

# **Programação Orientada a Objetos**

**JAVA**

**Caderno de Exercícios**

# Programação Orientada a Objetos

---

## Lista 1

- 1) O Java foi criado a partir de quais linguagens de programação?
  
  
  
  
  
  
  
  
  
  
- 2) Quais as principais razões que levaram os engenheiros da Sun a desenvolver uma nova linguagem de programação?
  
  
  
  
  
  
  
  
  
  
- 3) Por que um sistema escrito em Java pode rodar em qualquer plataforma?
  
  
  
  
  
  
  
  
  
  
- 4) Marque com "X" na alternativa correta. Para rodar uma aplicação Java, por mais simples que seja, é necessário possuir uma Java Virtual Machine.  
( ) Verdadeiro ( ) Falso
  
  
  
  
  
  
  
  
  
  
- 5) Qual o principal papel do JCP (Java Community Process)?

Anotações

2

---

---

---

---

# Programação Orientada a Objetos

---

6) Por quem é formado o JCP?

7) Marque “V” para verdadeiro ou “F” para falso.

- a. ( ) A Oracle, como dona da tecnologia poderá mudar os rumos do Java a qualquer tempo.
- b. ( ) Para a Oracle fazer uma alteração no Java será necessário se submeter ao JCP.
- c. ( ) O JCP é o responsável por defender os interesses da indústria da comunidade Java e da Oracle.

8) Como está estruturada a plataforma Java?

9) O que é Java?

**Anotações**

3

# Programação Orientada a Objetos

---

10) Marque com um "X" na alternativa correta.

É possível compilar um código Java para uma plataforma específica?  
( ) Verdadeiro ( ) Falso

11) Quais as principais características do Java?

12) Qual a função do Garbage Collector?

13) Quais são as três tecnologias Java para desenvolvimento de aplicativos?

**Anotações**

4

---

---

---

---

# Programação Orientada a Objetos

---

14) O Java é compilado ou interpretado?

15) Para que serve a Java Virtual Machine?

**Anotações**

5

# Programação Orientada a Objetos

---

## Lista 2

- 1) Quais são as convenções estabelecidas para a declaração de classes, métodos e variáveis?
  
  
  
  
  
  
  
  
  
  
- 2) Como inserimos comentários em um código Java?
  
  
  
  
  
  
  
  
  
  
- 3) Um dos principais motivos que contribuíram para o desenvolvimento da linguagem Java foi:
  - a. ☐ O nome da linguagem.
  - b. ☐ O desenvolvimento da Internet.
  - c. ☐ A linguagem ser relativamente simples.
  - d. ☐ O desempenho da linguagem em termos de velocidade.
  
  
  
  
  
  
  
  
  
  
- 4) Por que o aspecto da utilização do Java em multiplataforma é muito importante para os programadores?
  
  
  
  
  
  
  
  
  
  
- 5) Qual das características seguintes não diz respeito a linguagem Java:
  - a. ☐ Pode ser executada em qualquer computador, independente de existir uma máquina virtual java instalada.

**Anotações**

6

---

---

---

---

# Programação Orientada a Objetos

---

- b. ☐ É uma linguagem compilada e interpretada.
  - c. ☐ O desempenho dos aplicativos escritos em Java, com relação à velocidade de execução, é inferior à maioria das linguagens de programação.
  - d. ☐ É uma linguagem com um bom nível de segurança.
- 6) A sequência de desenvolvimento de um programa em Java é:
- a. ☐ Compilação, digitação e execução.
  - b. ☐ Digitação, execução e compilação.
  - c. ☐ Digitação, compilação e execução.
  - d. ☐ Digitação, execução e testes de funcionamento.
- 7) Qual a principal característica que distingue a plataforma Java das demais existentes?
- 8) Para a linguagem Java, as variáveis PATH e CLASSPATH correspondem a:
- a. ☐ Variáveis usadas em um programa Java.
  - b. ☐ Uma variável de ambiente e um caminho para a execução dos programas Java.
  - c. ☐ Um caminho para encontrar as classes e um caminho para encontrar os aplicativos da linguagem Java.
  - d. ☐ Um caminho para encontrar os aplicativos e um caminho para encontrar as classes da linguagem Java.
- 9) Qual a diferença entre uma variável do tipo primitivo e uma variável do tipo reference?
- 10) Quais são os tipos primitivos da linguagem Java?

