23/3/2022 version 1.2  
Appunti iniziali

the interface of a subroutine (o metodo) consists of the name of the subroutine, its return type, and the number and types of its parameters (signature, firma di un metodo).

This is the information you need to know if you want to call il metodo -the subroutine.

public Contatto cerca(String cognome)  oppure public Contact search(String cognome)

A subroutine/metodo  also has an implementation: the block of code which defines it and which is executed when the subroutine is called.

public Contatto cerca(String cognome){

// codice che serve a fare questa ricerca

}

In Java, **interface** is a reserved word with an additional, technical meaning.

An "interface" in this sense consists of a set of instance method interfaces, without any associated implementations.

A class can implement an interface by providing an implementation for each of the methods specified by the interface.

Here is an example of a very simple Java interface:

……interfaceRubrica

This looks much like a class definition, except that the implementation of the

Contatto cerca(String cognome) method is omitted.

A class that implements the interface interfaceRubrica must provide an implementation for Contatto cerca(String cognome).

Of course, the class can also include other methods and variables.

Note that to implement an interface, a class must do more than simply provide an implementation for each method in the interface;  it must also state(dichiarare) that it implements the interface, using the reserved word **implements** as in this example:

.

Any concrete class that implements the interfaceRubrica interface must define a Contatto cerca(String cognome) instance method.

class Rubrica **implements** interfaceRubrica{

//i suoi attributi

//i suoi costruttori

//i suoi metodi

public Contatto cerca(String){

  //siamo obbligati a scrivere il codice di questo metodo

}

}