Coleção Livro de Bolso

Programação básica em Java

Rogério Laureano Martins

Linguagem Java 5

Exibição de mensagem e textos 5

Variáveis 24

Classe Scanner e System.in 31
Operações Numéricas 34
Operações com texto 57
Operações com data 65
Transformações de caracter para número 93
Transformações de data para número 98
Transformações de número para data 101
Teste de variáveis e tomadas de decisões 104
Ciclos 119
Array de variáveis 125

Gravação e Leitura de arquivos 127

Gravação e leitura de arquivos texto 127

Comandos do sistema operacional 130

GUI – Interface gráfica com usuário 131

Encapsulamento 185

Encapsulamento 185

Banco de dados 198

Leitura de registros de dados.....198
Gravação de registros de dados.....200

JSP programação Web.....201

Conclusão.....204

Minha gratidão a Deus por me permitir compartilhar o que tenho aprendido. Minha gratidão à minha esposa Beatriz e ao meu filho João por terem paciência. Minha gratidão aos professores e youtubers que de alguma forma procura passar o conhecimento pessoal às pessoas.

Rogério Laureano Martins

Linguagem Java

A linguagem java é uma poderosa ferramenta de programação. Neste livro temos os conceitos básicos desta linguagem.

Exibição de mensagem e textos

O programa mais simples para qualquer linguagem é aquele que tem como objetivo retornar uma mensagem ao usuário.
Assim, para iniciarmos o estudo da linguagem Java, siga as instruções:

1 – Abra o programa Notepad
2 – Digite o seguinte fonte:

```
public class mensagem1
{
  public static void main(String[] args)
  {
    System.out.print("Olá Mundo");
  }
}
```

3 – Salve o arquivo como "C:\testeJava\mensagem1.java"
4 – Abra o programa cmd.exe
5 – Digite o comando "cd \ testeJava"
6 – Digite o comando "javac mensagem1.java"
7 – Digite o comando "java mensagem1"
8 – O sistema mostrará a seguinte tela:



Uma segunda maneira é utilizando a biblioteca iostream. Segue exemplo

```
public class mensagem2
{
  public static void main(String[] args)
  {
    System.out.println("Olá Mundo");
  }
}
```

Neste caso a diferença é que acrescentará uma linha no final do "Olá Mundo"

Outra maneira de ter este resultado, porém com mais recursos

1 – Abra o programa Notepad
2 – Digite o seguinte fonte:

```
public class mensagem3
{
  public static void main(String[] args)
  {
    System.out.printf("Olá Mundo");
  }
}
```

3 – Salve o arquivo como "C:\testeJava\mensagem1.java"
4 – Abra o programa cmd.exe
5 – Digite o comando "cd \ testeJava"
6 – Digite o comando "javac mensagem1.java"
7 – Digite o comando "java mensagem1"
8 – O sistema mostrará a seguinte tela:

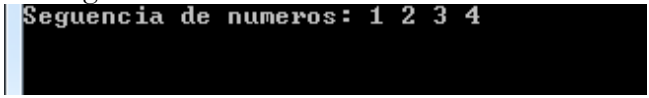


A função printf da linguagem java possibilita a exibição de mensagem. Tem a seguinte sintaxe:
printf(<caracteres> [, <valor 1>, <valor 2> ....])

O primeiro parâmetro é o texto a ser escrito. Os demais parâmetros são valores separados por "," que permitem substituir as expressões de controle. Exemplo:

```
public class mensagem4
{
  public static void main(String[] args)
  {
 System.out.printf(
  "Seguencia de numeros: %d %d %d %d",1,2,3,4);
  }
}
```

No fonte acima, o texto "Sequencia de números: " tem 4 expressões de controle "%d". Em seguida com os demais parâmetros que são "1, 2, 3 e 4". Assim o sistema exibirá a seguinte mensagem:



Abaixo a lista dos tipos de expressões de controle.

