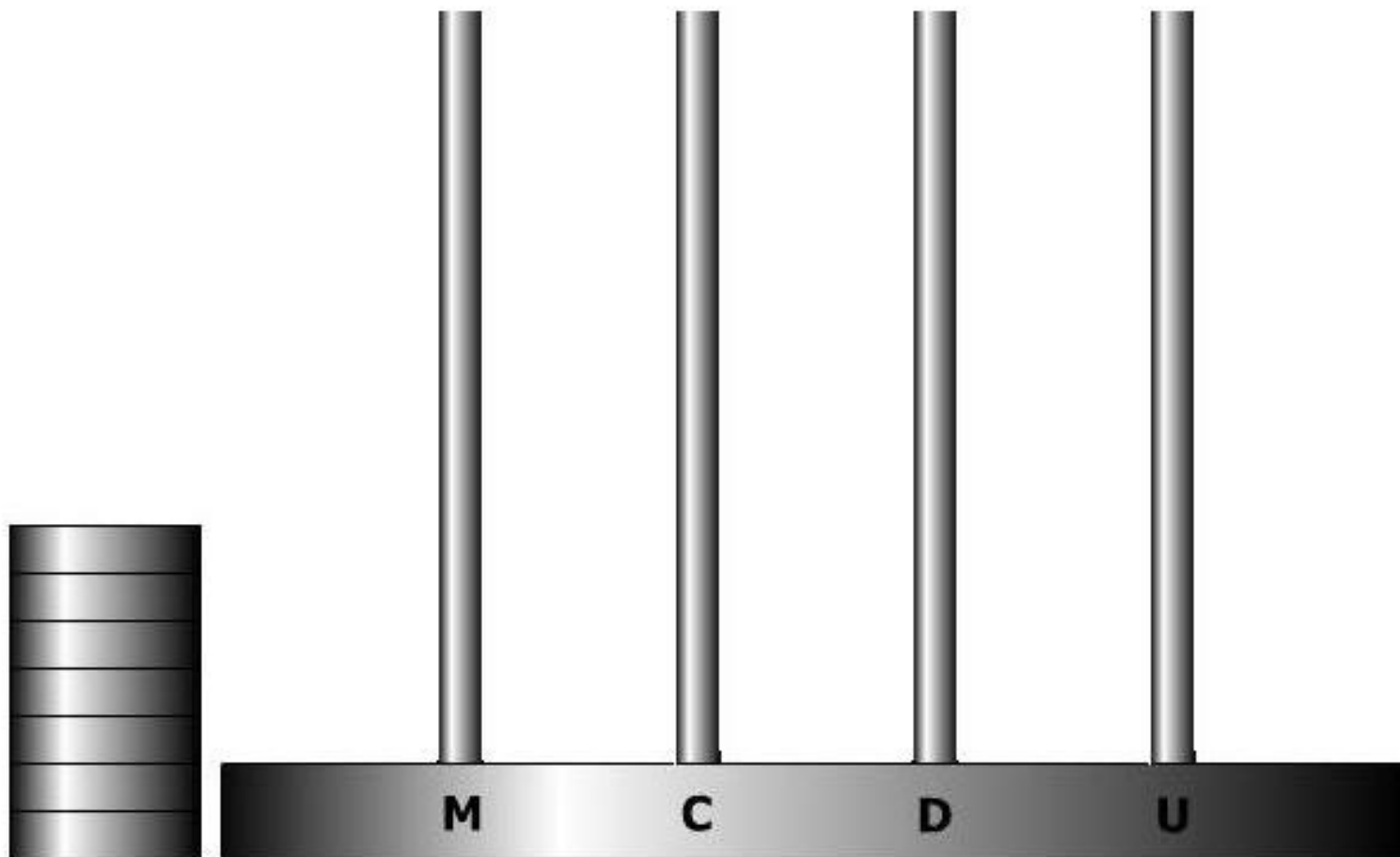
