POO - PROGRAMAÇÃO ORIENTADA A OBJETOS

INTRODUÇÃO - Paradigmas de Linguagem

Rodrigo R Silva

Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense
Campus Bagé

Projeto de Linguagens

Desafios no projeto de linguagens de programação:

- Instruções claras;
- Padrões de codificação de algoritmos.

Fatores de Influência no Projeto

Muitos fatores influenciam na decisão do projeto de linguagens de programação:

- Fatores relacionados a Ciência da Computação: legibilidade, capacidade de escrita, confiabilidade e custo;
- Fatores relacionados a arquitetura de computador: estrutura de organização de computadores;
- Fatores relacionados a metodologias empregadas na escrita de programas.

TADS
Tecnologia em Análise e Desenvolvimento de Sistemas

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

Arquitetura de Computador

Influência do modelo de Von Neumann no projeto das linguagens.

- Programas e dados são armazenados na mesma memória (RAM);
- Unidade Central de Processamento (UCP) é responsável pela execução das instruções do programa;
- As instruções do programa são executadas sequencialmente.

Arquitetura de Computador

Na organização do modelo de Von Neumann a memória ficou separada da UCP.

- As instruções do programa devem ser transmitidas da memória para UCP para que sejam executadas;
- Assim como os dados manipulados pelas instruções também devem ser transmitidos da memória para UCP;
- Os resultados devem ser enviados de volta a memória.

Arquitetura de Computador

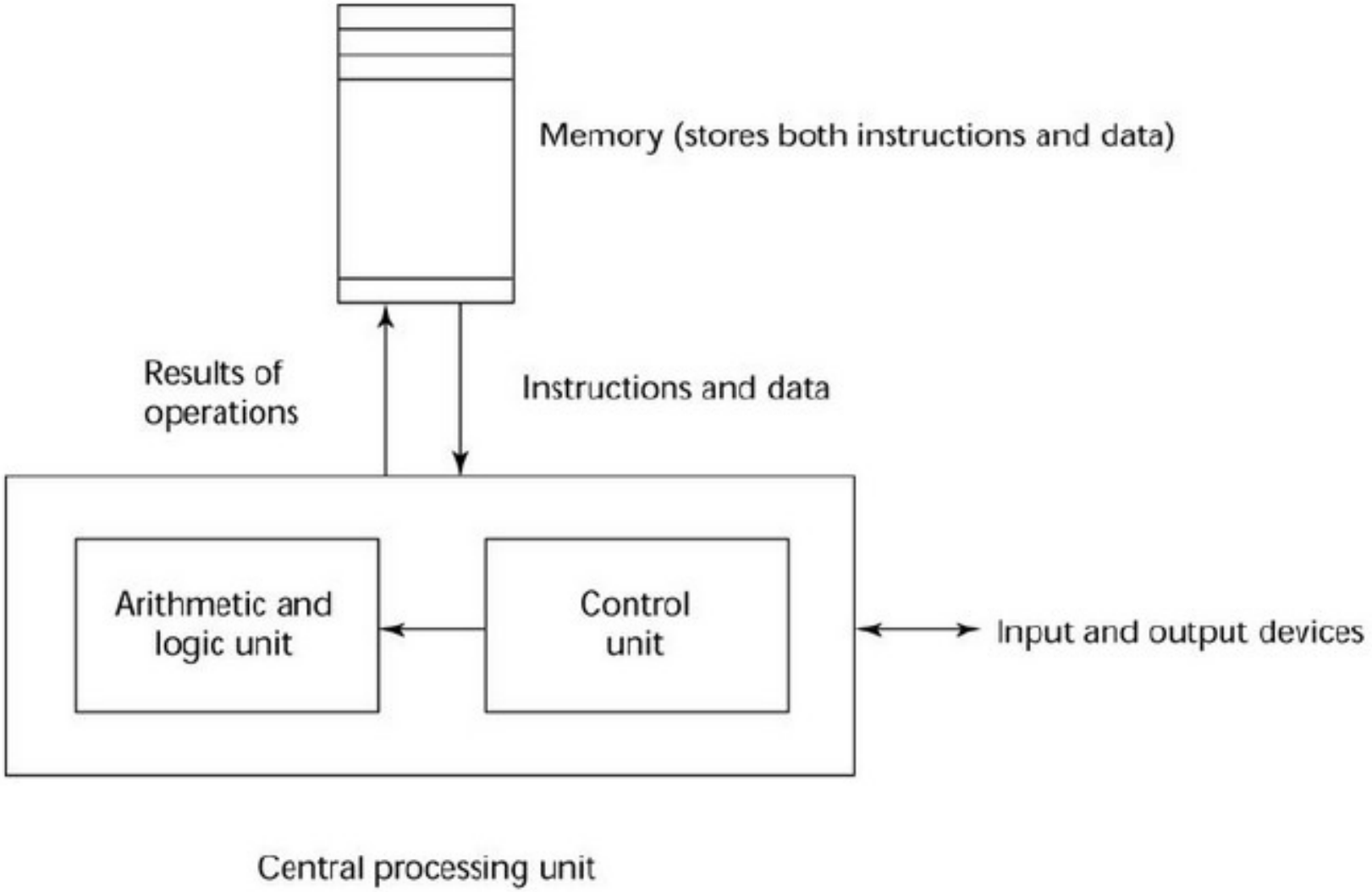
Para que ocorra a comunicação entre os componentes da UCP e memória (RAM) é utilizado um canal denominado barramento.

