

# Clases y Objetos

UACM SLT

- Una clase es una colección de datos, y métodos que operan sobre esos datos. Los datos y los métodos en conjunto sirven para definir algún tipo de objeto. Por ejemplo la clase Círculo:

<b>clase Círculo</b>
double x double y double r
double circunferencia()  double area()

```
public class Circulo
{
    // Variables de Instancia

    public double x,y;
    public double r;

    // Métodos de Instancia

    public double Circunferencia()
    {
        return 2*3.14159*r;
    }

    public double area()
    {
        return 3.14159*r*r;
    }
}
```

# Instancias de clase

- Al definir la clase `Círculo` en JAVA se ha creado un nuevo tipo de dato, ya que variables de ese tipo pueden ser declaradas como:
  - `Circulo c1,c2,c3,c4;`
- Pero estas variables son sólo nombres que hacen referencia a objetos círculo pero no son objetos en sí. En JAVA todo objeto se crea dinámicamente con la palabra `new`

```
Circulo c1,c2,c3,c4;
```

```
c1= new Circulo();
```

```
c2= new Circulo();
```

```
c3= new Circulo();
```

```
c4= new Circulo();
```

o bien:

```
Circulo c1= new Circulo();
```

```
Circulo c2= new Circulo();
```

```
Circulo c3= new Circulo();
```

```
Circulo c4= new Circulo();
```

Se dice entonces que c1, c2, c3 y c4 son variables de tipo **Círculo** que contienen instancias de la clase **Circulo** (es decir, objetos **Círculo**).

# Acceso a los datos del Objeto

- Una vez creado un objeto se pueden utilizar sus campos de datos o variables de instancia de la siguiente forma a manera de ejemplo:

```
Circulo c1= new Circulo();  
  
// Objeto Circulo c1 con coordenadas (2,2) y radio de valor 1  
c1.x=2.0;  
c1.y=2.0;  
c1.r=1.0;
```

# Uso de métodos del Objeto

- Así mismo se pueden utilizar los métodos de instancia del objeto sobre sí mismo (a través de sus datos) de la siguiente forma a manera de ejemplo:

```
Circulo c= new Circulo();  
double a;  
  
c.x=2.0;  
c.y=2.0;  
c.r=2.5;  
  
a=c.area();
```

# El Constructor (Creación de un objeto)

- Toda clase en JAVA tiene por lo menos un método constructor (con el mismo nombre que la clase), cuyo propósito es efectuar toda la inicialización necesaria para el nuevo objeto. En el ejemplo de los círculos:
  - `Circulos c=new Circulos();`
- JAVA proporciona un constructor por omisión que no pide argumentos ni efectúa una inicialización especial. A continuación, un ejemplo de un constructor para la clase Círculo definido por el programador:



```
public class Circulo
{
    // Variables de Instancia

    public double x,y;
    public double r;

    //Método Constructor

    public Circulo(double x, double y, double r)
    {
        this.x=x; //El argumento implícito llama "this"
        this.y=y; // y hace referencia a "este" objeto
        this.r=r; //o variable
    }
}
```

# Creación de Objetos

## ANTES

```
Circulo c;  
c=new Circulo();  
c.x=1.414;  
c.y=-1.0;  
c.r=0.25;
```

## DESPUES

```
Circulo c= new Circulo(1.414,-1.0,0.25)
```

# Ejercicio

- Realizar el main para la clase circulo y ejecutar para obtener el radio y perímetro de los métodos establecidos anteriormente.
- Nota: Para realizar el ejercicio tienen que crear un objeto de la clase circulo para poder acceder a los métodos.