

# Referencias en Java

Estructuras de Datos

Grado en Ingeniería Informática, del Software y Computadores

Universidad de Málaga

# Tipos en Java

Java soporta tipos:

- primitivos (**int**, **double**, **char**, **boolean**, etc.)
- referencia (**String**, **Random**, **Scanner**, etc.)

Los datos de tipos primitivos y referencia:

- residen en zonas diferentes de la memoria
- se gestionan de diferente manera

# Tipos primitivos

- La declaración reserva memoria para el **valor** de la variable

```
int x = 5;
```

```
int y = 7;
```

```
int z = 3;
```

Variable	Valor
x	5
y	7
z	3

Estado de la memoria tras las declaraciones

# Tipos referencia

- La declaración reserva memoria para la **referencia**, no para el valor
- Las referencias se inicializan por defecto a **null**

**Random r;**

**Point p;**

**LinkedList<Char> l;**

Variable	Referencia
r	null
p	null
l	null

Estado de la memoria tras las declaraciones

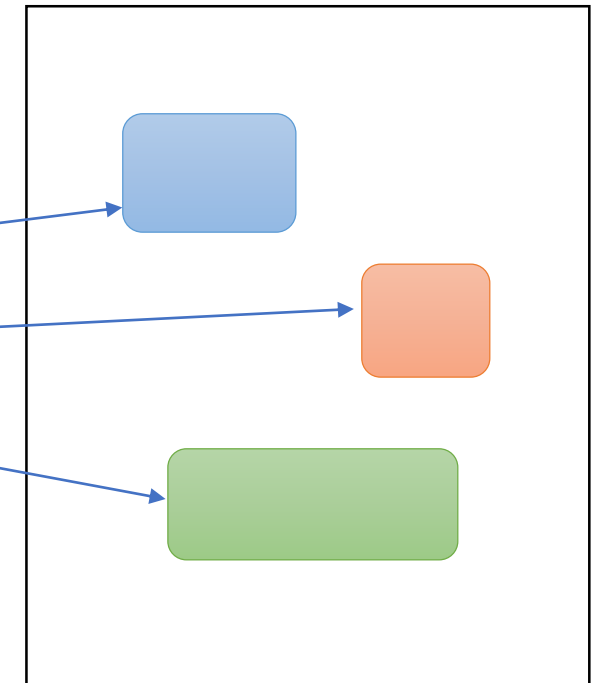
# Creación de objetos para referencias

- El objeto al que se refiere una referencia se crea con **new**
- Este objeto reside en otra zona de memoria llamada **montón** (*heap*)

```
r = new Random();  
p = new Point(7,3);  
l = new LinkedList<>();
```

Variable	Referencia
r	
p	
l	

Estado de la memoria tras la creación



Estado del montón tras la creación

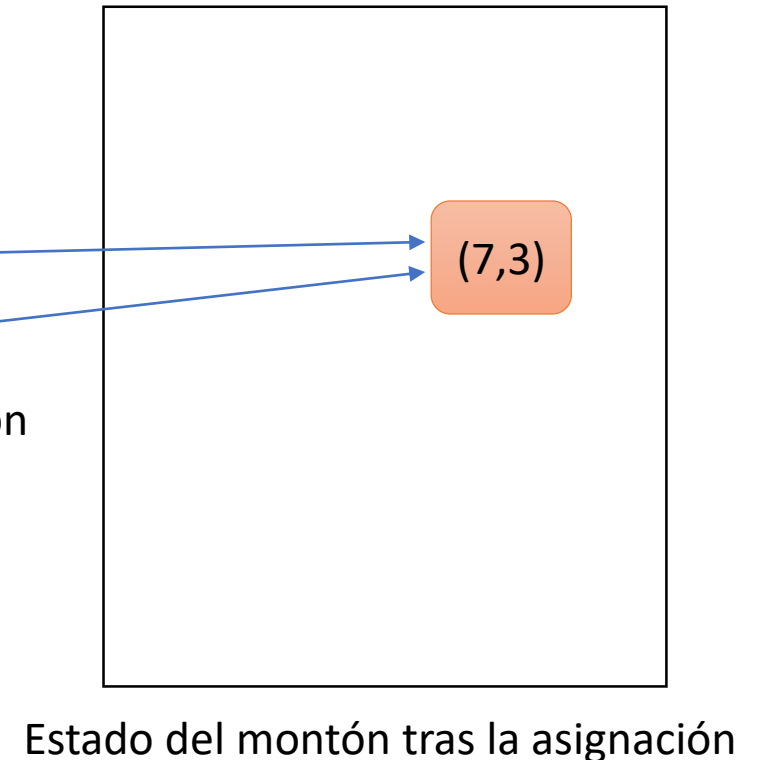
# Alias

- Una referencia puede referirse a un objeto que ya existía
- Ambas referencias son **alias**, se refieren al **mismo** objeto

```
p = new Point(7,3);  
Point q = p;
```

