

# Introdução à Orientada à Objetos

## Lista de Exercícios

Programação Orientada à Objetos - UNIVILLE

Utilize os seus conhecimentos de orientação à objetos para resolver os problemas solicitados.

### **1 Problema**

O que é um objeto?

### **2 Problema**

Identifique 3 características e 3 comportamentos de um carro.

### **3 Problema**

No contexto de Orientação a Objetos, as características e comportamentos são chamados respectivamente de?

### **4 Problema**

Qual é o objetivo da programação orientada à objetos?

### **5 Problema**

O que é abstração? Cite um exemplo.

### **6 Problema**

O que é classe?

## 7 Problema

Qual é o padrão utilizado para nomear as classes? Cite um exemplo.

## 8 Problema

Qual opção apresenta corretamente o nome da classe *carro elétrico*?

1. carroeletrico
2. carroEletrico
3. Carroeletrico
4. CarroEletrico
5. carro\_eletrico

## 9 Problema

Qual é o padrão utilizado para nomear os atributos? Cite um exemplo.

## 10 Problema

Qual opção apresenta corretamente o nome do atributo *cor de fundo*?

1. CordeFundo
2. CorDeFundo
3. corDeFundo
4. \_cor\_de\_fundo
5. \_corDeFundo
6. mCorDeFundo

## 11 Problema

Qual é o padrão utilizado para nomear os métodos? Cite um exemplo.

## 12 Problema

Qual opção apresenta corretamente o nome do método *está vazio*?

1. estaVazio

2. `_estaVazio`
3. `estavazio`
4. `EstaVazio`
5. `Estavazio`
6. `Esta_vazio`

## 13 Problema

No contexto de orientação à objetos, o que é um objeto? Em que momento existe um objeto? Quando ele deixa de existir?

## 14 Problema

Qual é o objetivo do operador *new*?

## 15 Problema

O que é o construtor? Qual é o seu objetivo? Qual deve ser o seu nome? Cite um exemplo.

## 16 Problema

Caso o construtor da classe *Aluno* não for declarado, qual será o seu construtor?

## 17 Problema

Crie um exemplo de instanciação da classe *Aluno*. Utilize o construtor padrão.

## 18 Problema

O que é encapsulamento?

## 19 Problema

Qual é o objetivo do modificador de acesso *public*?

## 20 Problema

Qual é o objetivo do modificador de acesso *private*?

## 21 Problema

Como é aplicado o encapsulamento em uma classe? Considere a classe *Aluno* com o atributo matrícula.

## 22 Problema

Qual o objetivo dos métodos *getters*? Crie um exemplo.

## 23 Problema

Qual o objetivo dos métodos *setters*? Crie um exemplo.

## 24 Problema

O diagrama de classe UML é composto por 3 partes. O que vai em cada parte?

## 25 Problema

Qual é o padrão utilizado para representar um atributo no diagrama de classe UML? Crie um exemplo.

## 26 Problema

Qual é o padrão utilizado para representar um método no diagrama de classe UML? Crie um exemplo.

## 27 Problema

Como o construtor de uma classe pode ser diferenciado no diagrama de classe UML? Crie um exemplo.

## 28 Problema

Quais são os símbolos utilizados no diagrama de classe UML para representar os modificadores de acessos aos atributos e métodos? Crie um exemplo.

## 29 Problema

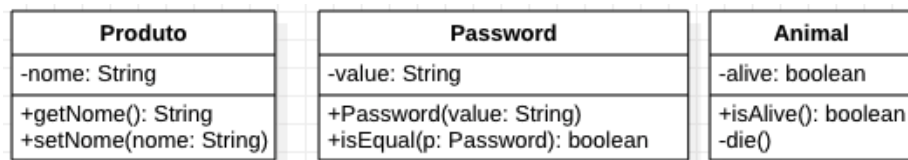
Considere a classe *Cliente*, com os atributos *nome*, *email* e *telefone* com os respectivos métodos *getters* e *setters*. Desenvolva o diagrama de classe UML.

## 30 Problema

Considere a classe *Cliente* apresentado no Problema 29, e desenvolva o código Java correspondente.

## 31 Problema

Desenvolva o código Java das classes do apresentadas no diagrama de classes UML abaixo:



## 32 Problema

Desenvolva o diagrama de classe dos códigos Java abaixo.

```
public class Livro{
    private String nome;
    public String getNome(){
        return nome;
    }
    public void setNome(String nome){
        this.nome = nome;
    }
}

public class ContaCorrente{
    private double saldo;
```

```

    public double sacar(double valor){
        return 0.0;
    }
    public void depositar(double valor){
        this.saldo = saldo;
    }
    private void recalcularSaldo(){
    }
}

public class Par{
    public String chave;
    public String valor;
}

public class Impressora{
    public void imprimir(Documento documento){

    }
}

```

### 33 Problema

O que é o estado de um objeto? Cite um exemplo com a classe *Aluno* com os atributos *nome*, *idade*, *matrícula* e *curso*. Utilize o diagrama de estado de objeto.

### 34 Problema

Qual é o estado do objeto da classe *Dog* quando é inicializado? Desenvolva o diagrama de objetos.

```

public class Dog{
    private int years;
    private String name;
    private boolean alive;
    //Getters e setters suprimido
}

```

### 35 Problema

Qual é o estado do objeto *dog* no final da execução do método *main*? Desenvolva o diagrama de objetos.

```

1: public static void main(String args[]){
2:     Dog dog = new Dog();

```

```
3:         dog.setYears(10);
4:         dog.setName("Spike");
5:         dog.setAlive(true);
6: }
```

## 36 Problema

Qual é o estado do objeto *pug* e *bulldog* após a execução da linha 6? Desenvolva o diagrama de objetos.

```
1: Dog pug = new Dog();
2: Dog bulldog = new Dog();
3: pug.setName("Spoke");
4: bulldog.setName("Spike");
5: pug.setAlive(true);
6: pug.setYears(2);
7: bulldog.setName("Floquinho");
8: pug.setYears(1);
```

## 37 Problema

Analise o código abaixo. Verifique se existem problemas, caso sim, indique o problema e sugere as correções.

```
1: public class Cat{
2:     private String name;
3:     public double weight;
4:     // Getters e Setters suprimidos
5: }
6: public class Main{
7:     public static void main(String args[]){
8:         Cat c = new Cat();
9:         c.weight = 3.5;
10:        c.name = "BlackCat";
11:    }
12: }
```