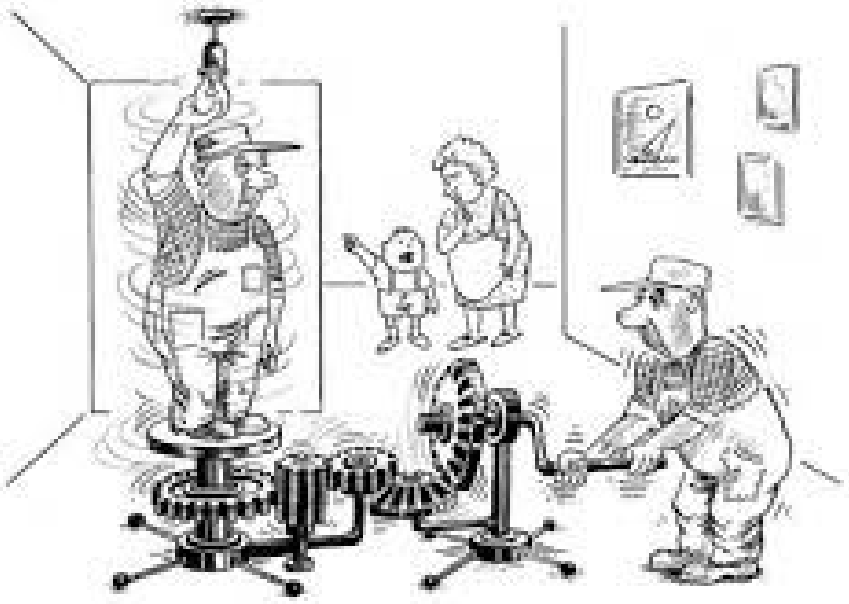
PUC Minas

Programação Orientada por Objetos

Aula Revisão de ATP

Curso: Sistemas de Informação
Disciplina: POO

Professor: Fábio Leandro Rodrigues Cordeiro



PUC Minas

(Re)visão e Introdução

Sumário:

→ **Algoritmos**

→ **Programas**

→ **Conceitos importantes**

→ **Tipos de estruturas**

→ **Vetores (arranjos) e Arranjos Multidimensionais**

→ **Funções e Procedimentos**

→ **Parâmetros**

PUC Minas

Algoritmos

Um algoritmo corresponde a uma descrição de um padrão de comportamento, expresso em termos de um conjunto finito de ações (Dijkstra, 1971).

- Um algoritmo é uma sequência de passos finitos para resolução de um problema.

- Algoritmos fazem parte do dia a dia:
- Instruções para uso de medicamentos
- Manual para montar um aparelho
- Receita de culinária

PUC Minas

Algoritmos

→ Um algoritmo é dito correto se, para cada entrada, ele gera uma saída esperada.

→ Um algoritmo pode ser especificado em uma linguagem comum, como um programa de computador ou projeto de hardware.

PUC Minas

Programas

→ Programar é basicamente estruturar dados e construir algoritmos.

→ Programas são formulações concretas de algoritmos abstratos, baseados em representações e estruturas específicas de dados.

→ Programas representam uma classe especial de algoritmos capazes de serem seguidos por computadores.

PUC Minas

Programas

→ Um computador só é capaz de seguir programas em linguagem de máquina.

→ É necessário construir linguagens mais adequadas, que facilitem a tarefa de programar um computador.

→ Uma linguagem de programação é uma técnica de notação para programar.

PUC Minas

Conceitos Importantes

Constantes:

→ Representam valores inalteráveis durante toda a execução.

Variável:

→ Espaço da memória reservado para armazenar um certo tipo de dado e tendo um nome para referenciar o seu conteúdo.

→ Este valor pode variar durante a execução

PUC Minas

Conceitos Importantes

Tipo de dados:

→ Define um conjunto de valores que uma variável pode armazenar

→ Define o conjunto de operações que pode ser executado com essa variável

(ex.: int, double, string).

→ Forma como a variável é armazenada e interpretada.

