Interface

Programação Orientada a Objetos

emerson@paduan.pro.br

Interface

Uma interface em Java pode ser entendido como um modelo para uma classe.

Mas especialmente é usada para permitir "baixo acoplamento" em um projeto de software.

De forma segura não expor detalhes de implementação

Pode ser usada para se realizar uma abstração ou conseguir "herança múltipla".

emerson@paduan.pro.br

Interface

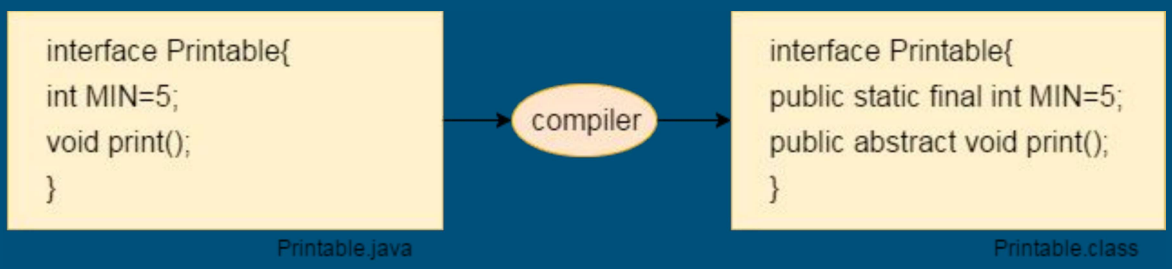
Declaração de uma interface:

interface <interface_name> {

}

Interface

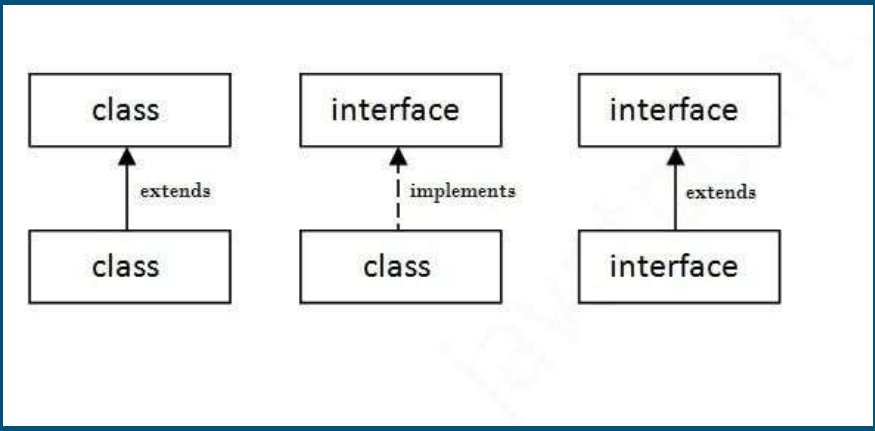
Por padrão, todo campo ("atributo") é public, static e final, e todo método é public e abstract.



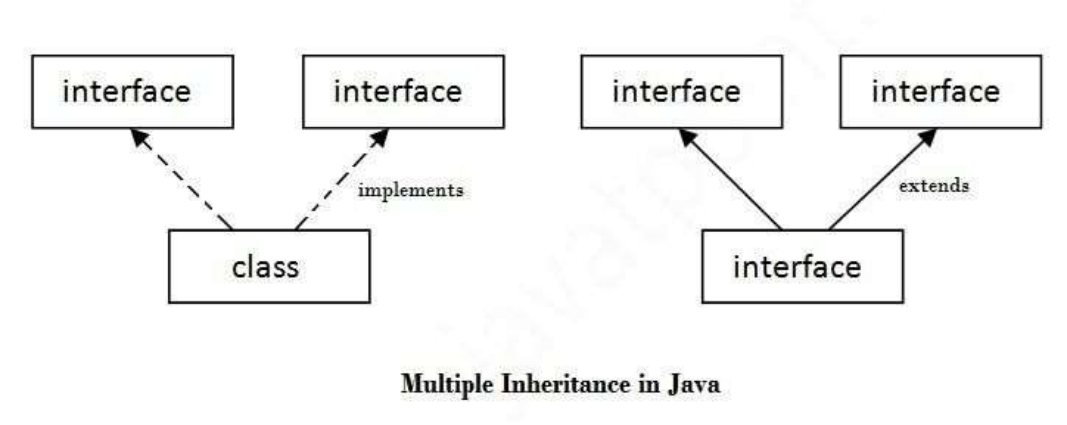
* A partir do Java 8 podemos ter métodos default e static.
* A partir do Java 9, também private

classe abstrata	Interface
Não pode instanciar objetos	Não pode instanciar objetos
Podemos fazer apenas um extends	Podemos fazer múltiplos implements
Pode conter atributos como uma classe	Todos os atributos são public, static e final
Contém construtor	Não contém construtor
Pode conter implementação de métodos, ou métodos abstratos	Contém apenas métodos abstratos (apenas declaração) *(Java 8+ pode ter default e static)
Pode conter qualquer modificador de acesso	Por padrão todos os métodos são public e abstract (Java 8+ pode ter default e static) (Java 9 pode também private)
Não suporta herança múltipla	Suporta "herança múltipla"

emerson@paduan.pro.br



emerson@paduan.pro.br



Multiple Inheritance in Java

Exemplo 1

```
interface printable {
        void print();
}

class Escrita implements printable {
        public void print() {
                System.out.println("Hello");
        }
}

public static void main(String args[ ] ) {
        Escrita obj = new Escrita ( );
        obj.print( );
 }
```