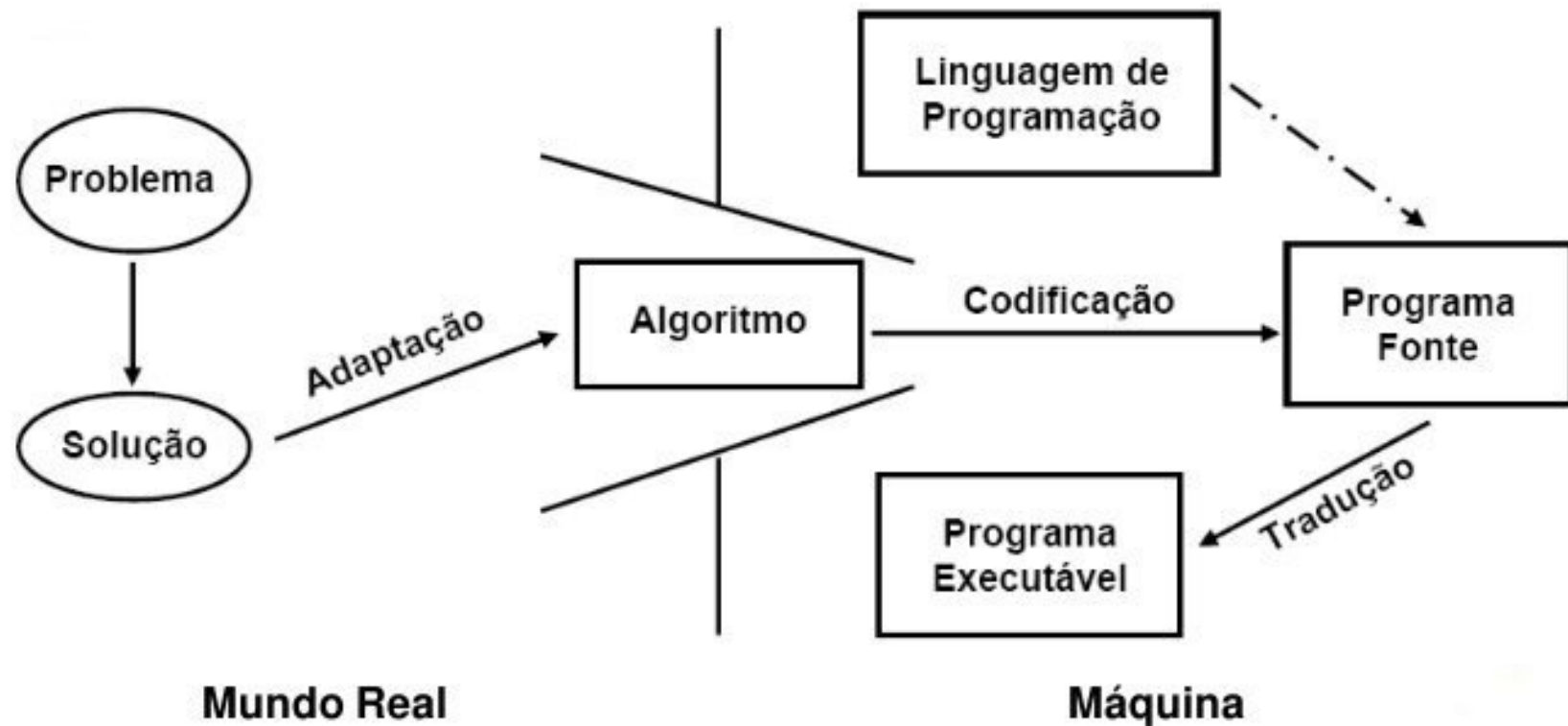