Expressão de controle	Funcionalidade
%d	Permite que seja efetuada a escrita de números inteiros
%e	Permite que seja efetuada a escrita de números em notação científica. Tem a opção de limitar a quantidade de dígitos após o ponto fluante. Se por exemplo for informado "%.2e" aparecerá a notação científica com dois dígitos após o ponto flutuante
%E	Igual ao %e. Porém com letras maiúsculas
%f	Permite que seja efetuada a escrita de números reais (números fracionados). Tem a opção de limitar a quantidade de dígitos após o ponto fluante. Se por exemplo for informado "%.2f" aparecerá a o número fracionado com dois dígitos após o ponto flutuante
%g	Permite que seja efetuada a escrita

Expressão de controle	Funcionalidade
	de números fracionados no formato mais curto. No caso da notação "%.2g" ou "%.3g" sempre exibirá o valor fracionado arredondado em 1 casa decimal a menos que a informada. Por exemplo: se o valor 2,1234 for colocado como "%.2g" retornará "2.1" se for colocado como "%.3g" retornará "2.12"
%G	Igual ao %g. Porém com letras maiúsculas
%o	Permite que seja efetuada a escrita de números octais
%s	Permite que seja efetuada a escrita de uma série de caracteres
%x	Permite que seja efetuada a escrita de um número hexadecimal
%X	Igual ao %x. Porém com letras maiúsculas
%p	Ponteiro em hexadecimal
%%	Imprime % no texto

Exemplo de utilização de um texto com várias expressões:

```
public class mensagem5
{
  public static void main(String[] args)
  {
    System.out.printf("teste %d %s %f",
                      1, "texto", 2.1);
  }
}
```

O sistema retornará o seguinte:



Exemplo de utilização das expressões de controle:

```
public class mensagem6
{
  public static void main(String[] args)
  {
  System.out.printf(
    "Informado '10'     Expressao d  : %d \n",
                    10);
  System.out.printf(
```

```
 "Informado '2.1'    Expressao e  : %e \n",
                  2.1);
System.out.printf(
"Informado '2.1'    Expressao .2e: %.2e \n",
                  2.1);
System.out.printf(
"Informado '2.1'    Expressao f  : %f \n",
                  2.1);
System.out.printf(
"Informado '2.1'    Expressao .2f: %.2f \n",
                  2.1);
System.out.printf(
"Informado '2.1234' Expressao g  : %g \n",
                  2.1234);
System.out.printf(
"Informado '2.1234' Expressao.2g : %.2g \n",
                  2.1234);
System.out.printf(
"Informado '2.1234' Expressao.3g : %.3g \n",
                  2.1234);
System.out.printf(
"Informado '2.1634' Expressao.2g : %.2g \n",
                  2.1634);
System.out.printf(
"Informado '2.1634' Expressao.3g : %.3g \n",
                  2.1634);
System.out.printf(
"Informado '10'     Expressao o  : %o \n",
                  10);
System.out.printf(
"Informado 'teste'  Expressao s  : %s \n",
                  "teste");
System.out.printf(
```

```
  "Informado '15'     Expressao x  : %x", 15);
  }
}
```

O sistema exibirá:

```
C:\testeJava>java mensagem6
Informado '10'      Expressao d  : 10
Informado '2.1'     Expressao e  : 2,100000e+00
Informado '2.1'     Expressao .2e: 2,10e+00
Informado '2.1'     Expressao f  : 2,100000
Informado '2.1'     Expressao .2f: 2,10
Informado '2.1234'  Expressao g  : 2,12340
Informado '2.1234'  Expressao.2g : 2,1
Informado '2.1234'  Expressao.3g : 2,12
Informado '2.1634'  Expressao.2g : 2,2
Informado '2.1634'  Expressao.3g : 2,16
Informado '10'      Expressao o  : 12
Informado 'teste'   Expressao s  : teste
Informado '15'      Expressao x  : f
```

Note que neste último script, além de ter as expressões de controle, tem também os caracteres "\n" ao final do primeiro parâmetro. Este é o caracter especial que representa a quebra de linha.

Abaixo a listagem dos caracteres especiais da função printf.

Caracter especial	Funcionalidade
\n	Gera uma nova linha
\t	Gera um espaço de tabulação.

