Tipos e Variáveis

Tipos de dados

No Java, existem algumas palavras reservadas, para a representação dos tipos de dados básicos, que precisam ser manipulados, para a construção de programas. Estes tipos de dados são conhecidos como tipos primitivos (Primitive Types).

> Os oito tipos primitivos em Java são:
>
> int, byte, short, long, float, double, boolean e char – esses tipos não são considerados objetos e portanto representam valores brutos. Eles são armazenados diretamente na pilha de memória. (Memory stack)

Tabela de Tipos Primitivos e seus valores:

Tipo	Memória	Valor Mínimo	Valor Máximo
byte	1 byte	-128	127
short	2 byte	-32.768	32.767
int	4 bytes	-2.147.483.648	2.147.483.647
long	8 bytes	-9.223.372.036.854.775.808	9.223.372.036.854.775.807

Os tipos primitivos, que podem conter partes fracionárias podem ser representados por dois tipos:

Tipo	Memória	Mínimo	Máximo	Precis
float	4 bytes	-3,4028E + 38	3,4028E + 38	6 – 7 d
double	8 bytes	-1,7976E + 308	1,7976E + 308	15 dígi

Apesar de o tipo **float,** ocupar metade da memória consumida do que um tipo double, ele é menos utilizado. Ele sofre de uma limitação que compromete seu uso em determinadas situações: somente mantém uma precisão decimal entre 6 e 7 dígitos.

Atualmente, com os computadores modernos, se tornou desnecessário utilizar os tipos `short` e `byte` , pois não precisamos nos preocupar tanto assim com o espaço de memória reduzido.

Da mesma forma, dificilmente utilizaremos o tipo `long` , pois não é tão comum trabalharmos com valores tão grandes.

Portanto, para representar números, na grande maioria das vezes, utilizaremos o tipo `int` , para representar números inteiros ou `double` para representar números fracionados.

> ⓘ *O ponto mais relevante, em compreender a definição dos tipos de dados é o momento da definição do tipo para uma variável.* ***Qual tipo de dados eu utilizaria para determinar a idade de uma pessoa ou o salário de um funcionário?***

```
/* FAMILIA  - TIPO PRIMITIVO - WRAPPER CLASS - TAMANHO
 * LÓGICO   -    boolean     -     Boolean    -  1 bit
 * LITERAIS -     char       -    Character   - 1 byte
 *          -      -         -     String     - 1 byte/cada
 * INTEIROS -     byte       -      Byte      - 1 byte
 *          -    short       -      Short     - 2 bytes
 *          -     int        -     Integer    - 4 bytes
 *          -    long        -      Long      - 8 bytes
 * REAIS    -    float       -      Float     - 4 bytes
 *          -    double      -     Double     - 8 bytes
 */
```

Tabela criada pela minha aluna Priscilla Aniboleti - Github - Pripii

Declaração de Variáveis

Variável, é uma identificação de um espaço em memória, utilizado pelo nosso programa. Seguindo as convenções em linguagem de programação, toda variável é composta por: tipo de dados + identificação + valor atribuído.

A estrutura padrão para se declarar uma variável sempre é:

```
<Tipo> <nomeVariavel> <atribuicaoDeValorOpcional>
```

Exemplos abaixo:

```
int idade; //Tipo "int", nome "idade", com nenhum valor atribuído.
int anoFabricacao = 2021; //tipo "int", nome "anoFabricacao", com valor 2021.
double salarioMinimo = 2.500; //tipo "double", nome "salarioMinimo", valor 2.500.
```

Atenção: existe algumas peculiaridades a trabalhar com alguns tipos específicos. Observe no exemplo abaixo:

```
public class TipoDados {
        public static void main(String[] args) {
                byte idade = 123;
                short ano = 2021;
                int cep = 21070333; // se começar com zero, talvez tenha que ser outro tipo
                long cpf = 98765432109L; // se começar com zero, talvez tenha que ser outro tipo
                float pi = 3.14F;
                double salario = 1275.33;
        }
}
```

> ⓘ Observe que o tipo long precisa terminar com L, o tipo float precisa terminar com F e alguns cenários do dia-a-dia, podem estimular uma alteração de tipos de dados convencional.

Muitas das vezes criamos uma variável, definimos um valor correspondente, manipulamos esta variável e temos consciência de seu valor na aplicação. Mas, cuidado!

Veja o cenário abaixo:

```
// TiposEVariaveis.java

short numeroCurto = 1;
int numeroNormal = numeroCurto;
short numeroCurto2 = numeroNormal;
```

> ⓘ Por mais que tenhamos ciência do valor que `numeroNormal` cabe é um short, o **Java** não permite correr o risco.

Variáveis e Constantes

Uma **variável** é uma área de memória, associada a um nome, que pode armazenar valores de um determinado tipo. Um tipo de dado, define um conjunto de valores e um conjunto de operações. **Java** é uma linguagem com rigidez de tipos, diferente de linguagens como JavaScript, onde declarar o tipo da variável não é obrigatório.

No Java, utilizamos identificadores que representam uma referência (ponteiro) a um valor em memória, e esta referência pode ser redirecionada a outro valor, sendo portanto esta a causa do nome "variável", pois o valor pode variar.

Já as **Constantes,** são valores armazenados em memória que não podem ser modificados depois de declarados. Em Java, esses valores são representados pela palavra reservada `final` , seguida do tipo.

Por convenção, **Constantes** são sempre escritas em CAIXA ALTA.

Abaixo, temos um exemplo explicativo sobre uso de variáveis e constantes:

```
public class ExemploVariavel {
        public static void main(String[] args) {
                /*
                 * esta linha é considerada como declaração de variável iniciamos a existência
                 * variavel numero com valor 5 regra: tipo + nome + valor
                 */
                int numero = 5;

                /*
                 * na linha abaixo iremos alterar o valor do varíavel para 10 observe que o tipo
                 * não é mais necessário, pois a variável já foi declarada anteriormente
                 */
                numero = 10;

                System.out.print(numero);

                /*
                 * ao usar a palavra reservada final, você determina que jamais
                 * esta variavel poderá obter outro valor;
                 * logo a linha 25 vai apresentar um erro de compilação
                 * isso é considerado uma CONSTANTE na linguagem Java
                 */
                final double VALOR_DE_PI = 3.14;

                VALOR_DE_PI=3.15; //Esta linha vai apresentar erro de compilação!
        }
```

```
}
```

> Compreendemos que, para declarar uma variável como uma constante, utilizamos a palavra `final`, mas por convenção, esta variável deverá ser escrita toda em caixa alta.

Sintaxe - Previous
Anatomia das classes

Next - Sintaxe
Operadores

Last modified 1mo ago