

## 2.B. Tipos de datos.

### 1. Los tipos de datos.

#### 1.3. Declaración e inicialización.

Llegados a este punto cabe preguntarnos ¿cómo se crean las variables en un programa? ¿Qué debo hacer antes de usar una variable en mi programa? Pues bien, como podrás imaginar, debemos crear las variables antes de poder utilizarlas en nuestros programas, indicando qué nombre va a tener y qué tipo de información va a almacenar, en definitiva, debemos **declarar la variable**.

Las variables se pueden declarar en cualquier bloque de código, dentro de llaves. Y lo hacemos indicando su identificador y el tipo de dato, separadas por comas si vamos a declarar varias a la vez, por ejemplo:

```
int numAlumnos = 15;
```

```
double radio = 3.14, importe = 102.95;
```

De esta forma, estamos declarando `numAlumnos` como una **variable de tipo int**, y otras dos variables `radio` e `importe` de tipo `double`. Aunque no es obligatorio, hemos aprovechado la declaración de las variables para **inicializarlas** a los valores 15, 3.14 y 102.95 respectivamente.

Si la variable va a permanecer inalterable a lo largo del programa, la declaramos como **constante**, utilizando la palabra reservada **final** de la siguiente forma:

```
final double PI = 3.1415926536;
```

En ocasiones puede que al declarar una variable no le demos valor, ¿qué crees que ocurre en estos casos? Pues que el **compilador** le **asigna un valor por defecto**, aunque depende del tipo de variable que se trate:

- Las **variables miembro** sí se inicializan automáticamente, si no les damos un valor. Cuando son de tipo numérico, se inicializan por defecto a 0, si son de tipo carácter, se inicializan al carácter **null** (\0), si son de tipo boolean se les asigna el valor por defecto **false**, y si son tipo referenciado se inicializan a **null**.
- Las **variables locales** no se inicializan automáticamente. Debemos asignarles nosotros un valor antes de ser usadas, ya que si el compilador detecta que la variable se usa antes de que se le asigne un valor, **produce un error**. Por ejemplo en este caso:

```
int p;
```

```
int q = p; // error
```

Y también en este otro, ya que se intenta usar una variable local que podría no haberse inicializado:

```
int p;
```

```
if ( . . . )
```

```
    p = 5 ;
```

```
int q = p; // error
```

En el ejemplo anterior la instrucción **if** hace que si se cumple la condición que hay entre paréntesis (cualquiera que indiquemos), entonces el

programa asignará el valor 5 a la variable `p`; sino se cumple la condición, `p` quedará sin inicializar. Pero si `p` no se ha inicializado, no tendría valor para asignárselo a `q`. Por ello, el compilador detecta ese posible problema y produce un error del tipo **"La variable podría no haber sido inicializada"**, independientemente de si se cumple o no la condición del `if`.

### Autoevaluación

De las siguientes, señala cuál es la afirmación correcta:

- ☐ La declaración de una variable consiste básicamente en indicar el tipo que va a tener seguido del nombre y su valor.
- ☐ Java no tiene restricción de tipos.
- ☐ Todos los tipos tienen las mismas operaciones a realizar con ellos: suma, resta, multiplicación, etc.
- ☒ Todas las anteriores son incorrectas.