



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada  
1er semestre 2015

## Actividad 3

### Polimorfismo - Properties

El Campus San Joaquín cuenta con diversos espacios como las salas de clases, salas de reuniones, patios, subterráneos, etc. Cada **Espacio** tiene sus propias dimensiones (alto, largo, ancho) que nos permiten conocer su área, la altura es común de 2 metros para todos los espacios. Estas características ponen un límite en el número de personas que puede haber en cada habitación, luego de llenarse una sala nadie más podrá entrar. Los espacios que consideraremos para la modelación son:

- Salas de clases
- Salas de reuniones
- Subterráneo

Tanto las **Salas de clases** como las **Salas de reuniones** tienen asientos disponibles, sin embargo el subterráneo no.

### TO DO

El sistema debe permitir lo siguiente:

- Almacenar una lista de espacios: `lista_espacios`.
- Elegidos dos espacios del sistema determina cuál es más grande. (i.e: `lista_espacios[i] > lista_espacios[j]`). Si ambos espacios poseen asientos se comparará la cantidad de estos, en cambio si al menos uno NO posee asientos se comparará por área.
- Unir dos espacios del mismo tipo para generar uno más grande. (i.e: `lista_espacios[i] + lista_espacios[j]`). El resultado será una sala con la suma tanto de sus asientos como de sus capacidades y de sus dimensiones (el alto no varía), imprimir características del nuevo espacio. Los espacios originales siguen existiendo y los subterráneos NO son unibles.

El nuevo sistema a desarrollar permite también conocer qué personas están dentro de cada espacio (Alumno/Profesor), identificándolos con nombre. Ambos tipos de personas pueden entrar a un espacio y usar un cupo de este:

- Cuando un alumno entra a un espacio con asientos este usará un asiento (si no hay asiento disponible no puede entrar).

- Cuando un profesor entra a una Sala de clases NO usa asiento.
- Cuando un profesor entra a una Sala de reuniones usa asiento (si no hay asiento disponible no puede entrar).
- El sistema debe ser poblado con:

Espacio	Ancho	Largo	Max Personas	Num Asientos
s_clases_1	20	25	6	5
s_clases_2	10	10	6	3
s_reuniones	15	30	10	10
subterraneo	20	20	40	-

- Además, crear 5 alumnos y 3 profesores. Hacerlos entrar todos a s\_clases\_2, imprimir nombre de la persona que entra y mostrar el estado de la sala con el ingreso de cada persona (estado: “asientos ocupados/asientos disponibles”, “personas dentro/máximo personas”)

## Prueba del Sistema

Para poblar automáticamente el sistema importa tu respuesta en AC3\_main.py que se encuentra en la carpeta de esta actividad en github y ejecuta.

## Formato Entrega

El archivo que debes subir a tu repositorio privado debe ser: AC3\_numAlumno1\_numAlumno2.py

Es suficiente con que uno de la pareja lo suba.

## Tips

- `Clase.__bases__` es la lista de las clases padres de Clase.
- `type(obj)` es el tipo del objeto obj.

## Puntaje

- (1.00 pts.) Definición de las clases necesarias para modelar el problema
- (1.00 pts.) Correcto uso de herencia
- (1.50 pts.) Correcto uso de polimorfismo: Comparación correcta de espacios
- (1.00 pts.) Correcta unión dos espacios
- (0.50 pts.) Correcto uso de properties
- (0.75 pts.) Entrada correcta de profesores a espacio
- (0.75 pts.) Entrada correcta de alumnos a espacio
- (0.50 pts.) Imprimir un output que indique quién y con qué cargo entró a una sala y el estado actual de la sala en cuanto a asientos y capacidad disponible.