



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

IIC2233 - Programación Avanzada  
1 semestre 2015

# Actividad 11

## Abstract Classes

### Figuras geométricas

Una empresa de software está construyendo un programa que opera con figuras geométricas, y necesita ayuda con el modelamiento inicial. Para la empresa es crucial que el modelamiento sea lo más extensible posible.

### Requerimientos

- **Toda Figura tiene:**

- Una propiedad **center** como un par ordenado  $(x,y)$ . Se debe poder obtener y setear. El constructor también debe pedir un centro.
- Un método **trasladar** que, dado un par ordenado, sume cada componente a cada componente del centro actual (i.e. una traslación del centro).
- Una propiedad **perimetro**. Se debe poder obtener a partir de las dimensiones de cada figura.
- Una propiedad **area**. Se debe poder obtener a partir de las dimensiones de cada figura.
- Un método **crecer\_area** que permita aumentar el área de la figura en  $x$  veces, aumentando sus dimensiones en forma proporcional. Por ejemplo, si la figura es un rectángulo se debe modificar su **ancho** y **largo**.
- Un método **crecer\_perimetro** que permita aumentar el perímetro en  $x$  unidades, aumentando sus dimensiones en forma proporcional. Por ejemplo, si la figura es un rectángulo se debe modificar su **ancho** y **largo**.
- Cada vez que se imprima cualquier figura, se debe imprimir con el siguiente formato:  
`NombreClase - Perimetro: valor, Area: valor, Centro: (x, y)`

- **Un Rectángulo tiene:**

- Dos propiedades, **largo** y **ancho**, que se deben poder obtener y setear. El constructor debe pedir ambas.

- **Un Triángulo Equilátero tiene:**

- Una propiedad **lado** que se debe poder obtener y setear. El constructor debe pedir un **lado**.

## To-Do

- Implemente las clases necesarias para la modelación del problema, tomando en cuenta que algunas de las clases y métodos pueden ser abstractos.
- **(Bonus: 2.0 pts.)** Implemente para toda figura la propiedad `vertices`, como una lista de pares `(x, y)` que representen los vértices de la figura en el plano. Implemente esto usando `map` y generadores.

## Ayuda

- No olvide que para cambiar lo que se muestra al hacer `print(objeto)` debe sobrescribir el método `__repr__`