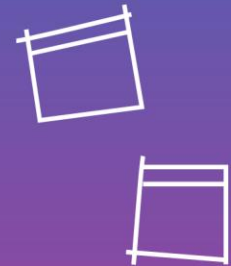




VEM SER
DBC



Variáveis e Condicionais

Variáveis e Constantes

Variável: espaço na memória do computador, esse espaço pode ser **alterável**.

Tipo

Nome

Valor

```
int minhaVariavel = 123456;
```

Variáveis e Constantes

Constante: espaço na memória do computador, esse espaço é **inalterável**.

Tipo

Nome

Valor

```
final int MINHA_CONST = 123456;
```

Se for uma constante utiliza **final**
Se for variável não utiliza nada

Padrão de Nomenclatura

- Nomes de variáveis sempre em camel case começando por minúsculo

- NÃO SE USA UNDERSCORE

```
String minhaVariavel = "teste";
int idadeDoPaciente = 10;
float calculo = 10.0f;
```

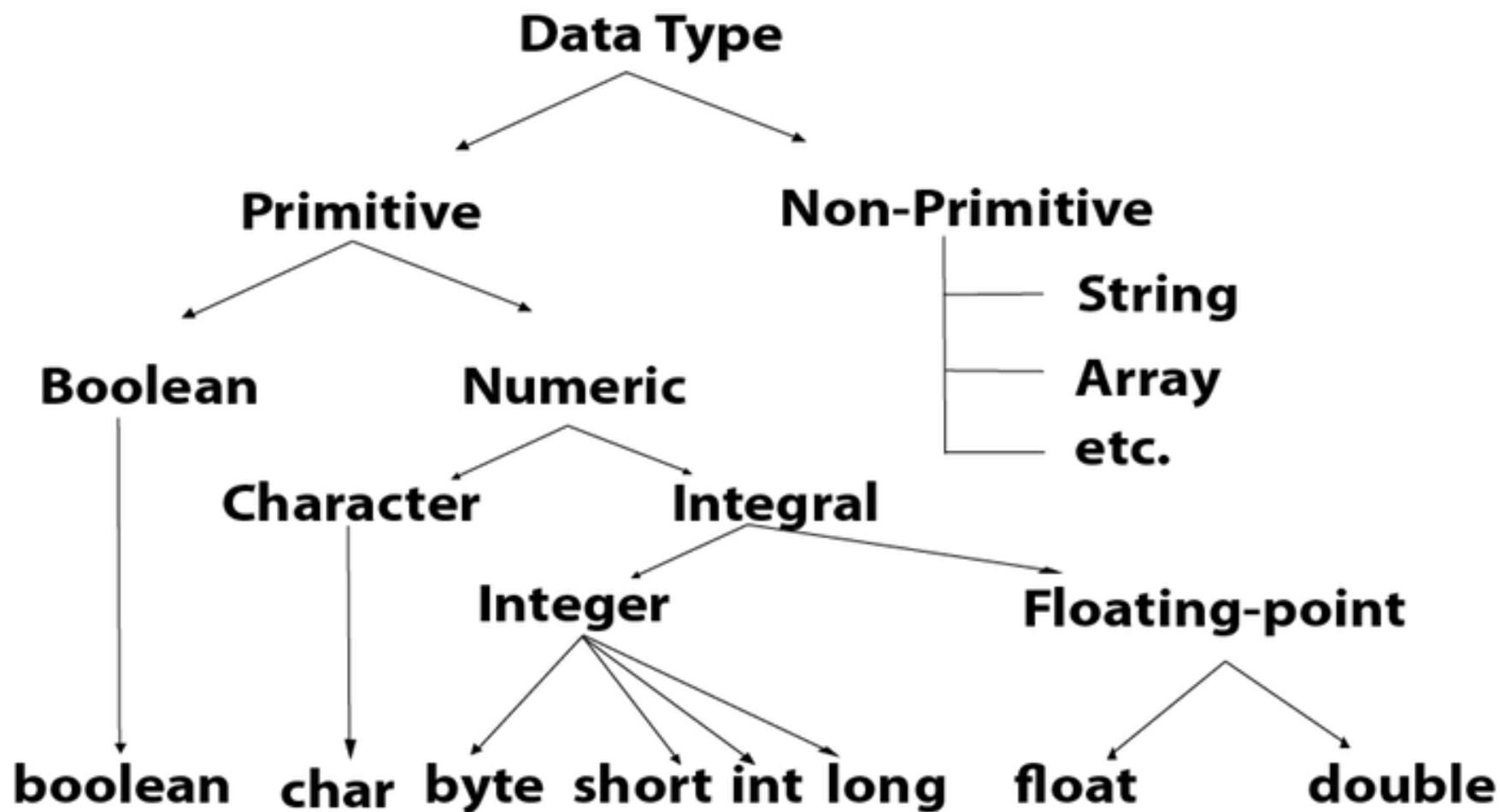
- Nomes de classes sempre em camel case começando por maiúsculo

```
public class Main {
public class MainFeature {
```

