

PROGRAMAÇÃO ORIENTADA A OBJETOS

Professor Manoel Carvalho Marques Neto
Email: manoelnetom@gmail.com

// O a

... Ga a 100 a a a a c ...

// E a a

... Ga a 500 a a a ...

b a a a a a b a a a . P a

Na ()

E a ()

Na A S E a .

Classes Além da Programação Estruturada

P a a a a a a a a a a a a a a b a

a a a a a E a a a a a a a a a a a a a

a a a a a a a a a a a a a a a a a a a

aba a a a a . P a a a a a a a a a a a a a

? M a a a a a a a a a a a a a a a a b

a a a a a a a a a a a a a a a a a a a

b . a a a a a a a a a a a a a a a a a a

M a a a a a a a a a a a a a a a a a a a

a a . E C a a a a a a a a a a a a a a a

C a a a a a a a a a a a a a a a a a a a

ab a a a a a a a a a a a a a a a a a a

a a a a a a a a a a a a a a a a a a a

. A ab a a a a a a a a a a a a a a a

C++ a a a a a a a a b . E a a a a a a a a

C. E a a a a a a a a a a a a a a a a a

a a C++ a a a a a a a a a a a a a a a

Ja a b a a a a a a a a a a a a a a a a

a a a a a a a a a a a a a a a a a a a

a a a a a a a a a a a a a a a a a a a

a a a a a a a a a a a a a a a a a a a

a a a a a a a a a a a a a a a a a a a

a a a a a a a a a a a a a a a a a a a

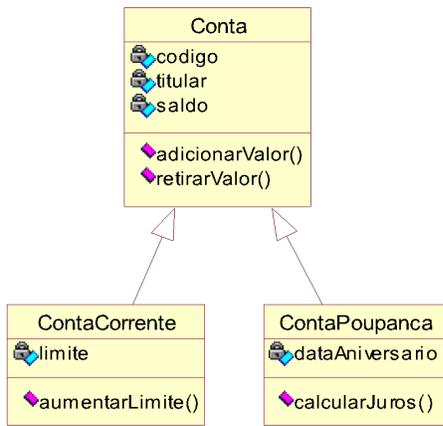
Encapsulamento

Encapsulamento é a técnica de ocultar os detalhes de implementação de uma classe, deixando apenas a interface pública para o usuário. Isso é feito através de atributos e métodos privados, protegendo o código de alterações não autorizadas e promovendo a segurança e a modularidade.

Por exemplo, em uma classe `Conta`, os atributos `codigo`, `titular` e `saldo` podem ser privados, enquanto os métodos `adicionarValor()` e `retirarValor()` são públicos. Isso garante que o estado interno da conta seja controlado apenas pelos métodos definidos.

Herança

Herança é a capacidade de uma classe (filha) herdar as propriedades e métodos de outra classe (pai). Isso permite a reutilização de código e a organização hierárquica das classes. No exemplo, `ContaCorrente` e `ContaPoupanca` herdam de `Conta`.



a a a A a(). C a
a F a b a a . T a a a a a a

O que é JAVA?

a a a a a a a a a b a S
1 . M a a C++, a a Ja a a a a,
a a a a a a a a a a a
a, a E a ab a b a a a a a a,
a a a b - . N
a a a a a a a a a

Declarando variáveis

```
A a a a :
:
S a ;
b a ;
O a a a a , b a a , ( ).
E a a . D a a - a
```

Tipos de variáveis

```
T a a :
a a :
1) U a a a b a
2) O a a a
3) U a a
V a b a a a . O a b
: , - a , a a b a ( a a ).
```

```
T I :
T Ta a A a
b 8 b -128 a 127
16 b -32.768 a 32.767
32 b -2.147.483.648 a 2.147.483.647
64 b -9223372036854775808 a 9223372036854775807
E - a : a (32 b , ) b
(64 b a).
```

Atribuições a variáveis

```
A a a a a a a b a a = :
a = 18;
```

Comentários

```
Ja a a : /* */ C C++. T
/* E a a a a
*/
D a ba a (//) a b a a a a a a a :
a ; // a a a a a
```

(// . a a . c).

