Universidad de Jaén
Departamento de Informática

Programación Orientada a Objetos

https://tiny.cc/ujapoo

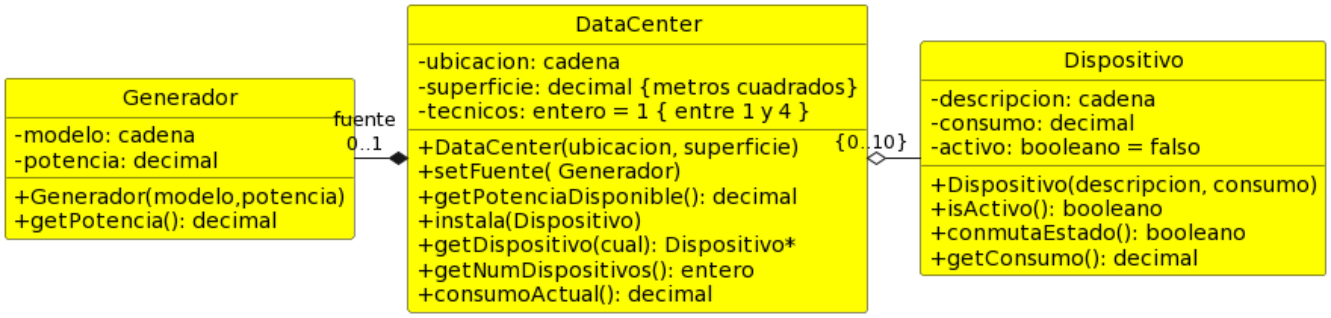
SEMINARIO DE PREPARACIÓN DE LA PRUEBA PRÁCTICA 1

Instrucciones

- Rellene la parte superior de esta hoja indicando su nombre, apellidos y el puesto que ocupa.
- **Dispone de 1 hora y 40 minutos** para la realización de los ejercicios propuestos.
- **Material:** Sólo puede hacer uso para la realización de la actividad del material de teoría y guiones de prácticas accesible en el espacio virtual de la asignatura. También dispone de código fuente de algunas partes del problema que debe completar siguiendo los ejercicios propuestos. El profesor le aclarará la sintaxis de cualquier duda que tenga sobre cualquier función o clase de la biblioteca estándar de C++ que necesite para la implementación siempre y cuando este considere que es necesaria para su realización. Para la edición del código, compilación y depuración debe utilizarse el ordenador del aula con cualquiera de los entornos de desarrollo y compiladores instalados.
- **Recomendaciones**: grabe continuamente el código generado en disco para evitar pérdidas de información en caso de bloqueo de la máquina. Si esto ocurriera, notifíquelo al profesor para que le indique cómo proceder. Vaya implementando, compilando y probando cada ejercicio de forma individual. Si no sabe cómo implementar algún método, implemente al menos su cabecera y deje el contenido en blanco o comentado. Al menos podrá llamarlo desde donde proceda y continuar realizando el resto de ejercicios. Si encuentra errores de compilación que no es capaz de resolver, comente el fragmento de código que genere los problemas y siga trabajando con el resto de ejercicios.
- **Está prohibido expresamente** utilizar cualquier otro material impreso, en formato digital o accesible de forma on-line para la realización del ejercicio. Tampoco se pueden realizar copias del presente enunciado o del material realizado en cualquier formato ni sacarlas del aula utilizando cualquier soporte de almacenamiento o servicio telemático. Cualquier incumplimiento de estas normas supondrá el suspenso inmediato del ejercicio.
- Al finalizar la actividad, **entregue todos los ficheros de código fuente y el último ejecutable obtenido**, comprimidos en un fichero en **formato .ZIP** a través del espacio virtual de su grupo de prácticas, en la actividad creada a tal efecto. **Entregue también esta hoja** debidamente cumplimentada.
- *No olvide borrar del ordenador todo el código realizado al finalizar su trabajo* y después de asegurarse de que lo ha enviado correctamente. Si tiene dudas al respecto, pregúntele al profesor si ha realizado correctamente la entrega.
- *Podrá acceder a todo el material que entregue así como a este enunciado a partir del día siguiente a la realización de esta actividad* en el espacio virtual de su grupo de prácticas.

Descripción de la actividad

CloudPOO es una solución para la monitorización remota de *DataCenters*. Su primera versión permitirá a los clientes controlar la actividad y consumo energético de los dispositivos albergados en alguno de sus Datacenters físicos. Completar el código C++ suministrado a partir del siguiente diseño UML y las funcionalidades descritas en los ejercicios. Para comprobar su correcto funcionamiento, **realizar las pruebas descritas en la función *main***.



Ejercicios

NOTA: si el proyecto no compila, la nota obtenida se penalizará en un 10%

1.- (1,5 puntos) Definir e implementar la clase Datacenter en su correspondiente módulo (.h, .cpp, *no in-line*) con los atributos y visibilidad indicados, además de los necesarios para establecer las relaciones del diagrama. Implementar un **constructor parametrizado** que permita añadir exclusivamente la ubicación y la superficie del datacenter.

2.- (1 punto) Implementar métodos para **consultar los atributos simples** de un DataCenter. Implementar un método para **modificar el número de técnicos** asignados al DataCenter según la restricción indicada en el diagrama. Si el valor no es correcto lanzar una excepción *std::out_of_range*.

> **Prueba 1 (0.4 puntos)**: Crea **un Datacenter** con 2 técnicos asignados **y tres Dispositivos** con los siguientes datos**:** *Rack de servidores de 500W, RAID de discos de 750W, Unidad de backup de 250W*.

3.- (1,5 puntos) Implementar los métodos necesarios en la **clase DataCenter** para **asignarle un generador** y **consultar la potencia total disponible** en el DataCenter. Esta potencia es la potencia del generador asignado o, lógicamente, 0 si no tuviera. El generador solo puede asignarse una vez al DataCenter. Si se intenta asignar una segunda vez se deberá lanzar una excepción *std::invalid_argument*. El generador debe destruirse junto con el DataCenter.

> **Prueba 2 (0.6 puntos)**: **Crea dos Generadores** de 1500W **y trata de asignárselos al Datacenter** (la segunda asignación no será posible pues el DataCenter ya tendrá un generador). Captura y muestra la información de la excepción. Muestra la potencia del *DataCenter*

4.- (1,5 puntos) Implementar métodos para **instalar, consultar y contabilizar los dispositivos en un DataCenter**. Los Dispositivos de un Datacenter se podrán consultar por su número de orden a la hora de instalarlos, e.g. el nº 1, el 2, etc. Si no se pudiera obtener un Dispositivo, se lanzará una excepción *std::out_of_range*. Solo se admitirán nuevos dispositivos si no se ha alcanzado su límite y se dispone de un generador.

5.- (0.5 puntos) Especificar en doxygen el método para **instalar un Dispositivo** en un DataCenter.

> **Prueba 3 (0.5 puntos)**: **instala los tres dispositivos en el dataCenter,** capturando cualquier excepción que pueda lanzar.

6.- (1 punto) Implementar el **constructor de copia de la clase DataCenter**.

7.- (1 punto) Implementar el método **DataCenter::consumoActual()** que devuelva la suma del consumo de todos los dispositivos que están activos.

> **Prueba 4 (0.5 puntos)**: Crea un DataCenter **que sea copia del que ya existe. Para la copia,** activa sus dos primeros dispositivos, visualiza el consumo actual del *DataCenter* y el número de dispositivos que tiene instalados.