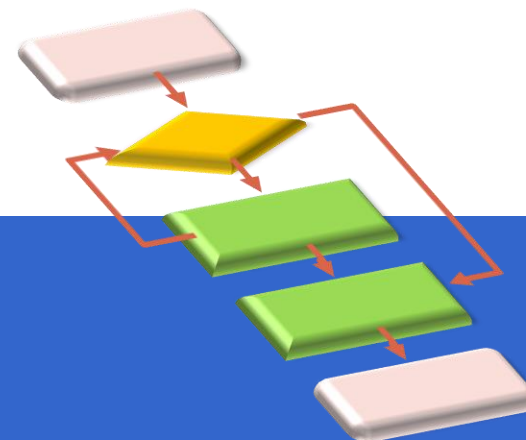




INSTITUTO DE COMPUTAÇÃO



Algoritmos II

MODULARIZAÇÃO

Refinamento de algoritmos

Prof.^a Vanessa de Oliveira Campos

Em Problemas Reais

- Algoritmos nas aulas atuais
~ 20 linhas de código
- Jogo para iPhone
~ 13 mil linhas de código
- SGBD PostgreSQL
~ 570 mil linhas de código
- Kernel do Ubuntu
~ 13 milhões de linhas de código



Em Problemas Reais

Tudo isso é feito em um único algoritmo?

Claro que não.



INSTITUTO DE
COMPUTAÇÃO

Refinamento

Sistema com 10 funcionalidades diferentes



**INSTITUTO DE
COMPUTAÇÃO**

Refinamento

Sistema com 10 funcionalidades diferentes

F01

F02

F03

F04

F05

...

F10



INSTITUTO DE
COMPUTAÇÃO

Refinamento

Sistema com 10 funcionalidades diferentes

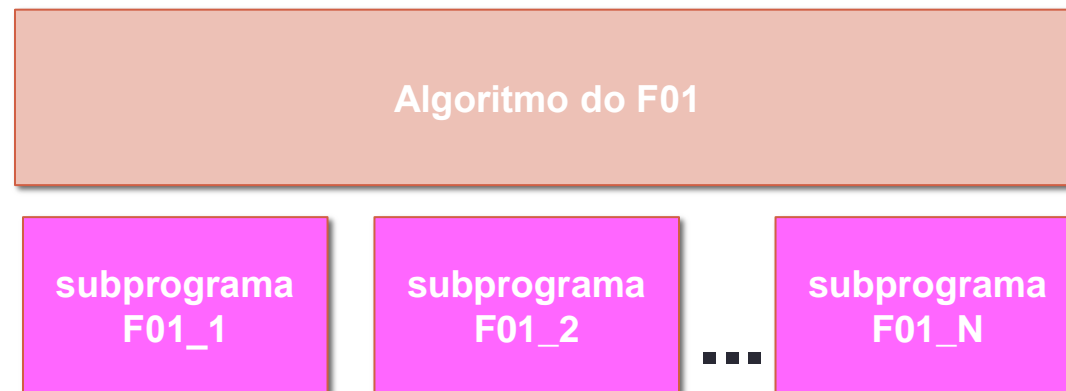


Algoritmo do F01



Refinamento

Sistema com 10 funcionalidades diferentes



Refinamento

Sistema com 10 funcionalidades diferentes



```
Algoritmo "F01_1"  
Var  
    a: inteiro  
Inicio  
    a <- 1  
    ...  
Fimalgoritmo
```



Refinamento

Ao depararmos com um problema complexo, devemos visualizá-lo como um conjunto de problemas mais simples.



Algoritmo GerênciaBiblioteca – passos gerais 1

{GERENCIA AS OPERAÇÕES DE UMA BIBLIOTECA}

início

repetir

mostrar menu

ler opção desejada

executar ação correspondente à opção

até opção de saída

fimalgoritmo

Algoritmo GerênciaBiblioteca – passos gerais 2

{GERENCIA AS OPERAÇÕES DE UMA BIBLIOTECA}

Variável de entrada:

opção (inteiro)

início

repetir

mostra menu

ler(opção)

caso opção seja

1: *cadastrar livros novos*

2: *registrar empréstimo*

3: *registrar devolução*

até opção = 9

fimalgoritmo

Algoritmo GerênciaBiblioteca

{GERENCIA AS OPERAÇÕES DE UMA BIBLIOTECA}

Variável de entrada:

opção (inteiro)

início

repetir

executar MostraMenu

ler(opção)

caso opção seja

1: executar CadastrarLivrosNovos

2: executar RegistrarEmpréstimo

3: executar RegistrarDevolução

até opção = 9

fimalgoritmo

Algoritmo GerênciaBiblioteca

{GERENCIA AS OPERAÇÕES DE UMA BIBLIOTECA}

Variável de entrada:

opção (inteiro)

início

repetir

executar **MostraMenu**

ler(opção)

caso opção seja

1: executar **CadastrarLivrosNovos**

2: executar **RegistrarEmpréstimo**

3: executar **RegistrarDevolução**

até opção = 9

fimalgoritmo

Subprogramas

MonstraMenu

CadastrarLivrosNovos

RegistrarEmpréstimo

RegistrarDevolução

Algoritmo GerênciaBiblioteca

{GERENCIA AS OPERAÇÕES DE UMA BIBLIOTECA}

Variável de entrada:

opção (inteiro)

início

repetir

MonstraMenu

ler(opção)

caso opção seja

1: CadastrarLivrosNovos

2: RegistrarEmpréstimo

3: RegistrarDevolução

até opção = 9

fimalgoritmo

Exemplo

15

Algoritmo "exemplo"

Var

a, b: inteiro

Procedimento leiaValores

Inicio

leia(a, b)

Fimprocedimento

Procedimento escrevaValores

Inicio

escreva("Valor de a: ", a)

escreva("Valor de b: ", b)

Fimprocedimento

Inicio

leiaValores

escrevaValores

Fimalgoritmo

Vantagens

- Dividir e estruturar um algoritmo em partes logicamente coerentes;
- Maior legibilidade;
- Facilidade em testar os trechos separadamente;
- O programador poderá criar sua própria biblioteca de funções;
- Programação mais eficiente;
- Diminuição do código fonte;
- Reaproveitamento de código.



Consideração Final

- Refinamentos sucessivos.



**INSTITUTO DE
COMPUTAÇÃO**