Caracteres especiais

Ca a \ S a
 \ N a L a
 \ Ta
 \ b Ba a
 \ R Ca
 \ F (a a a a a a a)
 \\ Ba a a
 \ A
 \ A a c
 \ O a
 \ H a a

Operadores Aritméticos

O a S c E R a
 + a a 3 + 4 7
 b a 5 7 -2
 c a 5 * 5 25
 / 14 / 7 2
 % () 20 % 7 6

E A :
 a A T
 b a a (S a [])
 = 6;
 = 4;
 a a = 12.5 ;
 a b = 7 ;

S . . (+ + , +);
S . . (+ = + (+));
S . . (- = + (-));
S . . (/ = + (/));
S . . (% = + (%));

S . . (a + a + , b + b);
S . . (a / b = + (a / b));

A a a a a a :

6, 4
+ = 10
- = 2
/ = 1
% = 2
a 12.5, b 7
a / b = 1.78571

Mais sobre atribuições

Va a b a a :
, ;
= = 0;

N a a b a 0;

Operadores de Atribuição

E S a
+= = +
-= = -
*= = *
/= = /

Incrementos e decrementos

C C C++ Ja a a b :
= ++;
= -- ;

A a a a (++ , --) a b a a a
. Q a , a (++) a , b a

Comparações

Ja a a a a a a . T a a
a a b a (a) .

Operadores de comparação

O a S a E
== I a == 3
!= D (N a) != 3
< M < 3
> Ma > 3
<= M a <= 3
>= Ma a >= 3


```

( )
a a ;
a ;
a a ;
a ;
a a : ;
a b ;
: a ;
O a a a a . S a b a
a a a b a a O a a a a a
a a a a , b a a a a.

```

Looping For

```

O Ja a a :
a a ; ; )
b a ;
V a b a . E : a , a a a
S A a [] = S [10];
( =0; < A a . ; ++ )
A a [] = ;
I a a a a 10 ;

```

Loop While

```

D a a a , a a a
( )
b a ;
A a b a . E :
=0;
( <a a 1. && a : 1[ ]!=0)
a a 2[ ]=( a ) a a 1[ ] ++);

```



```

...
:
a M b DaFa a
a S b ;//
S a N ;//
a a
a
...

```

Exemplo de uso:

```

M b DaFa a a a = M b DaFa a(); a a. b N = M a ;
M b DaFa a = M b DaFa a();
S . . ( . b );

```

S a a **Mendonça** a a , b a a , a a s a ic. a b
b . Ma a a a a b a a a

Definição de Métodos

```

A
O ;
O b ;
U a a ;
O ;
A b a a a a :

```

tipoderetorno nomedometodo(tipo1 arg1, tipo2 arg2, ...)

```

....
:
[] a Ra ( , ) ...
Ra Ca a :
a Ra Ca
[] a Ra ( , )
a [] = [ ( - ) + 1];
( =0; <a . ;++)
a [] = ++;
a ;

```

```

b | a | a (S a [])
  |   |   |
  |   |   |
Ra Ca Ra = Ra Ca ();
A a = Ra . a Ra (1,10);

S . . (T a a : [ ]);

```


R a aR a (P E , P D)

1 = E . ;
1 = E . ;
2 = D . ;
2 = D . ;
;

R a aR a (P E , a a , a a)

1 = E . ;
1 = E . ;
2 = (1 + a a);
2 = (1 + a a);
;

aR a ()
S . . (R : < + 1 + , + 1);
S . . (, + 2 + + 2 + >);
b a a (S a [])
R a a = R a ();
S . . (C a a aR a a a 25, 25, 50, 50 :
);
a . . aR a (25, 25, 50, 50);
a . . aR a ();
S . . (-----);
S . . (C a a aR a (10, 10) , (20, 20)
:);
a . . aR a (P (10,10) , P (20, 20));
a . . aR a ();
S . . (-----);
S . . (C a a aR a (10, 10) , a a
(50) a a (50) :);
a . . aR a (P (10,10) , 50, 50);
a . . aR a ();
S . . (-----);

Métodos Constructores

U a a a a . a a a a a a a a a a ;
D a a a a a a a a a a a a a a a a ;
a b a a , a a a a a a a a a a a a a a a a ;

Constructores Básicos

```
O a P a  
C  
C  
E :  
a P a  
S ;  
a ;  
P a ( S , )  
a = ;  
a = ;  
P a ()  
S . . ( O : + );  
S . . ( . E : + a + a );  
b a a ( S a [] )  
P a ;  
= P a ( Ma a , 20 );  
P a ();
```

Polimorfismo de Constructores

```
I a a P a a b a a R a  
a a a a . S a a a a a : R a  
a a a . P ;  
a R a  
1 = 0;  
1 = 0;  
2 = 0;  
2 = 0;  
R a ( 1, 1, 2, 2 )  
. 1 = 1;  
. 1 = 1;  
. 2 = 2;  
. 2 = 2;
```

```

R a (P E , P D )
1 = E . ;
1 = E . ;
2 = D . ;
2 = D . ;