Programação com Algoritmos



- O que é lógica?
 - “A arte ou técnica de pensar corretamente.”

- O que é algoritmo?
 - “É uma sequência de passos que visam atingir um objetivo bem definido.”



- Exemplos de algoritmos

ALGORITMO: TROCAR UMA LÂMPADA

PASSO 1: Pegar a lâmpada nova

PASSO 2: Pegar a escada

PASSO 3: Posicionar a escada embaixo da lâmpada queimada

PASSO 4: Subir na escada com a lâmpada nova

PASSO 5: Retirar a lâmpada queimada

PASSO 6: Colocar a lâmpada nova

PASSO 7: Descer da escada

PASSO 8: Ligar o interruptor

PASSO 9: Guardar a escada

PASSO 10: Jogar a lâmpada velha no lixo

ALGORITMO: SACAR DINHEIRO

PASSO 1: Ir até o caixa eletrônico

PASSO 2: Colocar o cartão

PASSO 3: Digitar a senha

PASSO 4: Solicitar o saldo

PASSO 5: Se o saldo for maior ou igual à quantia desejada, sacar a quantia desejada; caso contrário sacar o valor do saldo

PASSO 6: Retirar dinheiro e cartão

PASSO 7: Sair do caixa eletrônico

- Métodos de representação de algoritmos

Descrição Narrativa

Adquira uma resistência nova e localize o chuveiro a ser manipulado. Em seguida abra o chuveiro retirando a resistência defeituosa, coloque a resistência nova e feche o chuveiro. Após descarte a resistência defeituosa.

Pseudocódigo

- 1. Pegar (resistência nova);**
- 2. Pegar (chuveiro);**
- 3. Abrir (chuveiro);**
- 4. Retirar (resistência defeituosa);**
- 5. Colocar (resistência nova);**
- 6. Fechar (chuveiro);**
- 7. Largar (resistência defeituosa).**

- Alguns obstáculos de um programador
 - Definir o escopo
 - Até onde deveremos ir; definir a abrangência
 - Determinar a complexidade
 - Determinar a quantidade de situações diferentes que um problema pode apresentar
 - Garantir a legibilidade
 - A clareza com que sua lógica está exposta.

- Um método para construção de algoritmos
 - Ler atentamente o enunciado
 - Retirar do enunciado a relação das entradas de dados
 - Retirar do enunciado a relação das saídas de dados
 - Determinar o que deve ser feito para transformar as entradas determinadas nas saídas especificadas
 - Construir o algoritmo
 - Executar o algoritmo

- Na construção do algoritmo
 - Utilizar o método Cartesiano quando a complexidade não estiver totalmente absorvida, conhecida
 - Aplicar o Planejamento Reverso, ou seja, a partir das saídas, procurar desagregar, desmontando a informação, a fim de atingir os dados de entrada

- O método Cartesiano
 - Dividir o problema em suas partes principais
 - Analisar a divisão obtida para garantir coerência
 - Se alguma parte não for bem compreendida, aplicar a ela nova divisão
 - Analisar o objetivo para garantir entendimento e coerência

A primeira linguagem

- Tipos primitivos
 - Uma classificação dos tipos de informações que podemos manipular
 - Grupo básico
 - Inteiro (int)
 - Ex: 1 -234
 - Real (double)
 - Ex: -2.34 12.34
 - Texto (String)
 - Ex: "a" "teste"
 - Character (char)
 - Ex: 'a' 't','e','s','t','e'
 - Lógico (boolean)
 - Ex: true false

Classificação da Informação

- Constantes
 - Informação que não deve sofrer alteração durante a execução do programa
- Variáveis
 - Informação que tem a possibilidade de ser alterada durante a execução do programa

Formação de Identificadores

- Palavras para nomear constantes e variáveis, etc
 - Regras para os nomes
 - Começar por letra ou sublinhado “_”
 - Podem ser seguidos por letras, sublinhado “_” e números
 - Não podem ser iguais a uma “palavra-chave”

Declaração de variáveis

- Para que possamos processar as informações, estas devem ser armazenadas em áreas pré-definidas para que possamos:
 - Ler seu conteúdo
 - Gravar novo conteúdo
 - Apresentar seu conteúdo como resultado de algum processo

Declarando Variáveis

- Declaramos variáveis seguindo a seguinte sintaxe:

Tipo da variável Nome da variável, Nome da variável ;

- Sendo que a lista de variáveis tem a seguinte sintaxe:

