**Estruturas Lógicas de Programação**  
  
Estruturas Condicionais:

|  |  |
| --- | --- |
| If (<condição>){  ...  }  If (<condição>){  ...  } else{  ...  } | switch<variável>:  {  case<>:  {  ...  {  Case<>:  {  ...  }  else:  {  ...  {  } |

|  |
| --- |
| Operadores:  == Igual  != Diferente  <= Menor ou igual  >= Maior ou igual  && e  || ou  Instance of |

value = (<condição> ? valor1: defaultvalue)

equivalente a:

if (<condição>){

value = valor1;

} else{

value = defaultvalue;

}

## Estruturas de repetição:

|  |  |
| --- | --- |
| while (<condição>){  ...  }  do {  ...  } while (<condição); | for (int i=0; i<3; i++){  ...  }  For Each:  int [] vetor = {1,2,3}  int soma=0;  for (int valor : vetor){  soma += valor;  } |

# **A classe String**

Não precisa fazer import porque está no pacote ‘Lang’.

java.lang.String

.Math

|  |  |
| --- | --- |
| Tipos primitivos:  int  float  boolean  double  char  byte  short  long | Tipos referência:  Os demais tipos que forem  definidos por classes. |

...

String palavra1 = “Luciana”;

String palavra2 = “Luciana”;

System.out,println(palavra1 == palavra2); true

System.out.println(palavra1.equals(palavra2)); true

String palavra3 = new String (“Luciana”);

System.out.println(palavra1==palavra3); false

## String são imutáveis!

Ex:

String temp = “lower case”;

String t2 = temp.toUpperCase():

System.out.println(temp); -> Retorna “lower case”

System.out.println(t2): -> Retorna “LOWER CASE”

# **Estrutura de dados**

Arrays (vetores)

* é um tipo referência

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |

int [] valor;

vetor = new int[3];

Carta [] baralho = new Carta [52]

for (Carta carta: baralho){

System.out.println(carta.getNaipe()):

System.out.println(carta.getValor()):

}

## Conversões

|  |  |  |
| --- | --- | --- |
| Tipo | Para String | De String |
| int  long  float  double | String.valueOf(int)  String.valueOf(long)  String.valueOF(float)  String.valueOf(double) | Integer.parseInt(String)  Long.parseLong(String)  Float.parseFloat(String)  Double.parseDouble(String |
| char  short  boolean  byte | ... | ... |

int [] [] matriz; == int matriz[] []

matriz2 = new int[4] [2];

# **Array List**

É uma coleção de alocação dinâmica. (tamanho variável)

import java.util.ArrayList;

ArrayList <Carta> baralho;

baralho = new ArrayList<>();

baralho.add(new carta (3,”espadas”);

System.out.print(baralho.get(0).getNaipe();

baralho.remove(carta);

System.out.println(baralho.size());

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| null | null | n | n | n | n | n | n | n | n |

## Para percorrer um ArreyList:

- for each

- for -> usando size()

- itarator()