- Barramento de sinais de controle;
- Barramento de endereços;
- Barramento de dados.

TADS
Tecnologia em Análise e Desenvolvimento de Sistemas

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

Arquitetura de Computador



Arquitetura de Computador

O modelo de Von Neumann motivou diversos aspectos na concepção de linguagens de programação que são utilizados até hoje:

- O uso de variáveis para referenciar endereços de memória onde estão dados manipulados pelos programas;
- Fácil implementação de estruturas de controle como tomada de decisão e repetição;
- Com base nas características e implicações do modelo de Von Neumann, surgiram as primeiras metodologias no projeto de linguagens.

TADS
Tecnologia em Análise e Desenvolvimento de Sistemas

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

Paradigmas de Programação

Um paradigma de programação é um estilo de codificação de programas em uma determinada linguagem de programação.

- Linguagens fortemente orientadas a paradigmas;
- Linguagens multiparadigma.

Paradigma de Programação Imperativa

As primeiras linguagens foram fortemente inspiradas no modelo de Von Neumann.

- Foco no processo (sequenciação de instruções de um programa);
- Estruturação de sentenças imperativas que determinam o fluxo de movimentação dos dados entre memória e UCP;
- Paradigma mais antigo e mais próximo a forma como o computador funciona.

TADS
Tecnologia em Análise e Desenvolvimento de Sistemas

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

Linguagem Basic

```
1
2       10 INPUT A,B
3       20 LET I=A
4       30 IF I>B THEN 80
5       40 IF MOD(I,2)>0 THEN 60
6       50 PRINT I
7       60 LET I = I + 1
8       70 GOTO 30
9       80 END
```

Programação Estruturada

A programação estruturada surge como um estilo de programação que visa contornar o problema de uso de GOTO.

- Uso excessivo de GOTO gera o problema do código espaguete;
- Surgimento de estruturas de controle para tomada de decisão e repetição.

TADS
Tecnologia em Análise e Desenvolvimento de Sistemas

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

Linguagem C

```
1
2   int main(){
3       int a, b;
4       scanf("%d %d", &a, &b);
5       for(int i = 0; i <= a; i++){
6           if(i % 2 == 0)
7           printf("%d", i);
8       }
9   }
```

Linguagem Pascal

```
1
2       var
3           i, a, b: integer;
4       begin
5           read(a, b);
6           for i := 0 to b do
7               begin
8               if( a mod b = 0) then
9                   writeln(i);
10              end;
11          end.
```

Programação Estruturada

A programação estruturada também segue o paradigma imperativo.

- GOTO é mantido para fins de compatibilidade, mas não é utilizado;
- Programação estruturada também é conhecida como procedural.

Programa Monolítico

No início da atividade de programação a construção dos programas era na forma de uma estrutura monolítica.

- Todo código de uma aplicação em um único bloco ou módulo;
- Inviável para grandes aplicações de software.

Programa Monolítico

Com o aumento da complexidade e linhas de código de programas monolíticos, surgem alguns problemas.

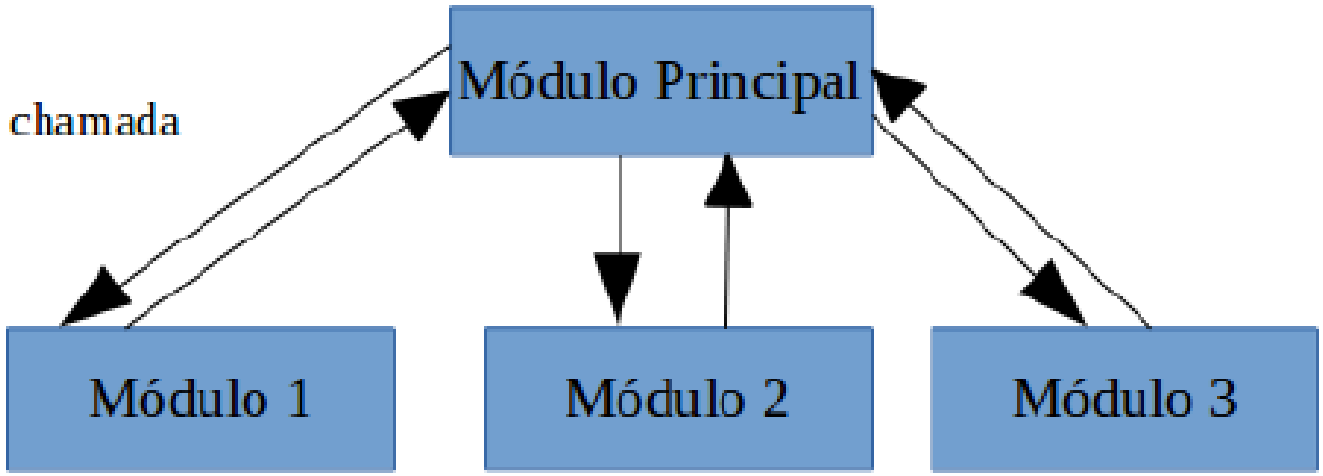
- Variáveis de memória;
- Estruturas de controle;
- Fácil indução de erros;
- Reutilização de código.