## Lista 3

### Anotações

7

# Programação Orientada a Objetos

---

1) O que são variáveis locais?

2) Dado o código a seguir:

```
public class App1{  
    public static void main(String args[ ]){  
        String s1 = args[1];  
        String s2 = args[2];  
        String s3 = args[3];  
        String s4 = args[4];  
        System.out.println("args[2] = " + s2);  
    }  
}
```

e a chamada de linha de comando sendo

**java App1 1 2 3 4**

Qual o resultado? (Selecione um)

- a. args[2] = 2;
- b. args[2] = 3;
- c. args[2] = null;
- d. A compilação falhará;
- e. Uma exceção será lançada no tempo de execução.

3) Dado o código a seguir,

**Anotações**

8



# Programação Orientada a Objetos

---

```
public class Foo {  
    public void main( String[] args ) {  
        System.out.println( "Hello" + args[0] );  
    }  
}
```

e a chamada de linha de comando sendo:

**java Foo world**

Qual o resultado? (Selecione um)

- a. Hello
- b. Hello Foo
- c. Hello world
- d. A compilação falhará;
- e. O código não executará.

4) Dado os códigos abaixo, analise a sintaxe e assinale (V) para verdadeiro (F) para falso.

- a. ☐ int x = 10.45;
- b. ☐ float f = 3.4;
- c. ☐ char a = "S";
- d. ☐ boolean z = true; if(z){ }

## Lista 4

Anotações

9

# Programação Orientada a Objetos

---

1) Crie um programa que recebe três nomes quaisquer por meio da linha de execução do programa, e os imprima na tela da seguinte maneira: o primeiro e o último nome serão impressos na primeira linha um após o outro, o outro nome (segundo) será impresso na segunda linha.

**Dica:**

```
String n1, n2, n3;  
n1 = args[0];
```

2) Faça um programa que receba a quantidade e o valor de três produtos, no seguinte formato: quantidade1 valor1 quantidade2 valor2 quantidade3 valor3. O programa deve calcular esses valores seguindo a fórmula  $\text{total} = \text{quantidade1} \times \text{valor1} + \text{quantidade2} \times \text{valor2} + \text{quantidade3} \times \text{valor3}$ . O valor total deve ser apresentado no final da execução.

**Dica:**

```
int total, q1, v1, ..., v3;  
q1 = Integer.parseInt(args[0]);  
v1 = Integer.parseInt(args[1]);  
int total = (q1 x v1) + (q2 x v2) + .....
```

3) Crie um programa que receba a largura e o comprimento de um lote de terra e mostre a área total existente.

4) Crie um programa que receba quatro valores quaisquer e mostre a média, somatório entre eles e o resto da divisão do somatório por cada um dos valores.

**Dica:**

```
total = (a+b+c+d);  
media = (total/4);
```

## Lista 5

Anotações

10

# Programação Orientada a Objetos

---

- 1) Faça um aplicativo que receba três valores inteiros na linha de comando e mostre o maior dentre eles.
- 2) Faça um programa que apresente o total da soma dos cem primeiros números inteiros ( $1+2+3+\dots+99+100$ ).
- 3) Faça um aplicativo que calcule o produto dos inteiros ímpares de 1 a 15 e exiba o resultado na tela.
- 4) Crie uma classe que gere um numero aleatório entre os valores máximo e mínimo recebidos do usuário na linha de comando.

# Programação Orientada a Objetos

---

## Lista 6

- 1) O que é um array?
  
  
  
  
  
  
  
  
  
  
- 2) Como um array unidimensional pode ser declarado?
  
  
  
  
  
  
  
  
  
  
- 3) Todo array declarado como uma variável local deve ser inicializado.  
  
( ) Verdadeiro      ( ) Falso
  
  
  
  
  
  
  
  
  
  
- 4) Todo array é uma variável do tipo:  
  
( ) Primitivo      ( ) Reference
  
  
  
  
  
  
  
  
  
  
- 5) Como obter o tamanho de um array?
  
