

- Sintaxis del lenguaje.
- Variables, tipos de datos primitivos, operadores.
- Estructuras de control de flujo.
- Clases
- Herencia



- Nombres de variables: camelCase
- Nombres de clases: Comienzo con Mayúscula
- Línea finaliza con ";"
- Diferencia entre mayúsculas y minúsculas
- Las clases deben llamarse igual que el archivo que las contiene
- Cada clase se escribe en un archivo con el nombre de la Clase

- boolean (1 byte)
- char (2 bytes)
- byte (1 byte)
- short (2 bytes)
- int (4 bytes)
- long (8 bytes)
- float (4 bytes)
- double (8 bytes)

Definición e inicialización

```
int entero;
```

```
long largo = 33;
```

```
byte unByte = 0x55;
```

```
boolean flag = true;
```

```
float f = (float) 3.14;
```

```
double d = 1.73;
```

| | |
|---------------------|----------------------------|
| • + | Suma |
| • - | Resta |
| • * | Multiplicación |
| • / | División |
| • % | Resto |
| • = | Igual |
| • instanceof | Es instancia de... |
| • > | Mayor |
| • < | Menor |
| • != | Distinto |
| • & | And lógico |
| • | Or lógico |
| • ^ | Xor Lógico |
| • ~ | Negación lógica |
| • >> | Desplazamiento a derecha |
| • << | Desplazamiento a izquierda |

```
if (expresionBooleana)
{
    // ...
}
else if (expresionBooleanaII)
{
    // ...
}
else
{
    // ...
}
```

```
switch (numero)
{
    case 0 :
    {
        //...
        break;
    }
    case 1 :
    {
        //...
        break;
    }
    default :
    {
        //...
    }
}
```

```
while (expresionBooleana)
{
    // ...
}
```

```
do
{
    // ...
}while (expresionBooleana);
```

```
for (inicializacion; expresionBooleana; accion)
{
    // ...
}
```

```
//for each
for (Objeto o : arrayDeObjetos)
{
    // ...
}
```

```
public class Persona
{
    public int a;
    private int b;

    public void metodo1 ()
    {

    }

    private void metodo2 ()
    {

    }
}
```

```
Persona juan = new Persona();
juan.a = 13;
juan.metodo1();
```

```
public class Persona
{
    private int a;

    public Persona (int a)
    {
        this.a =a ;
    }
}
```



```
public class Persona
{
    public int edad;

    public Persona()
    {
    }

    public void usarPc()
    {
        System.out.println("Persona de "+edad+" años usando
                             Open Office");
    }
}

public class Programador extends Persona
{
    public void repararPc()
    {
        System.out.println("Reparando PC");
    }
}
```