Variable	Referencia
p	
q	

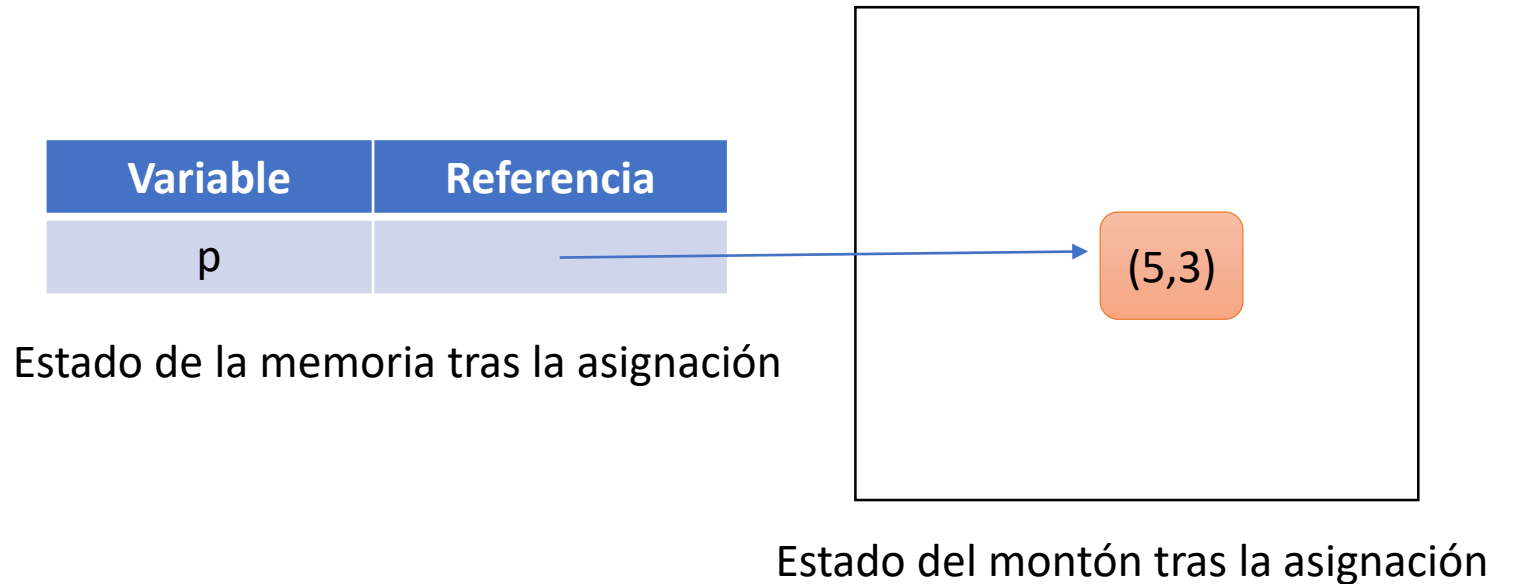
Estado de la memoria tras la asignación



# Acceso a objetos a través de referencias

- La referencia permite acceder a los atributos del objeto referenciado

```
p = new Point(7,3);  
p.x = 5;
```



La referencia **null** no se refiere a un objeto

- La referencia **null** no se refiere a ningún objeto
  - No es un error que una referencia valga **null**; tiene sus usos
  - El **error** es intentar acceder al objeto referenciado por **null**

