CLASES PRINCIPALES

1. EQUIPOS

Equipos está construida como una clase abstracta por la sencilla razón de que existe un método común que deben implementar todos los objetos “equipo”, pero este debe ser implementado de ditas formas según el equipo que se cree.

* MÉTODOS:

lang(self, fm = 1): Como bien he comentado antes existe un método abstracto que deben implementar todas las demás clases que hereden de Equipos.

Lang\_true(self,C,fm = 1): Este método es común a todas las clases que hereden de objeto.

William(self): Este método es común a todas las clases que hereden de objeto.

* CONSTRUCTOR:

\_\_init\_\_(self,\_coste\_ref = 0, \_capacity = 0,\_capacity\_ref=0, n= 0, \_installed = True): Todos los parámetro por defecto se ponen por default a 0 por la sencilla razón de que existen objetos en los que se conoce su capacidad de antemano. En esos casos estos parámetros deberán ser modificados al instanciar el objeto equipo.

* SUBCLASES

Las clases que heredan de la clase Equipo serán todos los equipos que hay o los que se quieran crear.

En estas subclases se implementa el método abstracto lang, antes mencionado.

1. PROCESOS

En esta clase me ha surgido una duda de diseño. No sabía si todos los procesos requieren los mismos operational costs y si los modelos financieros de todos los procesos son iguales, por esta razón he supuesto que todos los procesos usan los mismos operational costs y usan el mismo modelo financiero. En el caso de que esto no fuese así la solución que propongo es hacer algo similar a lo descrito anteriormente con la Clase Equipo que sería una clase abstracta y de la que se tendría que crear una clase por cada proceso a crear.

* MÉTODOS

No hace falta que describa muy en detalle los métodos aquí utilizados ya que son los mismos que se me proporcionaban con el código de inicio.

* CONSTRUCTOR