

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Subprogramas

Guilherme, Gustavo, Sean e Vinícius

Universidade Estadual de Londrina

October 8, 2013

Sumário

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Subprogramas
Genéricos

Questões de

- 1 Fundamentos
- 2 Ambiente de referenciamento locais
- 3 Aninhamento de subprogramas
- 4 Subprogramas Como Parâmetro
- 5 Chamar Subprogramas Indiretamente
- 6 Sobrecarga de Subprogramas
- 7 Subprogramas Genéricos
- 8 Questões de projetos referente a funções
- 9 Sobrecarga de operadores definidos pelo usuário
- 10 Closure
- 11 Co-rotinas

Introdução

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

- Conceito muito importante nas linguagens de programação.
- Máquina análitica de Babbage
- Reuso, economia de tempo e abstração.
- Métodos nas linguagens orientadas a objeto também são subprogramas

Características comuns

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
loais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

- Cada subprograma tem um único ponto de entrada.
- Há apenas um subprograma em execução em um dado momento.
- O controle sempre retorna para a estrutura que chamou quando a execução do subprograma termina.

Cabeçalho de subprograma

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Fornece um nome para o subprograma e especifica uma lista de parâmetros.

- Ruby e Python

```
def funcao (parametros)
```

- C-based

```
void funcao (parametros)
```

Corpo dos subprogramas

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Os corpos dos subprogramas definem as computações.

- *C-based*. Delimitadas por chaves { }
- *Python*. Identação
- *Ruby*. Palavra-chave **end**

Peculiaridade de Ruby

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

```
1 class Exemplo
2   def invocar_subprograma
3     puts self.method(:invocar_subprograma).owner #
        Exemplo
4     subprograma
5   end
6 end
7
8 def subprograma
9   puts self.method(:subprograma).owner # Object
10 end
11
12 exemplo = Exemplo.new
13 exemplo.invocar_subprograma
```

Parâmetros

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

- Parâmetros Reais (Argumentos)
- Parâmetros Formais (Parâmetros)

Exemplos de parâmetros por palavras-chave

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

- Parâmetros Formais (Parâmetros)

```
funcao(20, 10)
```

- Parâmetros Reais (Argumentos)

```
void funcao(int param1, int param2)
```

Exemplos de parâmetros por palavras-chave

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
loais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Ada, Fortran 95+ e Python

```
1 def subprograma(param1, param2, param3):  
2     print param1 # 30  
3     print param2 # 20  
4     print param3 # 10  
5  
6 subprograma(param3 = 10, param2 = 20, param1 = 30)
```

Desvantagem: Cliente precisa saber o nome dos parâmetros

Exemplos de parâmetros com valores padrão

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
loais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Python, Ruby, C++, Fortran 95+, Ada e PHP

```
1 def subprograma(param1, param2 = 20, param3 = 30):  
2     print param1 # 10  
3     print param2 # 20  
4     print param3 # 30  
5  
6 subprograma(10)
```

Passagem de hash e listas como parâmetros

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

```
1 class Conta
2   def transfere(argumentos, *valores)
3     destino = argumentos[:para]
4     data = argumentos[:em]
5
6   end
7 end
8
9 conta = Conta.new
10 conta.transfere({:para => :escola, :em => Time.now},
    [20.0, 30.0])
```

Ambiente de referenciamento locais

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re- ferenciamento locais

Aninhamento de subprogramas

Subprogramas Como Parâmetro

Chamar Subprogramas Indiretamente

Sobrecarga de Subprogramas

Suprogramas Genéricos

Questões de

Variáveis locais estáticas

São vinculadas ao armazenamento antes da execução do programa e continuam até seu término.

- + Endereçamento direto na memória.
- + Não causam sobrecarga na alocação e desalocação.
- - Não se comportam bem em programas recursivos.
- - Representam um estado global.

Ambiente de referenciamento locais

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de referenciamento locais

Aninhamento de subprogramas

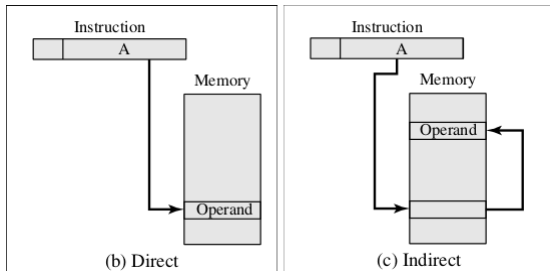
Subprogramas Como Parâmetro

Chamar Subprogramas Indiretamente

Sobrecarga de Subprogramas

Subprogramas Genéricos

Questões de



Ambiente de referenciamento locais

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

```
1 int sum (int arr[], int n)
2 {
3     static int result = 0;
4     if (n == 0)
5         return result ;
6     else {
7         result += arr[n - 1];
8         sum(arr, n - 1);
9     }
10 }
11
12 int main(void) {
13     int array[5] = {1,2,3,4,5};
14     printf("%d\n", sum(array, 3)); // 6
15     printf("%d\n", sum(array, 3)); // 12
16     printf("%d\n", sum(array, 3)); // 18
17     return 0;
18 }
```

