

Programação Estruturada





$\Rightarrow 50\% \rightarrow \underbrace{NP1/NP2} \mid \text{Sub Ex.}$

Se voltar antes de JUNTO

Introdução

- Forma de programação de computadores
- Desenvolvida por Michael A. Jackson no seu livro "Principles of Program Design" de 1975
- Foi a forma dominante na criação de software anterior à programação orientada por objetos.

Introdução

- Um programa é composto por blocos elementares de código
- Se interligam através de três mecanismos básicos 
 - Sequência 
 - Seleção  *Decisão*
 - Iteração  *Loop*
- Cada uma destas construções tem um ponto de início (o topo do bloco) e um ponto de término (o fim do bloco) de execução.

LPA, LPE

Estruturas Básicas

- Estruturas de **sequência**: Onde uma tarefa é executada após a outra, linearmente.
- Estruturas de **decisão**: Onde, a partir de um teste lógico, determinado trecho de código é executado, ou não.
- Estruturas de **iteração**: Onde, a partir de um teste lógico, determinado trecho de código é repetido por um número finito de vezes.

Programação Estruturada

- Na prática → **Programação Modular**
 - orienta os programadores para a criação de estruturas simples em seus programas, usando **subrotinas** e **funções**
 - pode descrever uma tarefa tão complexa quanto necessário, sendo que esta tarefa pode ser posteriormente descrita em termos de outro(s) fluxograma(s) ou pseudo-código(s)

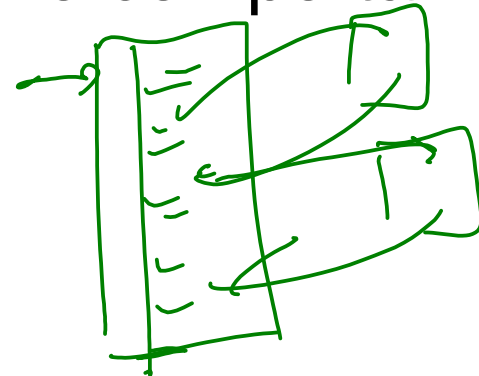
```
multi(a,b)  
-----  
vari  
multi(2,4)  
multi(3.5)
```

Vantagens/Desvantagens

- Vantagens:
 - Provê um melhor controle sobre o fluxo de execução do código
 - É fácil de se entender, sendo amplamente usada em cursos introdutórios de programação
 - Para a resolução de problemas relativamente mais simples e diretos a programação estruturada é muito eficiente

Vantagens/Desvantagens

- Desvantagens:
 - Ainda se foca em como a tarefa deve ser feita e não em o que deve ser feito.
 - Tende a gerar códigos confusos, onde tratamento dos dados são misturados com o comportamento do programas



Linguagens

- Pascal, C
- Cobol, PHP e Perl ainda utilizam o paradigma estruturado (muito embora possuam suporte para a orientação à objetos)

Pascal

- Foi criada em 1970 pelo suíço Niklaus Wirth
- Facilita a criação de procedimentos com baixo acoplamento e alta coesão
- Um programa em Pascal é composto de constantes e variáveis globais, procedimentos e funções re-entrantes e um programa principal

Pascal

- Exemplo:

```
Program Cinco_Numeros;  
Uses  
  WINCRT;  
Var  
  Numbers1, Numbers2, Numbers3,  
  Numbers4, Numbers5, Sum : Integer;  
Begin  
  ClrScr;  
  WriteLn('Entre com os cinco números: ');  
  ReadLn(Numbers1, Numbers2, Numbers3, Numbers4, Numbers5);  
  Sum := Numbers1 + Numbers2 + Numbers3 + Numbers4 + Numbers5;  
  WriteLn('A soma dos números é: ',Sum);  
  WriteLn('Pressione ENTER ...');  
  Readkey;  
End.
```

Linguagem C

- Criada em 1972, por Dennis Ritchie, no AT&T Bell Labs, para desenvolver o sistema operacional Unix
 - Fornece acesso de baixo nível à memória e baixos requerimentos do hardware
 - É uma linguagem de alto nível, para maior reaproveitamento do código
 - C foi útil para muitas aplicações que foram codificadas originalmente em Assembly

Linguagem C

- C tem como ponto forte, a sua eficiência
 - É a linguagem de programação preferida para o desenvolvimento de sistemas e softwares de base
 - Também é usada para desenvolver programas de computador]
- C tem como ponto fraco, a falta de proteção que dá ao programador
 - Praticamente tudo que se expressa em um programa em C, pode ser executado
 - Os resultados são muitas vezes inesperados, e os erros, difíceis de encontrar

Linguagem C

- Exemplo:

```
#include <stdio.h>

int fatorial (int numero) {
    int resultado = 1;
    int contador;
    for (contador = 1; contador <= numero; contador++)
        resultado *= contador;
    return resultado;
}

int main() {
    int n;

    printf("Calcular o fatorial de: ");
    scanf("%d",&n);
    printf("%d\n",fatorial(n));

    return 0;
}
```