1. Construtores:

- **O que são:** Construtores são métodos especiais em uma classe que são chamados automaticamente quando um objeto da classe é instanciado. Eles são responsáveis por inicializar o estado inicial do objeto.

- **Características**:
    - Um construtor tem o mesmo nome que a classe a que pertence.
    - Pode ou não ter parâmetros.
    - Pode ser sobrecarregado, ou seja, uma classe pode ter mais de um construtor com diferentes listas de parâmetros.
    - Não possuem tipo de retorno explicitamente declarado.
    - Podem inicializar os valores dos atributos da classe ou realizar outras operações necessárias durante a criação do objeto.

Exemplo de Código:

```java
public class Carro {
    private String marca;
    private String modelo;
    private int ano;

    // Construtor sem parâmetros
    public Carro() {
        this.marca = "Sem marca";
        this.modelo = "Sem modelo";
        this.ano = 0;
    }

    // Construtor com parâmetros
    public Carro(String marca, String modelo, int ano) {
        this.marca = marca;
        this.modelo = modelo;
        this.ano = ano;
    }
}
```

Exercício:
Escreva um programa que utilize a classe `Carro` e crie objetos utilizando ambos os construtores. Exiba os detalhes de cada carro criado.

2. Sobrecarga de Métodos:

- **O que é:** Sobrecarga de métodos é a capacidade de uma classe ter vários métodos com o mesmo nome, mas com diferentes listas de parâmetros.

- **Características:**
    - A sobrecarga de métodos permite que você use o mesmo nome de método para realizar operações semelhantes em diferentes tipos de dados ou com diferentes números de argumentos.
    - A assinatura do método, que inclui o nome do método e o tipo, ordem e número de seus parâmetros, deve ser diferente para cada método sobrecarregado.
    - A decisão sobre qual método sobrecarregado chamar é feita pelo compilador com base nos argumentos fornecidos no momento da chamada do método.

Exemplo de Código:

```java
public class Calculadora {
    // Método para somar dois inteiros
    public int somar(int a, int b) {
        return a + b;
    }

    // Método sobrecarregado para somar dois decimais
    public double somar(double a, double b) {
        return a + b;
    }
}
```

Exercício:
Adicione mais um método à classe `Calculadora` para somar três números inteiros. Em seguida, escreva um programa que utilize esse método para realizar uma operação de soma.

3. Encapsulamento:

O que é: Encapsulamento é um dos princípios fundamentais da programação orientada a objetos, que consiste em ocultar os detalhes de implementação de uma classe e fornecer uma interface pública para interagir com ela.

- **Benefícios:**
    - Promove a modularidade e a reutilização de código.
    - Protege os dados de uma classe, permitindo que apenas métodos específicos (métodos getter e setter) acessem e modifiquem os valores dos atributos.

- Permite que a implementação interna de uma classe seja alterada sem afetar outras partes do código que a utilizam.

- **Métodos Acessores:**
  - Métodos getter: usados para recuperar o valor de um atributo.
  - Métodos setter: usados para definir ou modificar o valor de um atributo.

O uso correto desses conceitos contribui para a construção de código mais robusto, coeso e fácil de entender e manter. Eles são fundamentais para o desenvolvimento de software orientado a objetos eficaz em Java.

Exemplo de Código:

```java
public class Pessoa {
    private String nome;
    private int idade;

    // Métodos getter e setter para o atributo nome
    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    // Métodos getter e setter para o atributo idade
    public int getIdade() {
        return idade;
    }

    public void setIdade(int idade) {
        this.idade = idade;
    }
}
```

Exercício:
Escreva um programa que utilize a classe `Pessoa` e demonstre o uso dos métodos getter e setter para acessar e modificar os atributos privados da classe.