Class Storage{

Int amount //cantidad de los acomodados

Float x //eje x

Float y //eje y

Float z //eje z

Float weight //peso de los acomodados

}

Class ListStorage:List<Storage>{

Int totalAmount

Int constZ;

//Int amountDesired

StorageKind( , , , , , , elements) // c = contenedor, e = elemento

}

StorageKind( , , , , , , elements){

= max(kinds);

if(elements<=)

return 1 //acomodo completado satisfactoriamente

If (< 1)

Return -1; //espacio insuficiente en la caga

Else{

Storage storage;

If (){

storage = new Storage( , , , )

this.add(storage)

this.totalAmount +=

r = this.storageKind( , , , , , )

this.constZ = z;

}

Else If (){

storage = new Storage( , , , )

this.add(storage)

this.totalAmount +=

r = this.storageKind( , , , , , )

this.constZ = z;

}

Else If (){

storage = new Storage( , , , )

this.add(storage)

this.totalAmount +=

r = this.storageKind( , , , , , )

this.constZ = y;

}

Else If (){

storage = new Storage( , , , )

this.add(storage)

this.totalAmount +=

r = this.storageKind( , , , , , )

this.constZ = x;

}

}

Return r

}

Para conocer el acomodo de un determinado número de elementos de dimensiones en una caja de dimensiones hacer:

ListStorage box = new ListStorage(); //lista de acomodos de elementos en caja

box.StorageKind() //esto retorna 1 si en la caja caben los n elementos o -1 en caso contrario, por lo tanto

BoxNumber = /box.getTotalAmount(); //esto es el numero de cajas que se necesitan.

Para conocer el acomodo de un determinado número de cajas de dimensiones en un pallet de dimensiones o en un pallet de dimensiones . Hacer:

ListStorage pallet1 = new ListStorage(); //lista de acomodos de cajas en pallet

ListStorage pallet2 = new ListStorage(); //lista de acomodos de cajas en pallet