Ambiente de referenciamento locais

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-ferenciamento locais

Aninhamento de subprogramas

Subprogramas Como Parâmetro

Chamar Subprogramas Indiretamente

Sobrecarga de Subprogramas

Suprogramas Genéricos

Questões de

Variáveis locais dinâmicas na pilha

Variáveis dinâmicas na pilha, são vinculadas ao armazenamento quando o subprograma inicia sua execução e desvinculadas do armazenamento quando ele se encerra.

- + Maior flexibilidade (programas recursivos).
- - Sobrecarga na alocação e desalocação.
- - Endereçamento indireto.

Ambiente de referenciamento locais

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Exemplos

- ALGOL 60 e suas linguagens descendentes, possuem variáveis locais dinâmicas na pilha.
- Funções em C possuem variáveis são dinâmicas na pilha a menos que sejam especificamente declaradas como static.
- Subprogramas Pascal e Ada e métodos em C++, Java, C# têm somente variáveis dinâmicas na pilha.

Aninhamento de subprogramas

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento de subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Linguagens como ALGOL 68, Pascal e Ada, JavaScript, Python e Lua permitem aninhamento de subprogramas. Linguagens descententes de C não permitem aninhamento.

```
function hipotenusa(a, b) {  
    function quadrado(x) {  
        return x * x;  
    }  
    return Math.sqrt(quadrado(a) + quadrado(b));  
}
```

Subprogramas Como Parâmetro

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas Como Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Subprogramas
Genéricos

Questões de

- Ideia simples, mas gera complicações.
- *Type checking*.
- referencing environment.

Referencing Environment

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

- Linguagens que permitem subprogramas aninhados.
- Shallow Binding
- Deep Binding
- Ad Hoc Binding

Exemplo

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

```
1 function sub1() {  
2   var x;  
3   function sub2() {  
4     alert(x);  
5   };  
6   function sub3() {  
7     var x;  
8     x = 3;  
9     sub4(sub2);  
10  };  
11  function sub4(subx) {  
12    var x;  
13    x = 4;  
14    subx();  
15  };  
16  x = 1;  
17  sub3();  
18 };
```

Shallow Binding

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas Como Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

O ambiente é o local onde o subprograma é chamado.

Shallow Binding

Subprogramas

Subprogramas Como Parâmetro

Questões de

```
1 function sub1() {
2     var x;
3     function sub2() {
4         alert(x);
5     };
6     function sub3() {
7         var x;
8         x = 3;
9         sub4(sub2);
10    };
11    function sub4(subx) { ←
12        var x;
13        x = 4;
14        subx();
15    };
16    x = 1;
17    sub3();
18 }
```

Deep Binding

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas Como Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

O ambiente refere-se onde o subprograma foi definido.

Deep Binding

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

```
1 function sub1() { ←
2   var x;
3   function sub2() {
4     alert(x);
5   };
6   function sub3() {
7     var x;
8     x = 3;
9     sub4(sub2);
10  };
11  function sub4(subx) {
12    var x;
13    x = 4;
14    subx();
15  };
16  x = 1;
17  sub3();
18 };
```

Ad Hoc Binding

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
loais

Aninhamento
de
subprogramas

Subprogramas Como Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

O ambiente condiz com o local que o subprograma foi passado por parâmetro. Nunca implementado.

Ad Hoc Binding

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

```
1 function sub1() {  
2   var x;  
3   function sub2() {  
4     alert(x);  
5   };  
6   function sub3() { ←  
7     var x;  
8     x = 3;  
9     sub4(sub2);  
10  };  
11  function sub4(subx) {  
12    var x;  
13    x = 4;  
14    subx();  
15  };  
16  x = 1;  
17  sub3();  
18 };
```

Chamar Subprogramas Indiretamente

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar Subprogramas Indiretamente

Sobrecarga de
Subprogramas

Subprogramas
Genéricos

Questões de

- Subprograma conhecido em tempo de execução.
- GUI e callback.
- C/C++ ponteiro para função.
- C# Delegate.

