Exemplos de classes: String, Integer, Float.

Certified Developer
The Ultimate Tech Degree

DigitalHouse >
Coding School

Temas

1 String

2 Integer

3 Float

1 String

String

Para usar dados de tipo de texto, vamos declará-los como String. Strings nos permitem usar funções já programadas que a pertencem. Nós os chamamos de métodos.

```
{}
public static void main(String[] args){
    String nome;
}
```

A partir dessa variável, veremos como usar alguns desses métodos usados com certa frequência.

Métodos usados: **.length(), .toUpperCase(), .equals(), .toChar()**

```
{}
public static void main(String[] args){
      String nome="Juan";
      int quantidade;
      char inicial;

      quantidade= nome.length();

      nome.toUpperCase();

      if (nome.equals("JUAN"))
      {
          System.out.println("Está em maiúscula");
      }
      inicial = nome.charAt();
}
```

{código}

```
String nome = ”Juan”;
int quantidade;
char inicial;

quantidade= nome.length();

nome.toUpperCase();

if (nome.equals(“JUAN”))
{
    System.out.println(“Está em maiúscula”);
}
inicial = nome.charAt();
```

Declaramos três tipos de variáveis, uma do tipo **String**, outra do tipo **int** e a última do tipo **char**. Atribuímos o nome da variável String ao nome "Juan".

{código}

```
String nome=”Juan”;
int quantidade;
char inicial;

quantidade= nome.length();

nome.toUpperCase();

if (nome.equals(“JUAN”))
{
    System.out.println(“Está em maiúscula”);
}
inicial = nome.charAt();
```

Usando o método .length () da classe String, podemos calcular o comprimento da string armazenada em nome.

{código}

```
String nome=”Juan”;
int quantidade;
char inicial;

quantidade= nome.length();
```

```
nome.toUpperCase();
```

Coloca todos os caracteres em maiúscula contidos no nome.

```
if (nome.equals(“JUAN”))
{
    System.out.println(“Está em maiúscula”);
}
inicial = nome.charAt();
```

{código}

```
String nome=”Juan”;
int quantidade;
char inicial;

quantidade= nome.length();

nome.toUpperCase();

if (nome.equals(“JUAN”))
{
    System.out.println(“Está em maiúscula”);
}
inicial = nome.charAt();
```

Verifica se ele foi convertido para maiúsculas corretamente. A string contida no nome da variável é exatamente "JUAN", lembre-se de que "Juan" não é igual a "JUAN".

{código}

```
String nome=”Juan”;
int quantidade;
char inicial;

quantidade= nome.length();

nome.toUpperCase();

if (nome.equals(“JUAN”))
{
    System.out.println(“Está em maiúscula”);
}
inicial = nome.charAt();
```

Uma mensagem é exibida, verificando se os caracteres estão em maiúsculas.

{código}

```
String nome=”Juan”;
int quantidade;
char inicial;

quantidade= nome.length();

nome.toUpperCase();

if (nome.equals(“JUAN”))
{
    System.out.println(“Está em maiúscula”);
}
inicial = nome.charAt();
```

Aqui obtemos o primeiro caractere da string, na variável inicial do tipo char.

String vazia

Se não atribuirmos nada às Strings, ela conterá um valor null. Nesse caso, os métodos não poderão ser utilizados.

```
{}
public static void main(String[] args){
      String nome;

      if (nome == null)
      {
          System.out.println("String com valor nulo");
      }

}
```

{código}

```
String nome;

if (nome==null)
{
    System.out.println(“String com valor nulo”);
}
```

Definimos o nome da variável, mas nada é atribuído a ela.

{código}

```
String nome;

if (nome==null)
{
    System.out.println("String com valor nulo");
}
```

Comprova se não foi inicializada.

{código}

```
String nome;

if  nome  null
{
    System.out.println("String com valor nulo");
}
```

Mensagem de indicação do valor nulo.

String vazia e String nula

Em uma String podemos ter as duas situações, ou pode ter um valor nulo ou estará vazia.

String nome;	String que não foi inicializada, terá valor nulo.
nome =“”;	String vazia.
nome=“Juan”;	String inicializada com valor “Juan”.

2 | Integer

Integer

Integer como **classe** e não como tipo primitivo tem maneiras diferentes de usar. Para usar uma Integer temos duas possibilidades:

```
{}  Integer valor = 0;
```

Neste caso, criamos e definimos uma Integer dando a ela um valor inicial de 0.

```
{}  Integer num = new Integer(1);
```

Na segunda forma, fazemos algo semelhante, mas a parte à esquerda é a definição e a parte da direita é a criação com um valor inicial 1.

Quando definimos apenas algo do tipo Integer, seu valor inicial é NULL, portanto é necessário dar um valor inicial.

Verificamos a relação entre dois números Integer, usando classes. Veja os métodos usados: **.equal()**, **.compareTo()**

```
public static void main(String[] args){
      Integer valor1 = 10;
      Integer valor2 = 30;
      int comparar;

      if (valor1.equals(valor2)){
          System.out.println("São iguais");
      }else{
            comparar=valor1.compareTo(valor2);
            if (comparar>0)
                  System.out.println("valor1 é maior que valor2");
            else
                  System.out.println("valor2 é maior que valor1");
      }
}
```

{código}

```
Integer valor1=10;
Integer valor2=30;
int comparar;
```

Definição das variáveis que vamos utilizar.

```
if (valor1.equals(valor2))
    System.out.println("São iguais");
else
{   comparar=valor1.compareTo(valor2);
    if (comparar>0)
        System.out.println("valor1 é maior que valor2");
    else
        System.out.println("valor2 é maior que valor1");
}
```

{código}

```
Integer valor1=10;
Integer valor2=30;
int comparar;

if (valor1.equals(valor2))
    System.out.println("São iguais");
else
{   comparar=valor1.compareTo(valor2);
    if (comparar>0)
        System.out.println("valor1 é maior que valor2");
    else
        System.out.println("valor2 é maior que valor1");
}
```

Comprovamos se são iguais.

{código}

```
Integer valor1=10;
Integer valor2=30;
int comparar;

if (valor1.equals(valor2))
    System.out.println(“São iguais”);
else
{   comparar=valor1.compareTo(valor2);
    if (comparar>0)
        System.out.println(“valor1 é maior que valor2”);
    else
        System.out.println(“valor2 é maior que valor1”);
}
```

Compare a relação entre os dois valores, se o valor1 for maior, ele dará 1, se o valor2 for maior, ele dará -1.

{código}

```
Integer valor1=10;
Integer valor2=30;
int comparar;

if (valor1.equals(valor2))
    System.out.println(“São iguais”);
else
{   comparar=valor1.compareTo(valor2);
    if (comparar>0)
        System.out.println(“valor1 é maior que valor2”);
    else
        System.out.println(“valor2 é maior que valor1”);
}
```

Mostra o resultado obtido na comparação.

3 | Float

Float

```
Float coeficiente=2.5f;
```

Neste caso, definimos e criamos um Float, dando a ele um valor inicial de 2,5f, o **f** significa Float, se não o colocarmos, assume-se que seja algo do tipo Double.

```
Float num= new Float(0.5);
```

Na segunda forma, fazemos algo semelhante, mas a parte à esquerda é a definição e a parte à direita a criação com um valor inicial de 0,5. Como o Integer, se não tiver um valor inicial, é considerado null.

Anotações

Atenção ao Float também!

Quando definimos apenas algo do tipo Float, seu valor inicial é NULL como no Integer, portanto é necessário dar um valor inicial.


DigitalHouse>
Coding School