PUC Minas

Conceitos Importantes

Tipo de variáveis:

Informa a quantidade de memória, em bytes, que a variável ocupará, e a forma como o valor deverá ser armazenado e interpretado.

int: números inteiros de 32 bits (4 bytes).

long: números inteiros (intervalo maior) de 64 bits (8 bytes).

double: ponto flutuante (maior precisão) 64 bits (8 bytes).

string: sequência de caracteres de 16 bits

(2 bytes) por caractere.

bool: booleano de 8 bits (1 bytes):

PUC Minas

Conceitos Importantes

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Revisao
{
    class Program
    {
        static void Main(string[] args)
        {
            int idade = 25;
            double salario = 250.00, Salario = 100;
            int maximo = 65000;
            int GUESS = maximo;
            int orcamento_de_contabilidade_de_2023 = 1000;
        }
    }
}
```

PUC Minas

Conceitos Importantes

Utilize nomes significativos para as variáveis

Exemplo:

int idade;

float salario;

Características dos nomes das variáveis

Sempre começa com uma letra, ou um *underline*

Com exceção do primeiro caractere, podem existir números

Palavras-chaves não podem representar nomes de variáveis

Caractere minúsculo ≠ caractere maiúsculo

PUC Minas

Tipos de Estruturas

Estrutura Condicional

Comando if

Forma Geral:

```
If (condição_for_verdadeira)
                    comando;
```

O comando if executa um teste utilizando um operador relacional do C#.

Condição_for_verdadeira

Pode ter apenas um operador relacional ou pode ter *n* operadores relacionais ligadas por operadores lógicos

PUC Minas

Tipos de Estruturas

Estruturas de repetição:

Repetir comandos um número específico de vezes.

```
for (inicialização; teste; incremento)
      comando(s);
```

Repetir comandos até que uma condição conhecida ocorra.

```
while (condição_for_verdadeira)
       comando(s);
```

Executar comandos pelo menos uma vez, possivelmente repetindo-os no futuro

```
do{
      comando(s);
} while (condição_for_verdadeira);
```

PUC Minas

Vetores (Arranjos)

→ Um *array* (vetor) é uma sequência não ordenada de elementos do mesmo tipo.

→ Arranjos podem ser:

- Unidimensional (Vetor)
- Bidimensional (Matriz)

→ Informalmente:

“arranjo é uma série de variáveis **do mesmo tipo** referenciadas por um único nome

cada variável (elemento do arranjo) é diferenciada por um índice

PUC Minas

Vetores (Arranjos)

→ Criando uma instância de *array*:

→ Ao declarar uma variável de *array* seu tamanho não é declarado, isso ocorre no momento de instanciar um *array* utilizando ***new***.

→ Exemplo:

```
int [] nota = new int [4];
```

→ Vetor de inteiros

nota [0], nota [1], nota [2], nota[3]

PUC Minas

Vetores (Arranjos)

Quando o vetor é declarado, o compilador alocará memória suficiente para conter todos os elementos do vetor.

```
int [ ] peso = new int [10];
double [ ] nota = new double [41];
string [ ] nome = new string [80];
```

PUC Minas

Arranjos Multidimensionais

→ É possível definir um vetor em que cada posição temos um outro vetor (matriz).

```
int [ , ] materia = new int[4,4]
```

→ Interpretação:
temos 4 matérias, cada uma com 40 alunos

```
int [ , ] materia = new int[linha, coluna]
```

PUC Minas

Funções e Procedimentos

→ Conjunto de instruções para cumprir uma tarefa particular e agrupadas numa unidade com um nome para referenciá-la.

→ Dividir a tarefa original em pequenas tarefas que simplificam e organizam o programa como um todo.

→ Reduzir o tamanho e a complexidade do programa.

→ Um programa pode conter uma ou mais funções.

PUC Minas

Funções e Procedimentos

→ Chamar uma função é o meio pela qual solicitamos que o programa desvie o controle e passe à função, execute suas instruções e depois volte para à instrução seguinte que a chamou.

→ Uma função pode conter zero ou mais argumentos.

→ Funções retornam algum valor e procedimentos não.

→ Duas funções podem ter o mesmo nome mas se diferem em relação aos parâmetros.

PUC Minas

Parâmetros

→ Programas podem passar dados para funções, chamados de parâmetros.

→ Quando declarar uma função, deve ser declarado também os tipos dos parâmetros da funções, estes parâmetros estão entre parênteses.

→ Uma função pode receber mais de um parâmetro, onde estes devem ser separados por vírgula.

→ Funções podem não receber parâmetros algum

PUC Minas

Referências (Bibliografia Básica)

LOUREIRO, Henrique. **C# 5.0 com Visual Studio® 2012**: curso completo. Lisboa: FCA, c2013. xxii, 585 p. ISBN 9789727227525 (Disponível no Acervo).

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de. **Algoritmos: lógica para desenvolvimento de programação de computadores**. 26. ed. rev. São Paulo: Érica, 2012. 328 p. ISBN 9788536502212 (Disponível no Acervo).

SZWARCFITER, Jayme Luiz; MARKENZON, Lilian. **Estruturas de dados e seus algoritmos**. 3. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos, c2010. xvi, 302 p. ISBN 9788521617501 (Disponível no Acervo).