```

```

R a (P E , a a, a a)
1 = E . ;
1 = E . ;
2 = ( 1 + a a);
2 = ( 1 + a a);

```

```

S . . ( R a : < + 1 + , + 1 );
S . . ( , + 2 + , + 2 + > );
b a a ( S a [] )
R a a ;
S . . ( R a a a 25, 25, 50, 50 : );
a = R a (25, 25, 50, 50 );
a . aR a ();
S . . ( ----- );
S . . ( R a (10, 10) , (20, 20) : );
a = R a ( P (10,10) , P (20, 20) );
a . aR a ();
S . . ( ----- );
S . . ( R a (10, 10) , a a (50) a a (50)
: );
a = R a ( P (10,10) , 50, 50);
a . aR a ();
S . . ( ----- );

```

Métodos Destructores

```

O b a a . O a a a a a a a a a a a a a a a a
a b a a b a a a a a a a a a a a a a a a a
... a ()
D b . a a a a a a a a a a a a a a a a

```


N

S

f

A

E

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

,

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

```

Ca 1 1 = Ca 1();
1. aC a ();
1. aO C a ();

Ca 1 2 = Ca 1();
2. aC a ();
2. aO C a ();

Ca 1 3 = Ca 1();
3. aC a ();
3. aO C a ();

Ca 1 4 = Ca 1();
4. aC a ();
4. aO C a ();

```

```

a a a a a a
a a a 1 a a
C a a a 1
a a a 2
C a a a 1
a a a 3
C a a a 1
a a a 4
C a a a 1
a a a " a "
Ca 1, a 1, a , a , a a , a a a a a b a
, , , a - a , a a a a a a a a :
/T S a a a
a C 2
// E a a a a (S )
S . . ( );
// R a a a a( 1, 2)
b a a( 1 * 2);

```


E E E
E E E
a (N E F E)
// a a a
= b . a E ("J R b ");

E 2
E E E
E ()
E = b . a E ("J R b ");
(// a a a a

N 1, a b a a a a; a b a, 2, a
N E F E a a a;


```

    a a .E      a b a      a a a
    a a      a a      a
- a a .E      a b a      a
  a a      a a
4) L a a      E      a a a      a a a      a . S a
L a a      a      a a      a
C a :      a a
      < - a      >
E :      ( V      a a      -1      O O B      E      ( );
      (      A a I      O O B      E      ( );
5) C a      a      E      a a a      a
      O      a      E      a a a      a
      :      M a E      a      E
      //      a      a
      ( a a E      a )
      M a E      a ( );

```

O pacote java.util

```

    a a      b      a a      a      a a      a      a      a a
    a a      - a      a a      a      a a      a a      a /      a a
    a      a      a      a      a      a      a      A a      a a
A      Ja a 2      a      a      a      a      a      a a a .
a ,      b      b      a ,      a      a      a

```

As coleções antigas

```

A API      Ja a 2      a      a      a      a a a      a a      a a
      . V      a      Ja a      a :
      a      ab
      a a
      B S

```

As novas coleções

A API

Set
Map

Ca

Ob

Set

U S (C

D a

ab a

a (Ob

Q a

C/C++

List

U L (a)

A

A

Map

Ma

Por que usar Collections em vez de arrays?


```
S . . . ( . . . A ());  
//F  
Ja a 2 a a V a a a V a a a .  
.E  
Usando Hashtables  
Ja ab a a .E a V a Ba a C a a  
a a / a a a a .A a a ab a a  
a a V a . U a a ab a a  
BitSet  
J B S a a a a B N a a  
a a " " .E a a S Ja a 2. a a ba a a  
b a b a U B S a a a a a a b  
a a a a B a a / a . a a a  
D a B E "T Ja a"  
" a a a a a ; a a ."  
a , a a a a a a a ."
```



```

Ca F I S a F O $ a F R a F W , I S a R a
A a a a a a a a a a a a a a a a a a a a a
A a a a a a a a a a a a a a a a a a a a a
b S a a a F a a a a a a a a a a a a a a a
I S a R a a a a a a a a a a a a a a a a a .

