

# Fundamentos de Programação 1

**Slides N. 2 – D / Prof. SIMÃO**

**Slides elaborados pelo Prof. Robson Linhares**

**<http://www.dainf.ct.utfpr.edu.br/~robson/>**

---

# Fundamentos de Programação I

---

Conceitos Básicos de Computação

---

# Tópicos

- Dados
  - Informação
  - Algoritmos e programas
  - Estruturas de dados
  - Da teoria à prática...
-

---

# Dados

- Definição segundo [1]: sequência de símbolos quantificados ou quantificáveis.
  - Inclui-se na definição: texto, fotos, figuras, sons gravados, animação, etc.
  - Dados são elementos *sintáticos* – podem ser descritos através de representações formais e estruturais
    - Exemplo: diário de classe
-

---

# Dados & computadores

- Em essência, computadores são máquinas que processam dados
  - Questões importantes:
    - Como são os dados que os computadores processam? De que forma eles são representados?
    - Programas de computador são dados?
    - Por que computadores não processam “informação”?
-

---

# Informação

- Definição segundo [1]: uma abstração informal (sem representação formal) que representa algo significativo para alguém.
    - Exemplo: “Curitiba é uma cidade bonita” é um exemplo de informação que possui significado para alguém que conheça os conceitos de “Curitiba”, “cidade” e “bonita”.
  - Informação é representada por meio de *dados*
    - Exemplo: situação de aprovação/reprovação dos alunos obtida a partir do diário de classe
    - Outro exemplo: um texto escrito em chinês é puro *dado* para quem não entende chinês, porém pode se constituir em *informação* para quem conhece chinês
  - Informação contém *semântica* – “significado”
-

---

# Algoritmos e programas

- Segundo [2], *um algoritmo é um conjunto finito de regras que fornece uma sequência de operações para resolver um problema específico.*
  - Algoritmo é composto por dados – “texto” do algoritmo
  - Algoritmo contém informação – sequência lógica de operações que o compõe
  - Algoritmos são a base para os *programas de computador*
    - Modelo de Von Neumann: programas são conjuntos de instruções executadas sequencialmente
-

---

# Algoritmos e programas

- Características de um algoritmo
    - **Finitude:** Um algoritmo deve sempre terminar após um número finito de passos.
    - **Definição:** Cada passo de um algoritmo deve ser precisamente definido. As ações devem ser definidas rigorosamente e sem ambiguidades.
    - **Entradas:** Um algoritmo deve ter zero ou mais entradas, isto é dados que são lhe são fornecidos para que o algoritmo seja executado.
    - **Saídas:** Um algoritmo deve ter uma ou mais saídas, isto é quantidades que tem uma relação específica com as entradas.
    - **Efetividade:** Um algoritmo deve ser efetivo. Isto significa que todas as operações devem ser suficientemente básicas de modo que possam ser em princípio executadas com precisão em um tempo finito por um humano usando papel e lápis.
-



---

# Algoritmos e programas

- Exemplos de algoritmos:
  - 1) Saque no banco 24 horas (do ponto de vista de quem saca...)

```
Verificar previamente a quantia a ser sacada
Ir até um banco 24 horas
Colocar o cartão
Digitar a senha
Fornecer a quantia desejada
Se o saldo for maior ou igual à quantia desejada,
    retirar o dinheiro
Retirar o cartão
Sair do banco 24 horas
Disponibilizar o dinheiro
```

---

---

# Algoritmos e programas

## □ 2) Saque no banco 24 horas (do ponto de vista do banco)

```
Enquanto cartão não é inserido
  Aguardar inserção do cartão
Mostrar tela de senha
Enquanto senha não é inserida
  Aguardar inserção da senha
Se a senha confere com a senha cadastrada
  Enquanto valor não é inserido
    Aguardar inserção do valor
  Se o valor for igual ou inferior ao saldo
    Disponibilizar o dinheiro
  Senão
    Mostrar mensagem de saldo insuficiente
  Senão
    Mostrar mensagem de senha incorreta
Voltar à tela inicial
```

---

---

# Algoritmos e programas

- ❑ 3) Converter um intervalo em segundos no equivalente em horas, minutos e segundos

Início

```
inteiro total, horas, minutos, segundos;  
escreva "Digite um intervalo em segundos: ";  
leia total;  
horas <- total/3600;  
minutos <- (total - horas*3600)/60;  
segundos <- total % 60;  
escreva "O intervalo corresponde a ", horas, " horas, ",  
minutos, " minutos e ", segundos, " segundos.";
```

Fim

---

---

# Estruturas de dados

- Programas de computador (algoritmos) recebem entradas e geram saídas - *dados*
  - Programas de computador utilizam *dados* durante o seu processamento
  - Dados em um programa podem ser representados por *estruturas de dados*
    - Estruturas de dados visam a organizar os dados de tal maneira que o seu armazenamento ou processamento seja facilitado em algum aspecto
    - Exemplo: diário de classe “eletrônico”
-