  
  
  
  
  
  
  
  
  
- 6) Como um array Bidimensional pode ser declarado?

## Lista 7

Anotações

12

---

---

---

---

# Programação Orientada a Objetos

---

- 1) Uma escola precisa de um programa que controle a média das notas dos alunos de cada classe e a média das notas de todos os alunos da escola. Sabendo que essa escola possui 3 classes com 5 alunos em cada classe, gerando um total de 15 notas, crie um programa que receba as notas de cada aluno de cada classe e no final apresente a média de cada classe e a média da escola em geral.
  
- 2) Qual a função da classe String?
  
- 3) Qual método da classe String retorna o tamanho da String?
  
- 4) Qual método da classe String converte qualquer tipo de dados em String?

## Lista 8

Anotações

13

---

---

---

---

# Programação Orientada a Objetos

---

- 1) Crie um aplicativo que receba uma frase e mostre-a de forma invertida.
- 2) Crie um aplicativo que mostre o efeito abaixo:

J
Ja
Jav
Java
Jav
Ja
J

- 3) Crie uma classe que leia um parâmetro passado na linha de comando no seguinte formato: dd/mm/aaaa. Desta maneira, a classe devera ser executada como java Exe04 11/09/2001. A saída gerada por essa execução deve ser a impressão separada do dia, do mês e do ano - utilizando apenas os métodos da classe String.
- 4) Uma empresa quer transmitir dados por telefone, mas está preocupada com a possibilidade de seus telefones estarem grampeados. Todos seus dados são transmitidos como inteiros de quatro dígitos. Eles pedem para você escrever um programa que criptografará seus dados de modo que estes possam ser transmitidos mais seguramente. Seu aplicativo deve ler um inteiro de quatro dígitos inserido pelo usuário na linha de comando e criptografá-lo como segue:  
substitua cada dígito por (a soma deste dígito mais 1). Então troque o primeiro dígito pelo terceiro e troque o segundo pelo quarto. A seguir imprima o inteiro criptografado. Escreva um aplicativo separado que recebe como entrada um inteiro de quatro dígitos criptografado e o descriptografa para formar o número original.

## Lista 9

### Anotações

14

# Programação Orientada a Objetos

---

1) O fatorial de um número inteiro não negativo  $n$  é escrito como  $n!$  (pronuncia-se fatorial de  $n$ ) e é definido como segue:

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 1 \text{ (para valores de } n \text{ maiores que ou iguais a } 1\text{)}$$

e

$$n! = 1 \text{ (para } n=0\text{)}$$

Por exemplo:  $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$ , o que dá 120.

Escreva um aplicativo que lê um inteiro não negativo via linha de comando, computa e imprima seu fatorial.

2) Escreva um aplicativo que recebe entradas de texto e envia o texto para saída com letras em maiúsculas e em minúsculas.

3) A série de Fibonacci 0,1,1,2,3,5,8,13,21.....  
inicia com 0 e 1 e tem a prioridade de que cada número de Fibonacci subsequente é a soma dos dois anteriores que o procedem.

Escreva um aplicativo que recebe a entrada do número de vezes que deve ocorrer a série.

4) Faça um aplicativo que verifique se uma palavra é um palíndromo. Ex: Ana.

## Lista 10

Anotações

15

# Programação Orientada a Objetos

---

Leia o que está sendo solicitado e implemente os códigos:

## Exercício 1

### Passo 1:

- Criar a classe Agencia
- Adicione os seguintes atributos na classe Agencia:  
nrAgencia(String)  
codBanco(int)

### Passo 2:

- Criar a classe TestaAgencia
- Crie um objeto da classe Agencia
- Inicialize todos os atributos deste objeto.
- Imprima os valores dos atributos da classe Agencia de forma a obter o seguinte resultado:  
\* -----  
\* AGENCIA : 1  
\* BANCO : 234  
\* -----

## Exercício 2

### Passo 1:

- Criar a classe Cliente
- Adicione os seguintes atributos na classe Agencia:  
nomeCliente(String)  
cpfCliente(String)

### Passo 2:

- Criar a classe TestaCliente
- Crie um objeto da classe Cliente
- Inicialize todos os atributos deste objeto.
- Imprima os valores dos atributos da classe Cliente de forma a obter o seguinte resultado:  
\* -----  
\* NOME :FULANO  
\* CPF : 234232323  
\* -----

