

# Programación OO con Java

- Sobre carga de constructores

```
- public Rectangulo() {  
-     x1 = 1;  
-     x2 = 2;  
-     y1 = 1;  
-     y2 = 2;  
- }  
- public Rectangulo(float x1, float y1) {  
-     this.x1 = x1;  
-     x2      = 2;  
-     this.y1 = y1;  
-     y2      = 2;  
- }
```

# Programación OO con Java

- Una clase implementa métodos
  - `public float calcularArea() {`
  - `return (x2-x1) * (y2-y1);`
  - `}`
- Un método puede ser sobrecargado
  - `public float calcularArea(int`  
`modificador) {`
  - `return (x2-x1) * (y2-y1) *`  
`modificador;`
  - `}`

# Programación OO con Java

- Paquete (Package)
  - Los paquetes son grupos relacionados de clases e interfaces y proporcionan un mecanismo conveniente para manejar un gran juego de clases e interfaces y evitar los conflictos de nombres. Además de los paquetes de Java puede crear sus propios paquetes y poner en ellos definiciones de clases y de interfaces utilizando la sentencia package.

# Programación OO con Java

- El paquete de lenguaje Java, también conocido como `java.lang`, contiene las clases que son el corazón del lenguaje Java.
- Para utilizar un paquete, se utiliza la sentencia **`import nombrePackage;`**
- Esta sentencia debe estar al comienzo del archivo.

# Programación OO con Java

- Control de accesos a atributos y metodos
  - public
    - Cualquier clase puede acceder a las propiedades y métodos públicos.
  - protected
    - Sólo las clases heredades y aquellas situadas en el mismo paquete pueden acceder a las propiedades y métodos protegidos.
  - private
    - Las variables y métodos privados sólo pueden ser accedidos desde dentro de la clase.

# Programación OO con Java

- private protected
  - Sólo las clases heredadas pueden acceder a las propiedades y métodos protegidos.

# Programación OO con Java

- Clases Public, Abstract, y Final
  - Public:
    - El modificador public declara que la clase puede ser utilizada por objetos que estén fuera del paquete actual. Por defecto, una clase sólo puede ser utilizada por otras clases del mismo paquete en el que están declaradas.
  - Abstract:
    - El modificador abstract declara que la clase es una clase abstracta. Una clase abstracta podría contener métodos abstractos (métodos sin implementación). Una clase abstracta está diseñada para ser una superclase.

# Programación OO con Java

- Final:

- Utilizando el modificador final se puede declarar que una clase es final, que no puede tener subclases.

- Existen (al menos) dos razones por las que se podría querer hacer esto:

- razones de seguridad.
    - razones de diseño.



# Programación OO con Java

- Implementación de clases abstractas
  - abstract class Geometria{
    - El cuerpo de la clase
    - Variables y métodos, incluyendo tipo de retorno;
  - }
- Implementación de subclases
  - class Circulo extends Geometria {
    - El cuerpo de la clase
    - implementación de métodos superiores ( pisarlas )
    - implementación de otros métodos
  - }

# Programación OO con Java

- Ejercicios:
  - Implementar la Clase Geometria ( abstracta )
  - La clase debe tener los siguientes métodos:
    - calculaArea();
    - calculaPerimetro();
  - Implementar las clases Rectangulo, Circulo y Triangulo que extienden la clase Geometria.
  - Haga un programa que reciba como argumentos el tipo de la geometría y las dimensiones.
  - ej: java calculo R 10 5
    - resultado: Rectángulo, área = 50 y perímetro = 30
  - ver: dir Abstractas