- Constantes sempre caixa alta separado por underscore

```
final String MINHA_CONST = "teste";
final int IDADE_DO_PACIENTE = 10;
final float CALCULO = 10.0f;
```

Tipo de Dados



Tipo de Dados Numéricos

Tipo	Mínimo	Máximo
byte	-128	127
short	-32768	32767
int	-2147483648	2147483647
long	-9223372036854775808	9223372036854775807

Tipo de Dados Numéricos Reais

Tipo	Tamanho	Exemplo
float	4 bytes	127.0f
double	8 bytes	32767.0

Tipo de Dados para Caracteres

Tipo	Tamanho	Exemplo
char	1 byte	'a'
String	infinito	"teste"

Tipo de Dados Diversos

Tipo	Tamanho	Exemplo
boolean	1 bit	true ou false
void	nada	

Comentários

- Serve para documentar alguma informação no código
- É ignorado pelo compilador

```
// comentário em uma linha
// comentário em outra linha
/*
    Bloco de comentário
    Nenhuma dessas informações será
    considerada
*/
```

Let's practice;

Exercício #1

- Crie um programa que tenha duas variáveis de tipos diferentes e imprima no console os respectivos valores.

Exercício #2

- Declare uma variável do tipo inteiro e imprima no console os seguintes valores:
 - `valor % 2 == 0`
 - `Math.log(valor)`
 - `Math.sqrt(valor)`
 - `valor / 2`

Operadores Aritméticos

- + : soma
 - - : subtração
 - / : divisão
 - * : multiplicação
 - % : módulo / resto da divisão
-
- ++ : incrementa o valor
 - -- : decrementa
 - += : soma com o valor que está na variável
 - -= : diminui com o valor que está na variável
 - *= : multiplica com o valor que está na variável
 - /= : divide com o valor que está na variável

Let's practice;

Comparadores

`==` : variavel é igual a outraVariavel

`!=` : variavel é diferente da outraVariavel

`<` : variável (< menor) outraVariavel

`>` : variável (> maior) outraVariavel

`>=` : variável (>= maior ou igual) outraVariavel

`<=` : variável (<= menor ou igual) outraVariavel

Let's practice;

Agregadores

! : não ... !ehSolteiro (não ehSolteiro) PS: se a variável estiver como TRUE, ela se torna FALSE e vice-versa.

&& : e ... ehSolteiro e temVidaSocial

|| : ou ... ehSolteiro ou ehCasado

Tabela Verdade

NÃO (!)

var1	!var1 (NÃO valor)
V	F
F	V

E (&&)

var1	var2	var1 && var2 (var1 E var2)
V	V	V
V	F	F
F	V	F
F	F	F

OU (||)

var1	var2	var1 var2 (var1 OU var2)
V	V	V
V	F	V
F	V	V
F	F	F

Let's practice;

Condicionais - IF

```
if (<condição>)  
    <comando>;
```

Condicionais - IF

if (<condição>)

 <comando>;

else

 <comando>;

Condicionais - IF

if (<condição>)

<comando>;

else

<comando>;

Condicionais - IF

~~if (<condição>)~~

~~<comando>;~~

~~else~~

~~<comando>;~~

if (<condição>) {

<comandos>;

}

Condicionais - IF

~~**if** (<condição>)
 <comando>;
else
 <comando>;~~

if (<condição>) {
 <comandos>;
} **else** {
 <comandos>;
}

Condicionais - IF

~~**if** (<condição>)
 <comando>;~~

~~**else**
 <comando>;~~

```
if (<condição>) {
    <comandos>;
} else if (<outraCondição>){
    <comandos>;
} else {
    <comandos>;
}
```

Let's practice;

Condicionais - Switch

```
switch (<seletor>) {
    case <val1>:
        <comandos>;
        break;
    case <val2>:
        <comandos>;
        break;
    case <valN...>:
        <comandos>;
        break;
    default:
        <comandos>;
        break;
}
```

Condicionais - Switch

```
switch (<seletor>) {
  case <val1>:
    <comandos>;
    break;
  case <val2>:
    <comandos>;
    break;
  case <valN...>:
    <comandos>;
    break;
  default:
    <comandos>;
    break;
}
```

SEMPRE COLOQUEM O BREAK...

Let's practice;

Exercícios #3

- Crie um programa que leia um valor inteiro e imprima se é par ou ímpar

Exercícios #4

- Crie uma calculadora simples que tenha as quatro operações básicas.
 - O programa deve pedir 3 valores do usuário, sendo eles:
 - Operação
 - 1 – soma
 - 2 – subtração
 - 3 – divisão
 - 4 – multiplicação
 - Valor1
 - Valor2
 - O programa deve utilizar o valor1 e o valor2 com a operação selecionada.

Homework

- Criar pasta na raiz do seu repo chamada “modulo 1”
- Criar pasta dentro da “modulo 1” chamada “lista1”
- Fazer a Lista1 e commitar no git