## Exercício 3

### Passo 1:

- Criar a classe Conta
- Adicione os seguintes atributos na classe Conta

Anotações

16



# Programação Orientada a Objetos

---

```
saldo(double)
nrAgencia(String)
titular(String)
nrConta(String)
codBanco(int)
```

## **Passo 2:**

- Criar a classe TestaConta
- Crie um objeto da classe Conta
- Inicialize todos os atributos deste objeto.
- Imprima os valores dos atributos da classe Cliente de forma a obter o seguinte resultado:

```
* -----
* AGENCIA: 1   BANCO : 234
* CONTA CORRENTE : 01945
* TITULAR: FULANO
* SALDO : R$10000.0
* -----
```

## **Lista 11**

**Anotações**

17

# Programação Orientada a Objetos

---

**Explique os seguintes conceitos :**

Objeto:

Classe:

Atributo:

Método:

## Lista 12

**Anotações**

18

# Programação Orientada a Objetos

---

Leia o que está sendo solicitado nos códigos e os implemente.

```
/*  
 * 1) Implemente os métodos que não foram implementados na classe Conta de  
acordo com a especificação nos métodos.  
*/
```

```
public class Conta {
```

```
    saldo(double)  
    nrAgencia(String)  
    titular(String)  
    nrConta(String)  
    codBanco(int)
```

```
    // crie o método construtor
```

```
/**  
 * @param valor: valor a ser sacado da conta  
 * 1. Verificar se o valor do saque é positivo.
```

**Anotações**

19

# Programação Orientada a Objetos

---

- \* 2. Verificar se ha saldo suficiente para efetuar o saque
- \* 2.1. Se o saldo for suficiente, efetuar o saque
- \* 2.2. Se o saldo for insuficiente imprimir na tela que o saldo e Insuficiente
- \*/

```
public void saque(double valor) {
```

```
}
```

```
/**
```

```
* @param valor Valor a ser depositado da conta
```

```
* Verificar se o valor do deposito e positivo.
```

```
*/
```

```
void deposito(double valor) {
```

```
}
```

```
/**
```

```
* Metodo para impressao de todos os dados da classe
```

```
*/
```

**Anotações**

20

# Programação Orientada a Objetos

---

```
public void imprimeDados() {
    System.out.println("\n-----");
    System.out.println("AGENCIA:\t"+nrAgencia+"\t BANCO:\t"+codBanco);
    System.out.println("Conta: \t"+nrConta);
    System.out.println("TITULAR: \t"+titular);
    System.out.println("SALDO: \t"+saldo);
    System.out.println("-----\n");
}

/**
 * @return saldo da conta
 */
double getSaldo() {
    return saldo;
}
}
```

**public class TestaConta {**

```
public static void main(String[] args) {
    // Criacao da conta
```

```
    // Inicializacao da conta
```

```
    // Impressao dos dados da conta
```

```
    // Saque da conta
```

**Anotações**

21

# Programação Orientada a Objetos

---

```
// Impressao dos dados da conta
```

```
// Deposito em conta
```

```
// Impressao dos dados da conta
```

```
// Impressao do saldo da conta, utilizando o metodo getSaldo();
```

```
}  
}
```

## Lista 13

**Anotações**

22

---

---

---

---

# Programação Orientada a Objetos

---

1) Considere o aplicativo chamado **Administracao** e a classe **Populacao**, apresentados a seguir e codifique as questões de 1 a 4:

```
public class Populacao{
```

```
    private int pop[ ][ ];
```

```
    public int estados, municipios;
```

- 1) Codificar neste quadro o construtor, da classe, que recebe como parâmetros o número de estados e o número de municípios, e cria a matriz de populações.

```
    public void atualizarPopulacao(int i, int j, int populacao){  
        if (i>=0 && i<4 && j>=0 && j<5 && populacao > 0)  
            pop[i][j] = populacao;  
    }
```

- 2) Codificar neste quadro o método que determina a população média de um dado estado.

```
}
```

```
public class Administracao{  
    public static void main (String p[ ]){
```

Anotações

23

# Programação Orientada a Objetos

---

3) Declarar variáveis

```
for (i=0; i<4; i++){  
    for (j=0; j<5; j++){  
        n = Integer.parseInt(JOptionPane.showInputDialog  
        ("Informe a população da cidade " +  
        String.valueOf(j+1) + "\ndo estado " +  
        String.valueOf(i+1));  
        pop.atualizarPopulacao(i, j, n);  
    }  
}
```

4) Codificar, neste espaço, a parte do aplicativo que recebe via janela de diálogo o número de um estado e exibe, também via janela de diálogo, a população média deste estado.