Caracter especial	Funcionalidade
\b	Executa um retrocesso de espaço
\”	Adiciona o símbolo aspa
\\	Adiciona o símbolo “\”
\f	No caso de impressora, gera uma quebra de página
\0	Adiciona nulo

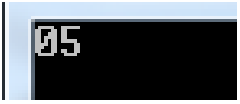
Se notar, também tem as expressões semelhantes a “%.2f”. Ou seja ainda temos outras possibilidades para exibição da mensagem. Na verdade a expressão de controle é composta da seguinte maneira:
%[opção][largura][.precisão][tamanho]tipo.
Os tipos já foram mencionados acima e são obrigatório. Exemplo: %s, %d, %f.

As opções são as seguintes:
0 (zero) – Permite, em combinação com a largura, definir zeros à esquerda de valores numéricos.

```
public class mensagem7
{
```

```
  public static void main(String[] args)
  {
  int teste = 5;
  System.out.printf("%02d",teste);
  }
}
```

O sistema exibe a mensagem:



A opção "-" permite alterar o alinhamento conforme o tamanho do campo:

```
public class mensagem8
{
  public static void main(String[] args)
  {
  String teste = "teste";
  System.out.printf("%-10s\n",teste);
  System.out.printf("%10s",teste);
  }
}
```

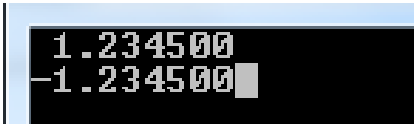
O sistema exibirá a mensagem:



Ou seja, o primeiro alinhou pela esquerda (invertendo o alinhamento padrão) e o segundo pela direita (mantedo o alinhamento padrão)
A opção “ ” (espaço) reserva um caracter para sinal de negativo para números menores que zero.

```
public class mensagem9
{
  public static void main(String[] args)
  {
  double numero = 1.2345;
  printf("% f\n",numero);
  numero = -1.2345;
  printf("% f",numero);
  }
}
```

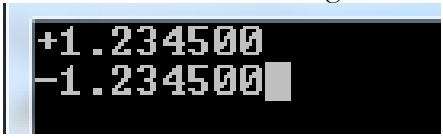
O sistema exibirá a mensagem:



A opção "+" coloca o sinal dos números positivos "+" e negativos "-"

```
public class mensagem10
{
  public static void main(String[] args)
  {
  double numero = 1.2345;
  printf("%+f\n",numero);
  numero = -1.2345;
  printf("%+f",numero);
  }
}
```

O sistema exibirá a mensagem:

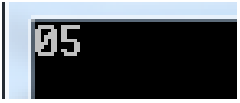


A opção (‘) permite formatar números com pontos de milhar, milhão, bilhão etc. Porém, funciona no UNIX.

O parâmetro de opções são estes citados acima. O segundo parâmetro possível é o de largura. Já vimos nos exemplos das opções “0” e “-” seu funcionamento. A largura permite limitar a exibição pela quantidade.

```
public class mensagem11
{
  public static void main(String[] args)
  {
  int teste = 5;
  printf("%02d",teste);
  }
}
```

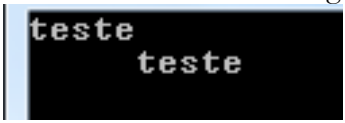
O sistema exibe a mensagem:



Neste exemplo foi definido uma largura de “2” completando com “0” à esquerda.

```
public class mensagem12
{
  public static void main(String[] args)
  {
  String teste = "teste";
  System.out.printf("%-10s\n",teste);
  System.out.printf("%10s",teste);
  }
}
```

O sistema exibirá a mensagem:



Neste exemplo foi definido uma largura de “10” com alinhamento invertido para primeira linha “teste ”. Na segunda linha uma largura de 10 com alinhamento padrão “ teste”.

O terceiro parâmetro é a precisão. Para valores numéricos fracionados, é limitada a quantidade de dígitos após o ponto fracionado.

```
public class mensagem13
{
  public static void main(String[] args)
  {
  double numero = 1.2345;
  printf("%f\n",numero);
  printf("%.2f",numero);
  }
}
```

O sistema exibirá a mensagem



No exemplo acima, o sistema exibiu o valor arredondando para baixo. Ou seja, com valor (1,23).

Caso a fração que será omitida seja maior ou igual a "5" o sistema arredonda para cima.