```

```

E :E a a A - F O S a
F = F (":\ " );
. ( );
F a = F ( , " . " );
F O S a = F O S a ( a );
. ( b [ ] ' ' ' ' ' ' );
F b = F ( , " b " );
b . ( );
S [ ] a = ( )
( =0; <a . ; ++ )
F = F ( , a [ ] );
S . . ( . Ab Pa ( ) );
(a . ( ) )
a . ( ); //O b a a a
. ( );

```

```

E :E a a A - F O S a
S . . ("D " );
F O S a 0 = F O S a (":\Sa a0. " );
b a = (b )S . a ( );
(a!='\ ' )
0. (a);
a=(b )S . a ( );

```

```

E :L a a A - F I S a
F a = F (":\ aA . " );
F I S a = F I S a a );
I S a R a = I S a R a ( );
B a b = B R a ( );
b a a = ; S a ;
( a )
a = b . a L ( );
a == )
a = a ;

```

S . . (a);

```
b. ();
. ();
```

```
E :E a a A -F R a
L a a A -F W
```

```
A a a a a a F R a
B R a .Pa a a a , F W P W .
```

```
E :L a a A -F R a
```

```
F R a = F R a (":\a . ");
B R a = B R a ();
```

```
S a = . a L ();
```

```
( a!= )
```

```
S . . ( );
```

```
a = . a L ();
```

```
. ();
```

```
E G a a a A -F W
```

```
I S a R a = I S a R a (S . );
```

```
B a b = B R a ();
```

```
b a a = ; S a;
```

```
F W = F W (":\a . ");
```

```
P W = P W ();
```

```
S . . ("D ");
```

```
( a)
```

```
a = b . a L ();
```

```
a . a (" ")
```

```
a = a ;
```

```
. ( a);
```

```
b. ();
```

```
. ();
```

```
F W = F W (":\a . ");
```

```
P W = P W ();
```

```
("A ");
```

```
("A ");
```

```
("L ");
```

```
. ();
```

Ca Ra A F
C a a a Ra A F a a a
E a a a a a a a a a a a
C a a a "F F Ra A F ' ', a
C a a a Ra A F ' ', a
C a a a Ra A F , a a
a . a a a

```
Ca Ra A F - a  
F Na = F (":\\A .  
Ra A F b = Ra A F ( Na , " ");  
=0; S ;  
( < b . ()  
= b . a L ();  
( == )  
b a ;
```

```
S . ( );  
++;  
Ca Ra A F - a a  
I S Ra = I S Ra (S . );  
B a b = E Ra ( );  
F a = F (":\\A 2C a a . ");  
Ra A F b = Ra A F ( Na , " ");  
=0; S ;  
b a a = ; S a;  
( a )  
a = b . a L ();  
a . a (" ")  
a = a ;  
b . B ( a + "\ " );  
b . ();
```

Serialização de objetos - interface Seializable Class ObjectOutputStream e ObjectInputStream

Ja a a a a b a a a . N
a b a a a . S a ab .

```

U b a a a a S a ab , a a a
a a a Ob () Ob O S a S a
a a a a Ob () Ob I S a .

```

Exemplo gravação e leitura de objetos serializados

```

a A a a .S a ab
a S ;
b A (S )
. = ;
b S N ()
;
//
A a = A ("Ma ");
A b = A ("A ");
F a = F ("\\G a aOb . ");
F O S a O = F O S a (a );
Ob O S a b O = Ob O S a (O );
b O . Ob (a);
b O . Ob (b);

F I S a I = F I S a (a );
Ob I S a b I = Ob I S a (I );
A =(A ) b I . a Ob ();
S . . ( . N ());
A =(A ) b I . a Ob ();
S . . ( . N ());

```

Classes do Pacote swing

JComboBox

```

I C () - R a a a a JC b B .
a I (S ) - A a a S a a JC b B
b I A ( ) -

```

JTextField

S T S) - D a
S T - L a
F () - C a F " Ob
a F L (Ob a Ca I a a I a C)
A a a E

JCheckBox

b a S () - D C B a N
S (b a) - Ma a D a a C B .
a A L (Ob a Ca I a a I a C)
A a a E

JRadioButton

b a S () - D JRa B S a N
S (b a) - Ma a JRa B .
a A L (Ob a Ca I a a I a C)
A a a E

ButtonGroup

a (JC) - a a a b

JButton

a A L (Ob a Ca I a a I a C)
A a a E

JTextArea

S T (S) - D a a a a
L a , T () - R a a a a T A a a S . J a a
a (S) - A a a S a a T A a

JOptionPane

a JO Pa . M a Da (, S , S , JO Pa .ERROR_MESSAGE)