Utilizar para isto uma matriz de população de 4 estados com 5 municípios cada, gerada através da classe Populacao.

```
}  
}
```

Anotações

24



## Programação Orientada a Objetos

---

2) Implemente uma classe chamada Carro com as seguintes propriedades:

**a.** Um veículo tem um certo consumo de combustível (medidos em km/litro) e uma certa quantidade de combustível no tanque.

**b.** O consumo é especificado no construtor e o nível de combustível inicial é 0.

**c.** Forneça um método andar( ) que simule o ato de dirigir o veículo por uma certa distância, reduzindo o nível de combustível no tanque de gasolina.

**d.** Forneça um método getCombustivel( ), que retorna o nível atual de combustível.

**e.** Forneça um método setCombustivel( ), para abastecer o tanque.

**f.** Escreva um pequeno programa que teste sua classe. Exemplo de uso:  
Carro uno(16); // 16 quilômetros por litro de combustível.

uno.setCombustivel(20); // abastece com 20 litros de combustível.

uno.andar(150); // anda 150 quilômetros.

uno.getCombustivel() // Exibe o combustível que resta no tanque.

## Programação Orientada a Objetos

---

3) Escreva uma classe chamada Aluno que contenha os atributos privados denominados nome, matricula, nota1, nota2, nota3, peso1, peso2 e peso3. Além disso,

- a.** Crie um construtor-padrão para a classe.
- b.** Crie um construtor que inicialize todos os membros de dados com os valores recebidos como argumento.
- c.** Crie os métodos de acesso (*getters* e *setters*) para todos os atributos. Os métodos *setters* devem validar os dados de entrada conforme as regras definidas abaixo:
  - o nome deve conter pelo menos dois caracteres e não deve possuir números;
  - a matrícula de ser constituída de 9 dígitos apenas;
- d.** Escreva um método público para calcular a média ponderada das três notas;

# Programação Orientada a Objetos

---

4) Escreva um programa que leia o nome e salário atual de um funcionário. O programa deve calcular seu novo salário (segundo a tabela abaixo) e mostrar o nome, o salário atual e o salário reajustado do funcionário:

Acréscimo		
Tabela de Reajuste		
De	Até	
--	150,00	25%
150,00	300,00	20%
300,00	600,00	15%
600,00	--	10%

- a. Leia um valor **N** inteiro pelo teclado e realize todo o processo acima descrito para os **N** funcionários;
- b. mostrar ao final do programa a soma dos salários atuais, a soma dos salários reajustados e a diferença entre eles.

Anotações

27

# Programação Orientada a Objetos

---

## Lista 14

1) Crie uma aplicação que simule o cálculo do valor final de uma venda, dependendo da forma de pagamento escolhida pelo usuário. O usuário entra com um valor, escolhe a forma de pagamento e o cálculo do preço final é realizado conforme os seguintes critérios: para pagamento em dinheiro, desconto de 5%, para pagamento em cheque, acréscimo de 5%, para pagamento com cartão, acréscimo de 10%. A figura abaixo apresenta a janela de execução deste exercício.

A janela 'Cálculo do preço final' possui um título azul com ícone de calculadora. O formulário contém: um campo de texto para 'Entre com o valor da venda:'; um menu suspenso para 'Escolha a forma de pagto:' com 'Cartão' selecionado; um campo de texto para 'Preço final a pagar'; e dois botões, 'Calcular' e 'Limpar', no rodapé.

2) Crie uma aplicação que simule o cadastramento de pessoas. O usuário digita o nome e endereço de uma pessoa, escolhe o sexo e o estado civil por meio de componentes do tipo Combo. Ao pressionar o botão mostrar, todos os dados cadastrados são copiados para um componente TextArea, conforme apresenta a figura abaixo.