Exemplo:

```
int x ;
```

```
String nome, endereco, email ;
```

A entrada de dados

- A entrada de dados é essencial para o processamento da informação.
- Os comandos para entrada de dados são:

Nome da variável = leTexto(Mensagem) ;

Nome da variável = leInteiro(Mensagem) ;

Nome da variável = leReal(Mensagem) ;

Ex.:

nome = leTexto("Informe seu Nome") ;

idade = leInteiro("Informe sua Idade") ;

A saída de dados

- Da mesma forma que a entrada é importante a saída também.
- Os comandos para saída de dados são:

```
escreva( Lista de variáveis e Textos );
```

```
escrevaL( Lista de variáveis e Textos );
```

- A diferença entre “escreva” e “escrevaL” é que o segundo comando insere uma linha após mostrar o conteúdo.

Ex.:

```
escreva( “Nome: ” , nome );
```

```
escrevaL( “Idade: ” , idade );
```


- A expressão utilizada para atribuir valores (literais) para as variáveis é através do símbolo “ = ”

Ex.:

```
int a = 5 ;
```

```
double valor1 = 3.14 ;
```

```
double valor2 = valor1 ;
```

Operadores aritméticos

- Abaixo temos a tabela de símbolos e suas respectivas operações aritméticas:

<i>Símbolo</i>	<i>Operação</i>
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da Divisão

Operadores aritméticos

- Existe uma precedência para as operações aritméticas, abaixo temos a tabela de precedência:

<i>Hierarquia</i>	<i>Operação</i>
1	Parênteses
2	Função
3	-, + (unários)
4	%, *, /
5	+, -

Operadores aritméticos

- Segue alguns exemplos de construção de expressões aritméticas:

$$3 / 4 + 5 = 5.75$$

$$3 / (4 + 5) = 0.3333333$$

$$3 / 2 * 9 = 13.5$$

$$11 \% 3 * 2 = 4$$

$$11 \% (3 * 2) = 5$$

$$(11 \% 3) * 2 = 4$$

$$3 / 2 + (65 - 40) * (1 / 2) = 14$$

Operadores relacionais

- Utilizamos os operadores relacionais para realizar comparações entre dois valores de mesmo tipo primitivo.
- Tais valores são representados por constantes, variáveis ou expressões aritméticas.

Operadores relacionais

- Abaixo temos a tabela de símbolos e suas respectivas operações relacionais:

<i>Operador</i>	<i>Ação</i>
>	maior que
>=	maior ou igual a
<	menor que
<=	menor ou igual a
==	igual a
!=	diferente de

Operadores lógicos

- Os operadores lógicos possibilitam relacionar condições de três formas diferentes, mais a negação das três:

<i>Símbolo</i>	<i>Função</i>
&&	Conjunção (e)
	Disjunção (ou)
^	Disjunção (ou exclusivo)
!	Negação (não)

Operadores lógicos

- A tabela verdade é o conjunto de todas as possibilidades combinatórias entre os valores de diversas variáveis lógicas, as quais se encontram em apenas duas situações, e um conjunto de operadores lógicos

A	B	A e B
F	F	F
F	V	F
V	F	F
V	V	V

A	B	A ou B
F	F	F
F	V	V
V	F	V
V	V	V

A	B	A xou B
F	F	F
F	V	V
V	F	V
V	V	F

A	não A
F	V
V	F

Linguagem de Programação

- A Linguagem de programação oferece os mecanismos necessários para a representação de um algoritmo em forma textual
- Chamamos este algoritmo em forma textual de programa e este programa é armazenado em um arquivo
- Para que o computador entenda o programa é necessário que este seja transformado em linguagem de máquina
- A transformação de um programa para linguagem de máquina é obtido através de um compilador

Estrutura do Programa

- A estrutura de um programa é semelhante ao apresentado abaixo:

```
public class Exemplo extends Programa {  
    public void inicio() {  
        int numero = leInteiro("Informe um N°");  
        escrevaL("O n° informado foi: ", numero);  
    }  
}
```