---

# Da teoria à prática...

- Algoritmos e estruturas de dados compõem a *lógica da programação* – são a semântica (significado) de como um programa funciona e de como seus dados são organizados
    - Algoritmos, de maneira geral, não podem ser executados diretamente por um computador
  - Programas de computador “reais” são compostos por *instruções de máquina* – possuem semântica para um tipo de máquina (processador) específico
    - Instruções de máquina, em geral, são complexas e de difícil manipulação pelo ser humano
  - Para gerar programas de computador a partir de algoritmos e estruturas de dados – utiliza-se *ferramentas de programação* apropriadas
-

# Da teoria à prática...

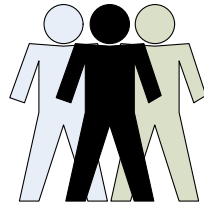
- Ferramentas de programação essenciais:
  - *Linguagem de programação* – conjunto de regras de sintaxe para expressão (“escrita”) de um algoritmo
    - Analogia: “se vou escrever uma receita para um *chef* francês, é importante que seja escrita em francês...”
  - *Ambiente de programação* – todo o “suporte” para que um algoritmo escrito em uma linguagem de programação seja transformado em um programa de computador
    - Exemplos: Borland C++, Delphi, C++ Builder, Microsoft Visual Studio .NET, Eclipse, NetBeans, KDevelop, etc.

---

# Da teoria à prática...

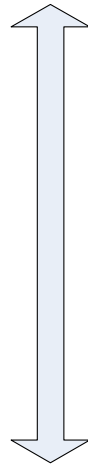
- Ferramentas de programação essenciais:
    - *Ambiente de execução* – todo o “suporte” para que um programa de computador seja *executado* e os seus resultados sejam verificados
      - Sistema operacional, *hardware*, etc.
  
  - Ferramentas de apoio à programação:
    - *Ferramentas de auxílio à engenharia de software (CASE)*
      - Análise, especificação, projeto, implementação e teste de software (programas)
      - Utiliza alguma linguagem de projeto, por exemplo, UML (*Unified Modeling Language*)
      - Exemplos: Enterprise Architect, Rational Rose, etc.
-

# Da teoria à prática...

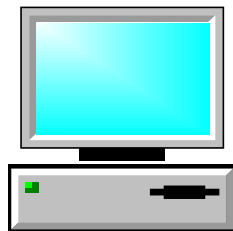


Usuário

- “Objetivo” do programa de computador
  - **ALGORITMO**
- Utiliza linguagem algorítmica (linguagem de programação) – para o computador, somente um texto comum...



- FERRAMENTAS DE PROGRAMAÇÃO
- Lógica de programação
  - Linguagem de programação
  - Ambientes de desenvolvimento



Computador

- Executa **SOFTWARE** – conjunto de instruções de máquina
- Código binário – complexo para o ser humano...



---

# Da teoria à prática...

- ❑ 1) Converter um intervalo em segundos no equivalente em horas, minutos e segundos (em linguagem Java)

```
public class Ex001
{
    public static void main(String[] args) {
        int total = 0, horas, minutos, segundos;
        System.out.println("Digite um intervalo em segundos: ");
        BufferedReader entrada = new BufferedReader (new InputStreamReader(System.in));

        try {
            String linha = entrada.readLine();
            total = Integer.parseInt(linha);
        } catch (IOException e) {
            System.exit(0);
        }
        catch (NumberFormatException e)
        {
            System.out.println("Intervalo inválido");
            System.exit(0);
        }
        horas = total/3600;
        minutos = (total - horas*3600)/60;
        segundos = total % 60;
        System.out.println("O intervalo corresponde a " + horas + " horas, " + minutos +
            " minutos, " + segundos + " segundos.");
    }
}
```

---

---

# Da teoria à prática...

- ❑ 2) Converter um intervalo em segundos no equivalente em horas, minutos e segundos (em linguagem C)

```
int main(int argc, char* argv[])
{
    int total, horas, minutos, segundos;
    printf ("Digite um intervalo em segundos: ");
    scanf("%d", &total);
    horas = total/3600;
    minutos = (total - horas*3600)/60;
    segundos = total % 60;
    printf ("\nO intervalo corresponde a %d horas, %d minutos e %d segundos",
    horas, minutos, segundos);
    return 0;
}
```

---

---

## Referências online

- **[1]** Valdemar Setzer: Dado, informação, conhecimento e competência  
<http://www.ime.usp.br/~vwsetzer/dado-info.html>
  - **[2]** O que são algoritmos? – Adriano Cruz e Jonas Knopman  
<http://equipe.nce.ufrj.br/adriano/algoritmos/apostila/algoritmos.htm>
-