Uma interface, é uma classe que não implementa os seus métodos, mas sim delega a implementação de outras classes. Uma interface nunca pode derivar de uma classe, mas pode ser derivada de outra interface. Os métodos declarados na interface costumam ser abstratos e os seus atributos são finais. A interface somente pode conter métodos vazios e/ou constantes.

Podemos utilizar a interface para gerar relação entre classes, que de outro modo, não seria possível, ou em distribuir pacotes de classes, indicando a estrutura da interface, mas não as das classes individuais.

Já nas classes, temos 4 tipos diferentes, tais como *Classes Abstratas*, *Classes Aninhadas*, *Classes Locais* e *Classes anonimas*.   
Que podemos usar para definir as constantes e os métodos que podem ser utilizados na interface, visto que não podemos declarar variáveis e criar objetos.

Utilizando a interface conseguimos criar ligações entre as classes, assim facilitando a leitura do código, e retirando do caminhos todo o processo desnecessário, o que tornaria o processamento mais lento ao ler o código.

Exemplo:

**interface** Radio{

///Na interface só podemos declarar métodos abstratos e os seus métodos são finais. A interface apenas contem métodos vazios ou constantes.

[abstract] void ligar();

void desligar()

void aumentarVolume (int quantid);

void diminuirVolume(int quantid);

void frequenciaRadio(double canal);

int getAtualFreq();

}

///Como não podemos criar funções na interface, utilizamos classes para atribuir as variáveis que serão atribuídas mais tarde a interface.

***class <nome\_classe> [extends <classe base>]***

***[implements <lista\_interfaces>]***

class Docooler **implements** Radio{

int volume, canal;

static final int MIN\_VOLUME = 0;

static final int MAX\_VOLUME = 100;

static final double MIN\_CANAL = 87.7;

static final double MAX\_CANAL = 108.0;

void apagar (){

System.out.println (“Radio off”);

///Podemos definir dentro da classe os objetos e atribuir as constantes, que serão utilizadas na interface. Tornando assim o código mais fácil de processar e organizando as funções, de forma a serem apenas utilizadas quando necessárias.

}

void ligar(){

System.out.println(“Radio on”);

}

void aumentarVolume (int quantid){

if (volume + quantid > MAX\_VOLUME)

volume = MAX\_VOLUME;

else volume += quantid;

}

void baixarVolume (int quantid){

if (volume – quantid > MIN\_VOLUME)

volume = MIN\_VOLUME;

else volume -= quantid;

}

void frequenciaRadio (doube canal){

if (canal < MIN\_CANAL || canal > MAX\_CANAL)

return;

this.canal = canal;

}

void getAtualFreq(){

return canal;

}

}