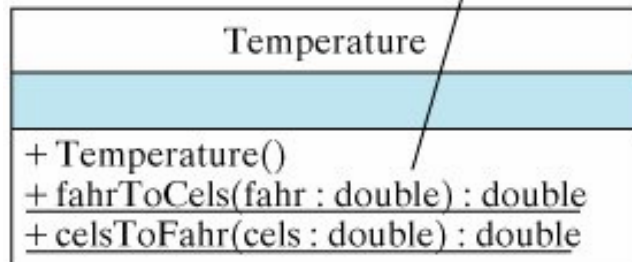
NumberFormat

```
+ getInstance() : NumberFormat
+ getCurrencyInstance() : NumberFormat
+ getPercentInstance() : NumberFormat
+ format(in n : double) : String
+ format(in n : long) : String
+ getMaximumFractionDigits() : int
+ getMaximumIntegerDigits() : int
+ setMaximumFractionDigits(in n : int)
+ setMaximumIntegerDigits(in n : int)
```

Exemplo



$$F = C * 9/5 + 32$$
$$C = (F - 32) * 5/9$$



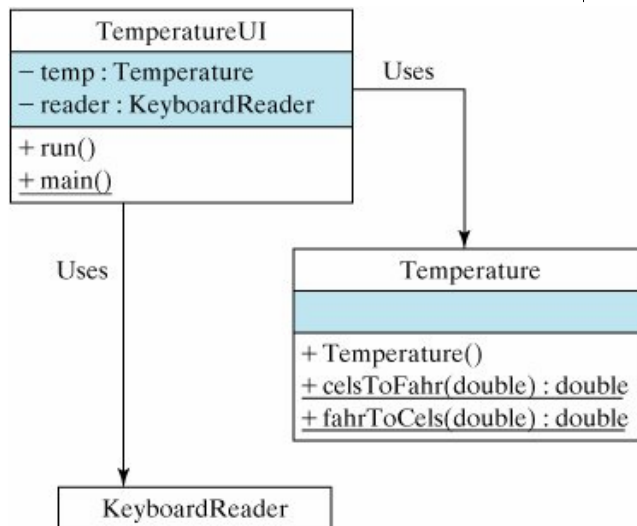
Exemplo



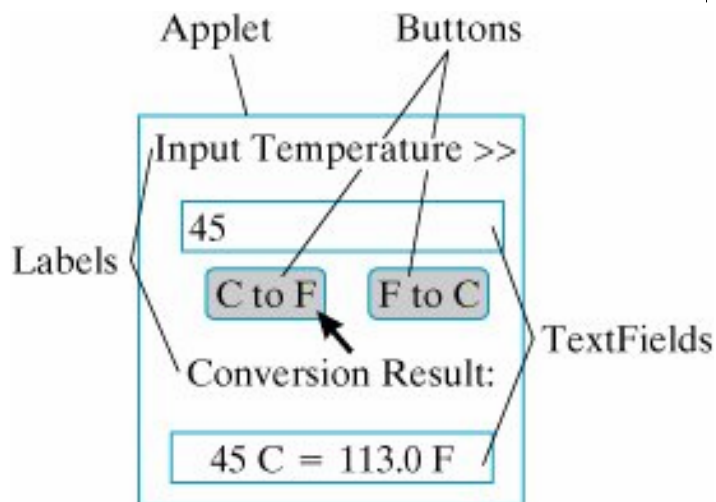
```
public class Temperature
{
    public Temperature() {}

    public static double fahrToCels(double temp)
    {
        return (5.0 * (temp - 32.0) / 9.0);
    }
    public static double celsToFahr(double temp)
    {
        return (9.0 * temp / 5.0 + 32.0);
    }
} // Temperature class
```

Exemplo



Exemplo



Variáveis Caracteres



- Abrangência
 - Inteiro sem sinal de 16-bits.
 - Total of $2^{16} = 65536$ caracteres Unicodes
 - <http://www.unicode.org/>
 - Valores de 0 to 65535.

Variáveis Caracteres



```
Code 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
Char SP! " # $ % & ' ( ) * + , - . /
Code 48 49 50 51 52 53 54 55 56 57
Char 0 1 2 3 4 5 6 7 8 9
Code 58 59 60 61 62 63 64
Char : ; < = > ? @
Code 65 66 67 68 69 70 71 72 73 74 75 76 77
Char A B C D E F G H I J K L M
Code 78 79 80 81 82 83 84 85 86 87 88 89 90
Char N O P Q R S T U V W X Y Z
Code 91 92 93 94 95 96
Char [ \ ] ^ - '
Code 97 98 99 100 101 102 103 104 105 106 107 108 109
Char a b c d e f g h i j k l m
Code 110 111 112 113 114 115 116 117 118 119 120 121 122
Char n o p q r s t u v w x y z
```



Variáveis Caracteres

```
public class Test {  
    public static void main(String argv[]) {  
        char ch = 'a';  
        int k = (int)'b';  
        System.out.println(ch);  
        System.out.println(k);  
        ch = (char)k;  
        System.out.println(ch);  
        System.out.println(toUpperCase('a'));  
        System.out.println(toUpperCase(ch));  
        System.out.println(digitToInteger('7'));  
    }  
}
```



Variáveis Caracteres

```
public static char toUpperCase(char ch) {  
    if ((ch >= 'a') && (ch <= 'z'))  
        return (char)(ch - 32);  
    return ch;  
}  
public static int digitToInteger(char ch) {  
    if ((ch >= '0') && (ch <= '9'))  
        return ch - '0';  
    return -1 ;  
}  
} // Test class
```