

# Declaración e inicialización de variables y su ámbito

# Declaración de variables

## ➤ Declaración:

tipo identificador;  int mivar;

## ➤ Declaración e inicialización:

int p=10;

## ➤ Declaración múltiple:

int p, s, a=5;

# Identificadores

➤ Se puede utilizar cualquier combinación de letras, números, \$ y \_.

➤ Existen las siguientes restricciones:

- No se pueden utilizar palabras reservadas como identificador (incluido "goto"). A excepción de "var".

- No puede comenzar por carácter numérico

```
int _1=10; //ok  
char break; //error  
int 3aj; //error  
float car.t; //error
```

# Ámbito

## ➤ La variables pueden declararse:

- A nivel de clase compartidas por todos los métodos. Se les conoce como atributos o campos
- En el interior de un método. Se les conoce como locales. Solo visibles dentro de ese método

```
class MiClase{  
    int n; //variable atributo  
    public void metodo(){  
        int c; //variable local  
        int n; //local con mismo nombre que atributo  
        n=10; //acceso a variable local  
        this.n=3; //acceso a variable atributo  
    }  
}
```

# Inicialización por defecto

- **Variables locales:** NO se inicializan por defecto. Es necesario asignarles un valor antes de utilizarlas.

```
public void metodo(){  
    int c;  
    c=c+3; //error de compilación  
}
```

- **Variables atributo:** Se inicializan por defecto:

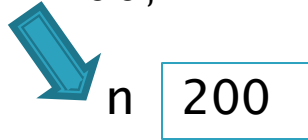
- **Enteras:** 0
- **Decimales:** 0.0
- **boolean:** false
- **char:** '\u0000' (carácter nulo)
- **Objeto:** null

```
class Test{  
    int c;  
    boolean car;  
    public void metodo(){  
        c=c+3; //0+3  
        System.out.println(car); //false  
    }  
}
```

# Variables objeto y de tipos primitivos

➤Tipos primitivos. La variable contiene al dato

```
int n=200;
```



➤Tipos objeto. La variable contiene una referencia al dato

```
Object ob=new Object();
```

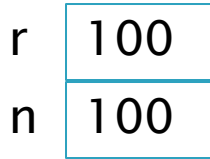


# Diferencias objetos / primitivos

➤ **Tipos primitivos.** En una asignación, cada variable tiene una copia del dato

```
int r = 100;
```

```
int n=r;
```



➤ **Tipos objeto.** Con variables objeto, ambas variables apuntan al mismo objeto

```
Object ob=new Object();  
Object cp=ob;
```