C/C++ - Ponteiro Para Função

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

```
1 //declaracao da funcao
2 int sum(int a, int b)
3 {
4     return a + b;
5 }
6
7 //ponteiro para a funcao
8 int (*sum_pointer)(int, int);
9 sum_pointer = &sum;
10
11 //chamar a funcao
12 (*sum_pointer)(1,2);
```

C# - Delegate

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

```
1 //declarar um delegate
2 public delegate int SumDelegate(int a, int b);
3 ...
4 //instanciar um delegate (funcao sum tem a mesma
   assinatura)
5 SumDelegate sumDelegate = new SumDelegate(sum);
6 //executar
7 sumDelegate(2,3);
```

Sobrecarga de Subprogramas

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de Subprogramas

Subprogramas
Genéricos

Questões de

- Subprogramas (diferentes) com o mesmo nome.
- Parâmetros diferentes.
- Subprogramas relacionados.
- Exemplo: Sobrecarga de construtor.
- Ada, Java, C++, C# e F#.

Subprogramas Genéricos

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Subprogramas Genéricos

Questões de

- Reuso de software é algo importante.
- Subprogramas com tipos genéricos.
- Exemplo: Ordenação independente de tipo.
- C++ - Templates
- Java e C# - Generics

C++ - Templates

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

```
1 //declarar funcao template
2 template <class myType>
3 myType GetMax (myType a , myType b) {
4 return (a>b?a:b);
5 }
6 ...
7 //exemplo de chamada para inteiro
8 GetMax<int> (1,2);
9 ...
10 //exemplo de chamada para float
11 GetMax<float> (1,2);
```

Java - Generics

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

```
1 //declarar um metodo generico.  
2 public static <T> T dolt(T[] list) {  
3 ...  
4 }  
5 ...  
6 //chamar o metodo para String  
7 dolt<String>(myList);  
8  
9 ...  
10 //chamar o metodo para Integer  
11 dolt<Integer>(myList);  
12  
13 ...  
14 //isso causaria um erro (tipo primitivo)  
15 dolt<int>(myList);
```

Questões de projetos referente a funções

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Considerações

- Efeitos colaterais
- Tipos de valores retornados
- Quantidade de valores retornados

Efeitos colaterais

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Exemplo de aliasing

```
int x = 3;  
...  
... // se int* y = &x;  
*y = 9;
```

Tipos de valores retornados

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Alguns exemplos

- C permite qualquer tipo ser retornado por suas funções exceto vetores e funções.
- C++ permite tipos definidos pelo usuário ou classes serem retornados.
- Java e C#, qualquer tipo ou classe podem ser retornados por seus métodos.

Quantidade de valores retornados

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Linguagem Lua

Lua permite o retorno de múltiplos valores de suas funções.
Por exemplo, a chamada da função:

```
a, b, c = fun()
```

Recebe três valores de retorno da função `func()`:

```
return 3, sum, index
```

Sobrecarga de operadores

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
loais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Linguagens como Ada, Python, Ruby e C++ suportam sobrecarga de operadores.

```
1 CVector CVector::operator+ (CVector param) {  
2     CVector temp;  
3     temp.x = x + param.x;  
4     temp.y = y + param.y;  
5     return (temp);  
6 }  
7  
8 int main () {  
9     CVector a (3,1);  
10    CVector b (1,2);  
11    CVector c;  
12    c = a + b;  
13    cout << c.x << ", " << c.y;  
14    return 0;  
15 }
```

Closure

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Closure é uma variável local em uma função que é mantida viva (não é desalocada) após o retorno dessa função.

Linguagens como C# e JavaScript possuem closure.

```
1 function foo(x) {  
2     var tmp = 3;  
3     return function (y) {  
4         alert(x + y + (++tmp));  
5     }  
6 }  
7  
8 var bar = foo(2);  
9 bar(10);
```


Co-rotinas

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

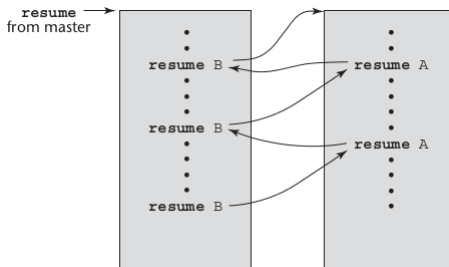
Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

Co-rotinas são um tipo especial de subprogramas. A linguagem Lua é uma das linguagens que possui co-rotinas. Geralmente, corrotinas são criadas pela aplicação por uma unidade chamada de unidade mestre.



Co-rotinas

Subprogramas

Guilherme,
Gustavo, Sean
e Vinícius

Fundamentos

Ambiente de
re-
ferenciamento
locais

Aninhamento
de
subprogramas

Subprogramas
Como
Parâmetro

Chamar
Subprogramas
Indiretamente

Sobrecarga de
Subprogramas

Suprogramas
Genéricos

Questões de

