

La memoria e gli oggetti

Cosa succede quando si fa `p=q` oppure `p==q`?

Come si copiano/confrontano gli oggetti?

Come vengono passati gli oggetti ai metodi?

Variabili

```
int x;  
Point p;
```

Per ogni variabile esiste una casellina (spazio di memoria)

x

p

Non c'è nessuna differenza! (finora)

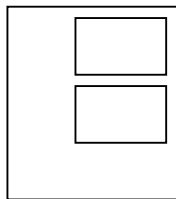
Creazione di oggetti

La espressione `new Point()` crea un oggetto e restituisce l'indirizzo di memoria dell'oggetto.

Effetto di `new Point()`:

x

p

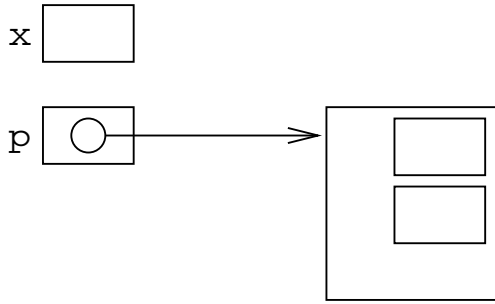


`new Point()` è un metodo;
il valore restituito è la posizione dell'oggetto creato in memoria.

Oggetti e variabili

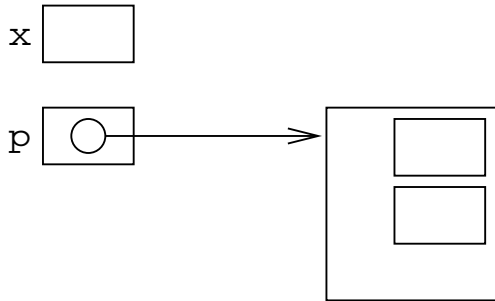
```
p=new Point();
```

Significa: l'indirizzo di memoria in cui si trova l'oggetto creato va in p



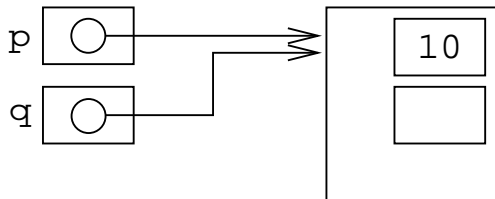
Assegnare oggetti

Quando faccio `p=`, l'indirizzo dell'oggetto va in `p`



Copiare un oggetto

Se `p` e `q` sono variabili `Point`, facendo `q=p` ottengo:



Non ho fatto nessuna copia.

Ho due variabili che indicano lo stesso oggetto.

Come copiare gli oggetti

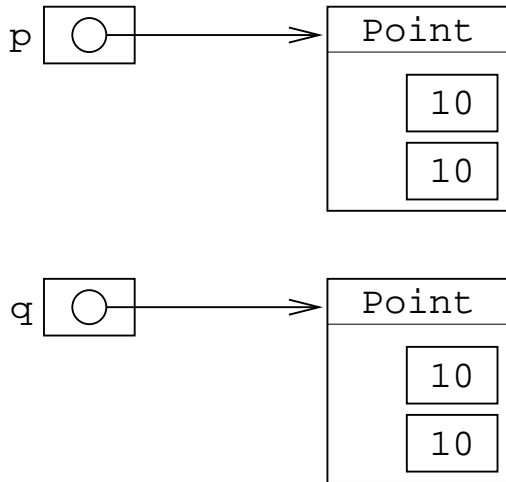
1. creare un nuovo oggetto e copiare una per una le componenti:

```
q=new Point();
q.x=p.x;
q.y=p.y;
```

2. usare il metodo `clone`:

```
q=(Point) p.clone();
```

Il risultato è lo stesso:



Confronto fra variabili

`if (a==b)` ... confronta i valori delle variabili a e b:

a e b sono scalari (es. interi)

viene confrontato il loro valore

a e b sono variabili oggetto (es. `Point`)

viene confrontato il loro valore

Nel secondo caso, i valori delle variabili sono gli indirizzi dei due oggetti.

`a==b` è vero solo se a e b contengono l'indirizzo dello stesso oggetto.

Confronto fra variabili

```
int x, y;
Point p, q;

x=10;
y=10;

p=new Point(10, 10);
q=new Point(10, 10);

if(p==q)
    System.out.println("Sono uguali");
else
    System.out.println("Non sono uguali");

if(x==y)
    ...
```

`x==y`

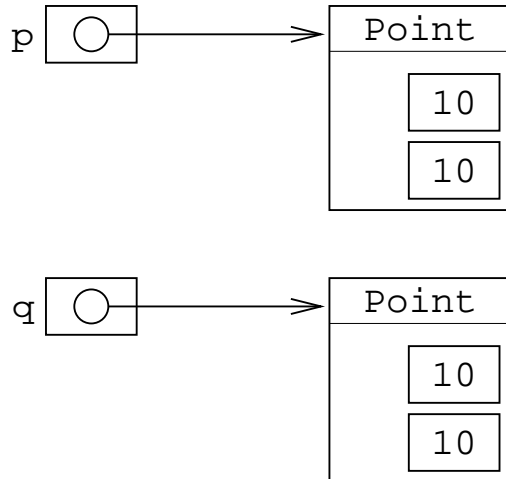
è vera se le due variabili contengono lo stesso valore

`p==q`

è vera se le due variabili contengono lo stesso valore

In memoria

Dopo aver creato i due oggetti:



Ogni volta che faccio `new`, ho un nuovo oggetto, in una nuova posizione.

Sono due oggetti diversi, per cui le loro posizioni in memoria sono diverse!

Sono diversi.

Confrontare oggetti

Confronto fra `p` e `q`:

voglio sapere se sono lo stesso oggetto:

uso `==`

voglio sapere se contengono gli stessi valori:

1. confronto una per una le componenti:

```
if((p.x==q.x)&&(p.y==q.y))  
...
```

2. uso il metodo `equals`:

```
if(p.equals(q))  
...
```

clone e equals

Sono metodi definiti per tutti gli oggetti.

Per le classi predefinite di Java (come `Point`) si possono usare tranquillamente.

Per le classi definite da noi: vanno ridefiniti (vedremo poi come farlo).