### Comparativa entre Classe e Interface:

**Definição de classe**

Uma classe é um aspecto dos mais importantes em Java, pois cria tipos de referência e os mesmos são usados para criar objetos. O corpo de uma classe possui atributos e métodos, por vezes já implementados. Cada objeto instanciado na classe, contém um cópia de todos os campos da classe, seguindo o molde, incluindo os atributos e métodos. Os métodos e atributos estáticos são constantes e comuns a todos os objetos daquela classe, já os não-estáticos ficam com características individuais para cada objeto. Os membros de uma classe podem ter diferentes especificadores de acesso como público, privado e protegidos. Uma classe pode herdar e ser herdada por outra classe, permitindo assim que os atributos e métodos sejam passados de uma classe para outra. Só é permitido herdar de uma única classe, mas é possível que uma classe pai herde para diversas classes filhos.

**Definição de interface**

Uma interface também é um aspecto dos mais importantes em Java, principalmente por criar relações entre classes, visto não ser possível a herança múltipla(herdar de mais de uma classe), assim sendo um substituto aplicável. Uma interface tem um corpo muito semelhante ao da classe, contudo apenas são declarados os atributos e métodos, não são de fato implementados, isso significa que os métodos estarão vazios, não haverão constanstes, mas sim declarações de atributos e métodos com método de acesso público que deverão ser implementados nas classes que decidam aplicar determinada interface, sendo possível que uma classe implemente mais de uma interface. Neste contexto, devemos lembrar que é obrigatório que a classe implemente cada uma das características declaradas na interface o que causará erro se alguma funcionalidade da interface nºao for implementada. Deste modo, não é possível instanciar objetos na interface, pois a mesma comporta-se como uma espécie de contrato que vincula a classe, a interface prevê e declara todos os atributos e métodos e a classe implementa cada um deles. Por consequência, na interface, diferentemente das classes, não deverá conter qualquer construtor. Ainda é possível uma interface herdar ou ser herdada de outra interface.

| Base para comparação | Classe | Interface |
| --- | --- | --- |
| Base | Uma classe é instanciada para criar objetos. | Uma interface nunca pode ser instanciada, pois os métodos não podem executar nenhuma ação na chamada. |
| Utilização | class nome\_classe{atributos...métodos} | interface nome\_interface{tipo\_atributos...tipo\_métodos()} |
| Especificador de acesso | Os membros de uma classe podem ser privados, públicos ou protegidos. | Os membros de uma interface são sempre públicos. |
| Métodos | Os métodos de uma classe são definidos para executar uma ação específica. | Os métodos em uma interface são puramente abstratos. |
| Implementar / estender | Uma classe pode implementar qualquer número de interface e pode estender apenas uma classe. | Uma interface pode estender várias interfaces, mas não pode implementar nenhuma interface. |
| Construtor | Uma classe pode ter construtores para inicializar as variáveis. | Uma interface nunca pode ter um construtor, já que praticamente não existe nenhuma variável para inicializar. |

**Exemplo de interface:**

public interface Poligono {  
 double calcularArea(double raio);  
  
 double calcularPerimetro(double raio);  
}

**Exemplo de classe que implementa uma interface;**

public class Circunferencia implements Poligono {  
 public static final double *pi* = 3.14159265359;  
 public double raio;  
  
 public Circunferencia(double raio) {  
 this.raio = raio;  
 }  
  
 @Override  
 public double calcularArea(double raio) {  
 return raio \* raio \* *pi*;  
 }  
  
 @Override  
 public double calcularPerimetro(double raio) {  
 return raio \* 2 \* *pi*;  
 }  
}

**Exemplo de implementação completa do programa:**

import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 double raio;  
 System.*out*.println("\nVamos calcular a área e perímetro de uma circunferência. Por favor, informe o raio da circunferência:");  
 raio = Double.*parseDouble*(sc.nextLine());  
 Circunferencia circulo = new Circunferencia(raio);  
 System.*out*.println("\nÁrea da circunferência: " + circulo.calcularArea(raio) + ".");  
 System.*out*.println("\nPerímetro da circunferência: " + circulo.calcularPerimetro(raio) + ".");  
 }  
}