Full Developer

Beginner

Functions vs Procedures

Functions

The functions are routines which perform some work or specific algorithm to solve the problem and return one of the following values:

- **Value (Number, Alphanumeric, Etc.)**
- **Function (Object Function)**
- **Array**
- **Set of Values**

Procedures

The procedures are routines which perform a work or algorithm meanwhile other functions do not return any value, receive parameters, each one of them is a different type of data, for example::

- **Int**
- **Double**
- **Float**
- **Date**
- **Char**

Procedures

The routine is a series of commands where it performs an algorithm or problem, and it tries to solve such problem which till the end of this same routine does not return any value. It receives parameters, and some of them are optional, others are mandatory.

Classes

Encapsulation

The encapsulation is for containing into one entity everything which it can do such entity, for example:

- Class
    - Car
- Property
    - Color
    - Type of Insurance
- Method
    - Turn On
    - Stop
    - Park

Constructor

The constructor is a intrinsic method of work which perform when it creates the instance of object. At the moment that creates an instance of class, it calls to this method by default. This used commonly to create the instance of the object.

The constructor can be overwritten in different ways so that it can give different instances of the same object, and from there, it can be called in different ways for creating different instances of the same class.

Constructor

It means that it can create different similar objects of the same class.

Destructor

The destructor is an intrinsic method that calls when it destroys the instance of an object. It is useful to free an space of memory and of this manner which the program do not full of trash.

In this way, the program can work in a light way without not so much load and this is the way that can continue till finishes the program.

Destructor

In the contrast, it can full in of much memory or the use of memory without using, and from there can finish the program much before. And this does not happen very much when it works with volumes of data or tasks very big.

This is precise, that this kind of particular method. From there, the program can finalize of different much way and with major performance. This method is important without it. the program could delay in many occasions.

Method

This method is an action of the class. And it can be called after it creates an instance of the object, from there each method can be called to solve some particular problem, and you can create as many methods as possible.

Each one of them, it will realize a different task and on this way it will solve many problems during the execution of the program.

Method

Such beginning or ending of program, is very important its task, because of that, it is used such many times during the running of the program.

If inside this method, it is used many resources of the system, we should consider that we have something to free those resources because they are blocks or chunks or memory.

Visibility

Visibility

The visibility is a feature that each property or method of the class has, for example:

- **Private**
  - **It means when it can not be accessed from outside. Only it can be called from the class.**
- **Protected**
  - **It means when it can not be accessed from outside. Only it can be called from the same instance of the object.**

Visibility

- **Public**
  - **It means when it can be accessed from outside. After the creation of the instance of the object.**

Overwrite

It means when it repeats the same method but with more parameters. In other words, and in other words a bit more technical it has the same signature.

And it can be overwritten such many times as you wish, however we sould be careful with it, because if we repeat it many times, it could create confusions at the moment that another developer tries to follow it or oneself wants to follow it.

Overwrite

It means when it repeats the same method but it has the same parameters. In other words, and a bit more technical it has the same signature but with different <type> of parameters.