A janela 'Cadastramento de Pessoas' possui um título azul com ícone de pessoas. O formulário contém: campos de texto para 'Nome:' (contendo 'Jadir') e 'Endereço:' (contendo 'Rua a'); menus suspensos para 'Sexo' (com 'Masculino' selecionado) e 'Estado Civil' (com 'Casado' selecionado); e dois botões, 'Mostrar' e 'Limpar', no rodapé. Abaixo dos botões, há uma área de texto (TextArea) que exibe os dados cadastrados: 'Nome: Jadir', 'Endereço: Rua a', 'Sexo: Masculino' e 'E.C.: Casado'.

Anotações

---

---

---

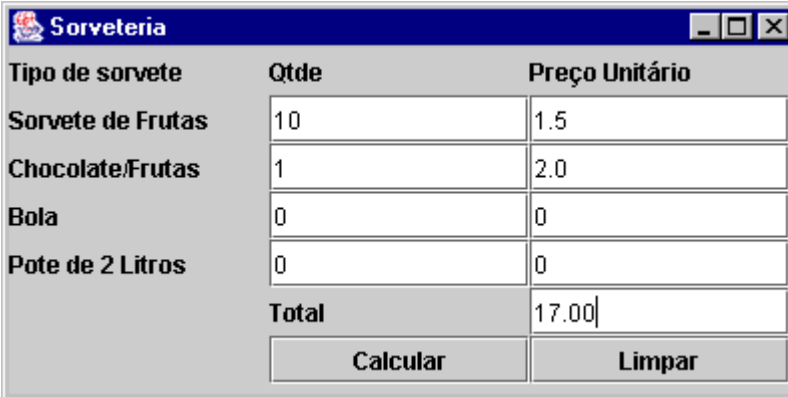
---

28

## Programação Orientada a Objetos

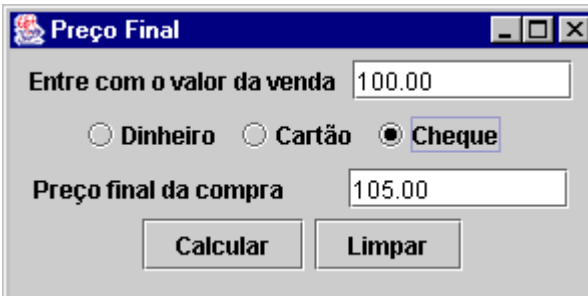
---

3) Crie uma aplicação que simule vendas de sorvete em um sorveteria, de acordo com o apresentado na figura abaixo.



Tipo de sorvete	Qtde	Preço Unitário
Sorvete de Frutas	10	1.5
Chocolate/Frutas	1	2.0
Bola	0	0
Pote de 2 Litros	0	0
<b>Total</b>		17.00
<b>Calcular</b>		<b>Limpar</b>

4) Crie uma aplicação que simule o cálculo do valor final de uma venda, dependendo da forma de pagamento escolhida pelo usuário. O usuário entra com um valor, escolhe a forma de pagamento e o cálculo do preço final é realizado conforme os seguintes critérios: para pagamento em dinheiro, desconto de 5%, para pagamento em cheque, acréscimo de 5%, para pagamento com cartão, acréscimo de 10%. A figura abaixo apresenta a janela de execução deste exercício.



<b>Preço Final</b>	
Entre com o valor da venda	100.00
<input type="radio"/> Dinheiro <input type="radio"/> Cartão <input checked="" type="radio"/> Cheque	
Preço final da compra	105.00
<b>Calcular</b>	<b>Limpar</b>