Modularização de Código

Pouca eficiência no processo de desenvolvimento de software motivou a ideia de segmentação do código monolítico em porções menores.

- O software passa a ser organizado em um conjunto de módulos;
- Princípio da Engenharia de Software denominado Divisão e Conquista.
- Implementado em programação na forma de módulos (sub-programas).

A Modularização de Código



Modularização de Código

Em programação imperativa a modularização é implementada na forma de funções ou procedimentos.

- Funções são tipos de módulos de código que sempre retornam valor;
- Procedimentos são módulos que não retornam valores;
- Nem sempre as linguagens possuem construções diferentes para funções e procedimentos.

Pascal e Modularização de Código

```
1
2   program mediaNotas;
3   procedure mostraParImpar(media: real);
4   begin
5       if( media mod 2 = 0) then
6           writeln("A média é par");
7       else
8           writeln("A média é ímpar");
9       end;
10
11  function calculaMedia(a, b: real): real;
12  begin
13      calcularMedia := (a + b) / 2;
14  end;
15
16  begin
17      mostraParImpar(calculaMedia(8.5, 6.5));
18  end.
```

C e Modularização de Código

```c

void mostraParImpar(float media){
    if( media % 2 == 0)
        printf("A média é par.");
    else
        printf("A média é ímpar.");
}

float calculaMedia(float a, float b){
    return (a + b) / 2;
}

int main(){
    mostraParImpar(calculaMedia(8.5, 6.5));
}
```

TADS Tecnologia em Análise e Desenvolvimento de Sistemas

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

Programação Estruturada

As linguagens imperativas também apresentam suporte a manipulação de tipos de dados estruturados.

- Os tipos estruturados são utilizados para criação de novos tipos de dados os quais combinam em sua estrutura um conjunto de tipos primitivos;
- Os tipos estruturados formam estruturas heterogêneas, diferente de estruturas como vetores e matrizes que são homogêneas;
- Tipos estruturados em linguagens também são denominados registros.

Tipos estruturados em Pascal

```
1
2       program mediaNotas;
3
4       type
5           agenda = record
6               string nome[30];
7               integer idade;
8               long celular;
9           end;
10      var
11          contatos: agenda;
12
13      begin
14          contatos.nome = "Fulano";
15          contatos.idade = 18;
16          contatos.celular = 99887711;
17      end.
```

Tipos estruturados em C

```
1
2       struct agenda{
3           char nome[30];
4           int idade;
5           long celular;
6       }
7
8       int main(){
9           struct agenda contatos;
10          contatos.nome = "Fulano";
11          contatos.idade = 18;
12          contatos.celular = 99887711;
13  }
```

Limitações da Programação Estruturada

Apesar das possibilidades oferecidas pelas estruturas de dados, a programação estruturada continua focada no processo.

- Os dados da aplicação são desvinculados dos algoritmos que os manipulam;
- Podemos criar um módulo somente com a estrutura dos dados e outro com os algoritmos que irão manipular tais estruturas;
- Ou seja, não há um vínculo forte entre os dados e as operações.

Limitações da Programação Estruturada

A desvinculação dos dados com as operações que os manipulam dificulta o reuso de software.

- Difícil reutilizar código em novas aplicações;
- Não permite a extensão das funcionalidades de um módulo.

Programação Orientada a Objetos

O princípio da programação orientada a objetos é o foco nos dados manipulados pelas aplicações.

- Dados e operações são descritos dentro do mesmo módulo de programa;
- Aplicações são constituídas de módulos de software que trocam informações entre si.

Programação Orientada a Objetos

A construção de uma aplicação inicia pela especificação do modelo de dados. Diferente do paradigma imperativo onde o foco é no processo.

- Foco na definição de estruturas de dados e operações;
- Abstração é o mecanismo utilizado para definição dos módulos da aplicação.

Programação Orientada a Objetos

Abstração é um mecanismo cognitivo, ou seja, uma habilidade mental do programador que é utilizada na identificação das características fundamentais de um módulo de software.

- Os módulos de software são denominados classes na programação orientada a objetos;
- Abstração é utilizada na definição da estrutura de uma classe.

Programação Orientada a Objetos

Uma classe é uma estrutura modular que define um conjunto de dados e operações em uma única unidade de software.

- Classes são modelos utilizados em programas orientados a objetos que descrevem características e comportamentos de objetos;
- A partir do modelo de objeto descrito em uma classe as aplicações instanciam objetos que serão manipulados durante o ciclo de vida de um programa.

OBRIGADO!

TADS
Tecnologia em Análise e Desenvolvimento de Sistemas

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA