**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Capítulo 1**

**Conceitos Básicos**

Acho importante explicar aqui os conceitos básicos da matéria, pois vejo muitos erros a respeito disso, até mais do que a respeito da matéria propriamente dita.

***1.1 JAVA***

A linguagem de programação que usamos em LPTI durante o ano todo foi JAVA, ela tem algumas características, tais como:

-Totalmente Orientada a Objetos(POO);

-Trabalha implicitamente com ponteiros em seus TAD’s.

Explicarei com mais detalhes mais a frente.

***1.2 Básico de programação***

***1.2.1 O que é programação***

Hoje em dia a presença de programas no nosso dia-a-dia é constante, os usamos por diversos motivos. Desde fazer contas nas calculadoras até utilizar contas bancárias.

Contudo programas não são mágicos, para se criar um programa é feito uma linha de raciocínio lógico que processadores consigam entender e processar para então ter um resultado.

Essa linha de raciocínio geralmente é linear e imperativa, ou seja, o programador(pessoa que cria um programa) manda o computador fazer algo, ele faz, depois manda fazer outra coisa, ele faz. Seria como se fosse uma receita de bolo, onde nela está escrito de forma linear e imperativa também como bem podemos ver: coloque o leite, misture. A pessoa que quiser fazer o bolo terá que colocar o leite e depois misturar, nessa ordem.

***1.2.1.1 Linguagem de Programação***

O computador é movido a energia, portanto a única forma de “comunicar” com ele seria “esta entrando energia”, “não está entrando energia”. consolidando assim, o que se chama código binário, que é representado 1 quando está entrando e por 0 quando não está.

Inicialmente então, devemos escrever uma linha de raciocínio que o computador entenda, ou seja, escrever um código binário de tal forma a mandar seu computador fazer algo.

0100010101110101010010110010101010101011000010110101111100101010101101010101101010101010. Pode-se notar que trabalhar com binários não é nada intuitivo, mas era a única forma antigamente.

Muitos programadores perceberam que vários dos seus trechos de código se repetia, então fizeram programas que se, por exemplo, escreve-se if, trocava aquela palavra pelos seus binários correspondentes(aqueles que repetiam) e assim as linguagens de programação foram evoluindo até chegar nos dias de hoje.

***1.2.2 Variáveis***

Em programação, existe variáveis, cujo sua função é guardar dados, para assim, poder usamos esses dados por qualquer motivo que seja, por exemplo, para fazer contas simples**:**

*Você guarda o valor de um número em uma variável e depois guarda o valor de outro número em outra variável e depois soma o valor delas.*

Uma variável em java tem um nome e um valor tipado armazenado. Os nomes das variáveis são definidos pelo programador pois elas são criadas na hora que estamos programando(ou seja, fazendo programas). O valor a toda hora durante a execução do seu programa pode ser mudado, mas não o seu tipo, pois java é fortemente tipado, ou seja, uma variável do tipo inteiro, não pode armazenar valores reais e virce-versa.

Curiosidade: O padrão do java para variáveis é o camelCase([Wiki](http://en.wikipedia.org/wiki/CamelCase))

***1.2.2.1 Como declara-os e atribuir valor***

Para declarar uma variável(ou seja, criar uma variável) em java basta dizer seu tipo e o seu nome. por exemplo:

int variavel;

Para atribuir um valor a variável basta usar o operador reservado =

variavel = 1;

Note que 1 = variavel; vocé estará tentando atribuir o valor da variável com o nome variavel no número 1, o que não faz sentido já que 1 não é uma variável.

O operador = se lé “recebe”.

É possível também declarar uma variável já com um valor, por exemplo:

int variavel = 1;

***1.2.2.2 Tipos Primitivos***

Existem muitos tipos de dados que são usados com frequências como por exemplo tipos inteiros, reais, caracteres, boleanos(verdadeiro ou falso). Em várias linguagens de programação, incluindo java, tem esses tipo pré-definidos pelo cara que criou a linguagem.

O que é relevante ser levantado desses tipos é: eles não usam ponteiros.

Uma tabela dos tipos primitivos de java: [^](http://www.dm.ufscar.br/~waldeck/curso/java/part22.html)

| Tipo | Descrição |
| --- | --- |
| **boolean** | Pode assumir o valor **true** ou o valor **false,** sendo true verdadeiro e false falso |
| **char** | Caractere em notação Unicode de 16 bits. Serve para a armazenagem de dados alfanuméricos. Também pode ser usado como um dado inteiro com valores na faixa entre 0 e 65535. |
| **byte** | Inteiro de 8 bits em notação de complemento de dois. Pode assumir valores entre -27=-128 e 27-1=127. |
| **short** | Inteiro de 16 bits em notação de complemento de dois. Os valores possívels cobrem a faixa de -2-15=-32.768 a 215-1=32.767 |
| **int** | Inteiro de 32 bits em notação de complemento de dois. Pode assumir valores entre -231=2.147.483.648 e 231-1=2.147.483.647. |
| **long** | Inteiro de 64 bits em notação de complemento de dois. Pode assumir valores entre -263 e 263-1. |
| **float** | Representa números em notação de ponto flutuante normalizada em precisão simples de 32 bits em conformidade com a norma IEEE 754-1985. O menor valor positivo represntável por esse tipo é 1.40239846e-46 e o maior é 3.40282347e+38 |
| **double** | Representa números em notação de ponto flutuante normalizada em precisão dupla de 64 bits em conformidade com a norma IEEE 754-1985. O menor valor positivo representável é 4.94065645841246544e-324 e o maior é 1.7976931348623157e+308 |
| **String** | Não é um tipo primitivo, mas é relevante colocar aqui pois é bastante usado e tem funcionalidades bastante similares aos tipos primitivos(tais como não usar ponteiro) |

***1.2.2.3 Tipos Abstratos(TAD)***

**1.2.2.3.1 Conceito de tipos abstratos.**

Além dos tipos primitivos é possível também você criar seu próprio tipo. Que nada mais é do que um conjunto de variáveis em uma variável só. Também conhecido como “classes” ou “TAD’s”(tipo abstrato de dados). Para criar um tipo abstrato de dados em java basta colocarmos class ONomeDaClasse e delimitar o que queremos colocar nela. Veja o exemplo:

*class* ***Pessoa*** *{*

*String olho;*

*int quantidadeDeOlhos;*

*}*

Se você tiver esse treicho de código em seu código fonte você poderá usar o tipo de variável Pessoa. Mais a frente ensinaremos como declarar e usar, é necessário saber algumas coisas antes.

***1.2.2.3.2 C*omo funciona então?**

Como a variável vai guardar vários valores? como vou acessa-los?

Na verdade o que a variável guarda é um endereço de memória, ou seja, aonde está guardado na memória, o valor, das variáveis primitivas dentro da sua variável TAD, tanto, que se você tentar imprimir uma variável TAD o java irá te mostrar o enderço de memória.

Para acessar é simples basta colocar o nome da variavelTAD **.** o que você quiser de dentro daquela variavel, veja o exemplo da pessoa(suponto que guilherme já esteja declarado, que ensinaremos mais pra frente).

guilherme**.**quantidadeDeOlhos = 2;

int x = guilherme**.**quantidadeDeOlhos;

A variável x terá o valor 2.

***1.2.2.3.3 Métodos***

Uma variável TAD também pode ter métodos.

Métodos são treichos de códigos nomeados e delimitados que, quando são chamados, faz o código transcrito no bloco demilitado. Vamos editar nossa classe(tipo) pessoa para que tenhamos uma função que aumente a quantidade de olhos.

*class* ***Pessoa*** *{*

*String olho;*

*int quantidadeDeOlhos;*

*void adicionarOlho() {*

*adicionarOlho = adicionarOlho +1;*

*}*

*}*

*O método que criamos adiciona um olho na variável instanciada, por exemplo, vamos pegar o guilherme que já tinha dois olhos.*

*guilherme.adicionarOlho();*

*Agora guilherme tem três olhos.*

***1.2.2.3.4 Construtor***