Anotações

---

---

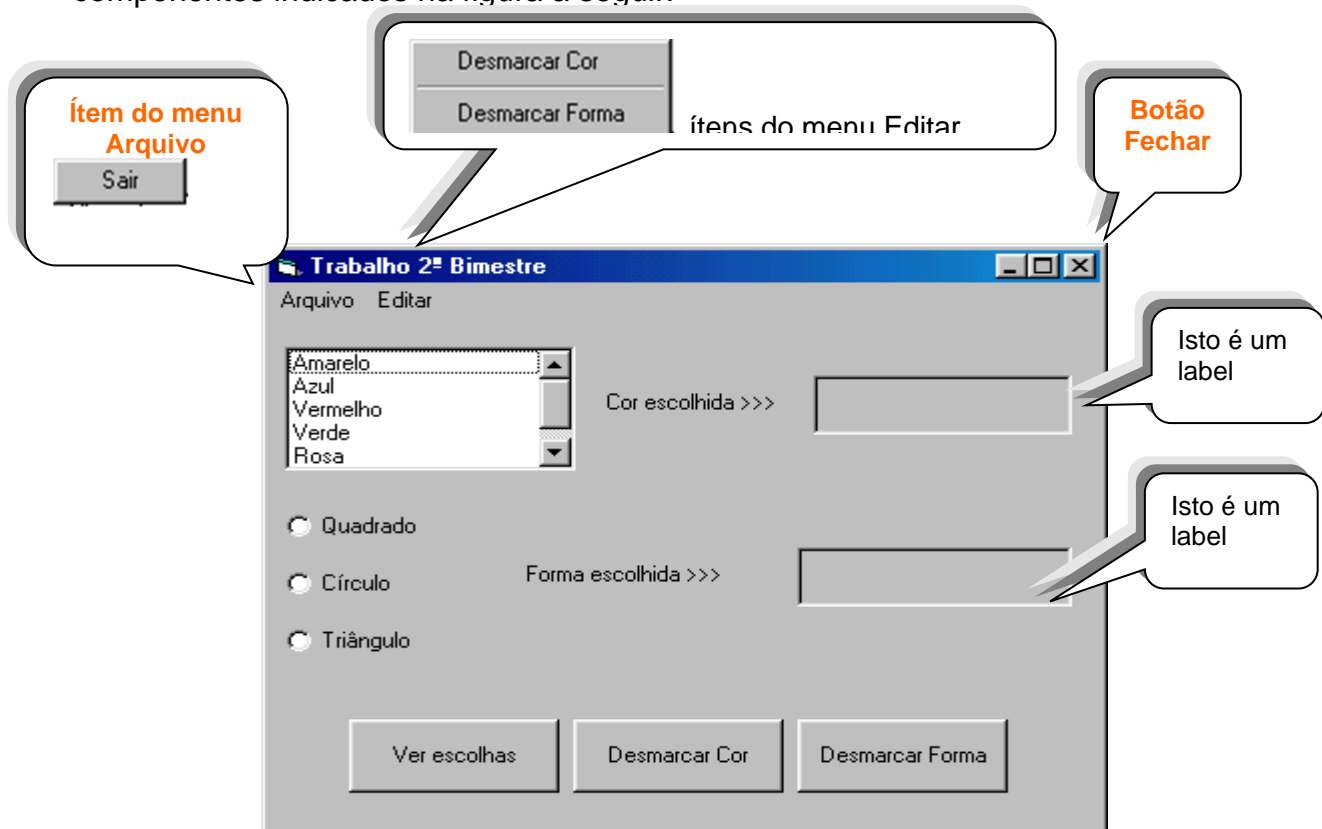
---

---

29

# Programação Orientada a Objetos

5) Codificar uma classe MinhaJanela (extensão de JFrame), que contenha os componentes indicados na figura a seguir:



## Observações a respeito da janela:

- ✎ A lista de cores deve ser carregada dinamicamente com cores que serão fornecidas via linha de comando do aplicativo que utilizar esta janela.

## Lista 15

### Anotações

30

# Programação Orientada a Objetos

---

Implemente um aplicativo usando as Classes Swing e JDBC, conforme a tela abaixo:

IMPORTANTE:

Nome da base de dados: **Biblioteca**

Nome da tabela: **Leitor**

Nomes dos campos:

**codLeitor** – numérico de 4 bytes

**nomeLeitor** – texto de 40 bytes

**tipoLeitor** – texto de 20 bytes

Manutencao do Leitor

Código do Leitor

Nome do Leitor

Tipo de Leitor

Professor  
Aluno

Novo Gravar Consultar Alterar Excluir

Banco Conectado com sucesso

---

Anotações

31