No Java, "interface" e "class" são conceitos fundamentais, mas servem a propósitos diferentes e têm características distintas. Aqui estão as principais diferenças entre eles:

Classe (Class)

1. **Definição**: Uma classe é um modelo ou blueprint para criar objetos. Ela define atributos (variáveis) e métodos (funções) que os objetos criados a partir dessa classe terão.
2. **Herança**: Uma classe pode herdar de outra classe (herança de classe única). Isso significa que uma classe pode estender outra classe e herdar seus atributos e métodos.
3. **Implementação**: Uma classe pode conter métodos concretos (com implementação) e abstratos (sem implementação, se a classe for abstrata).
4. **Construtores**: Uma classe pode ter construtores para inicializar seus objetos.
5. **Modificadores de Acesso**: Uma classe pode ter métodos e atributos com diferentes níveis de acesso (public, private, protected, default).
6. **Instanciação**: Uma classe pode ser instanciada para criar objetos.

Interface (Interface)

1. **Definição**: Uma interface é um contrato que define um conjunto de métodos que uma classe deve implementar. Ela não contém implementação de métodos, apenas suas assinaturas.
2. **Herança**: Uma interface pode herdar de múltiplas interfaces (herança múltipla). Isso permite que uma interface estenda várias outras interfaces.
3. **Implementação**: Antes do Java 8, uma interface só podia conter métodos abstratos (sem implementação). A partir do Java 8, interfaces podem conter métodos default e static com implementação.
4. **Construtores**: Uma interface não pode ter construtores.
5. **Modificadores de Acesso**: Todos os métodos em uma interface são implicitamente públicos e abstratos (a menos que sejam métodos default ou static).
6. **Instanciação**: Uma interface não pode ser instanciada diretamente. Ela deve ser implementada por uma classe.

Exemplos

Classe

```
public class Animal {
    private String nome;

    public Animal(String nome) {
        this.nome = nome;
    }

    public void fazerSom() {
        System.out.println("Som genérico de animal");
    }
}
```

Interface

```
public interface Animal {
    void fazerSom();
}
```

Implementação de Interface

```
public class Cachorro implements Animal {
    @Override
    public void fazerSom() {
        System.out.println("Au Au");
    }
}
```

Resumo

- **Classe**: Define atributos e métodos, pode ser instanciada, pode ter herança de classe única.
- **Interface**: Define um contrato de métodos, não pode ser instanciada, pode ter

herança múltipla.

Essas diferenças permitem que classes e interfaces sejam usadas de maneiras complementares para criar sistemas flexíveis e modulares em Java.