## Point q;

```
q.x = 7; // error
```

Variable	Referencia
q	null

## Estado de la memoria tras la asignación

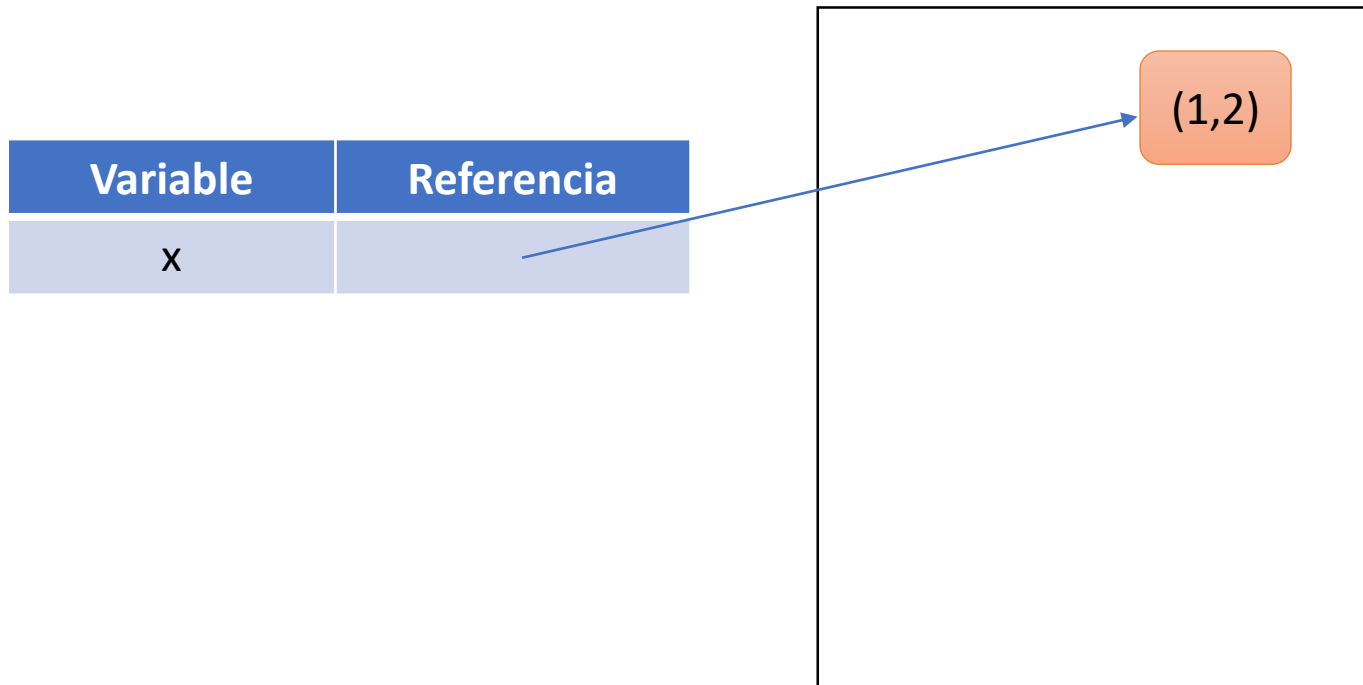
Asegúrate **siempre** de que la referencia no es **null** antes de acceder



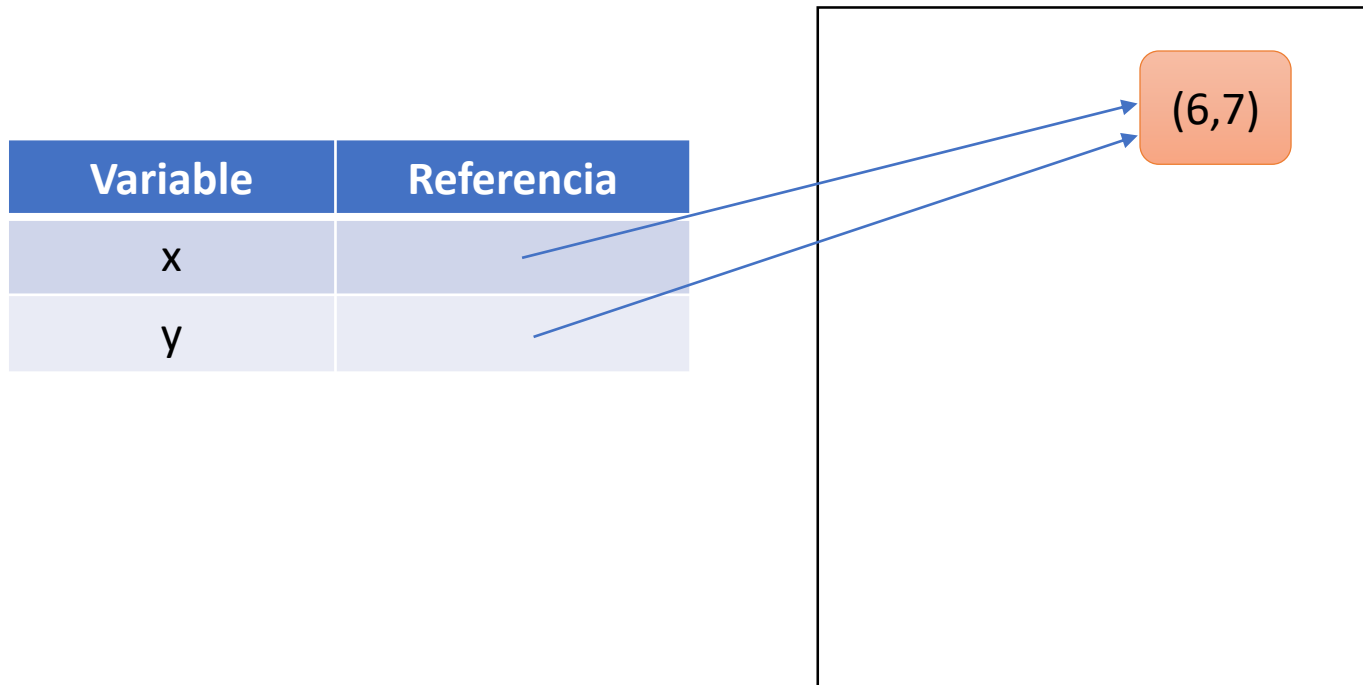


Ejecución de PointDemo.java

# Estado de la memoria y del montón en A

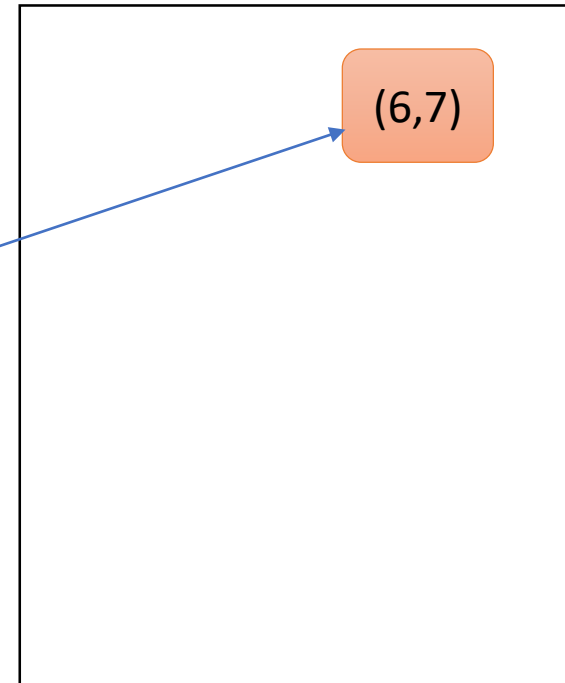


# Estado de la memoria y del montón en B



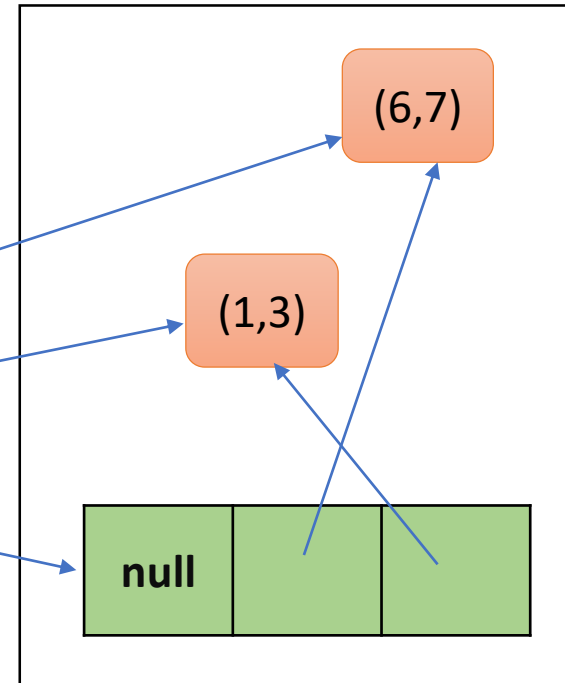
# Estado de la memoria y del montón en C

Variable	Referencia
x	null
y	



# Estado de la memoria y del montón en D

Variable	Referencia
x	null
y	
z	
points	



# Estado de la memoria y del montón en E

Variable	Referencia
x	null
y	
z	
points	

