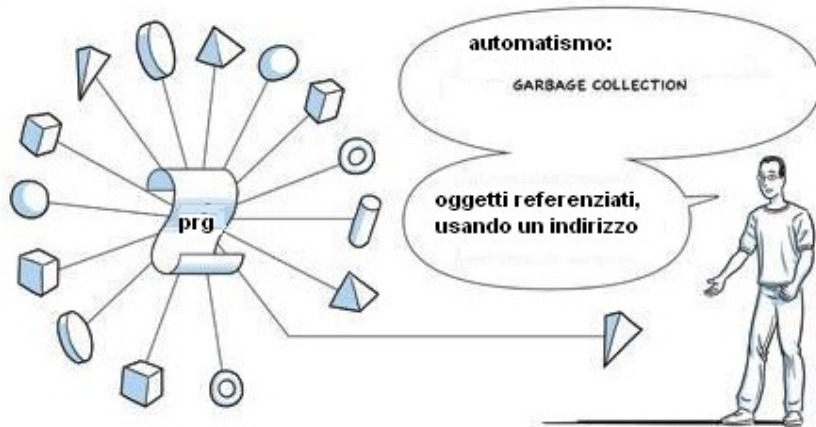


GARBAGE COLLECTION

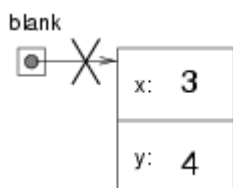


Quando **non** esistono riferimenti ad un oggetto il meccanismo automatico di *garbage collection* individua e libera tale area di memoria *inaccessibile*

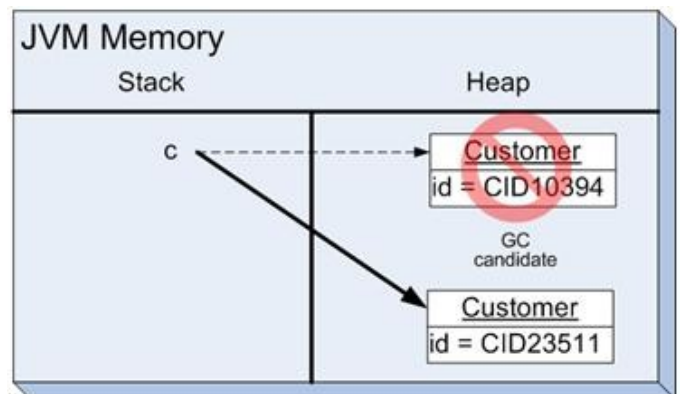
Esempi che creano oggetti *candidati* al *garbage collection*:

```
Customer c = new Customer(CID10394);
Customer c = new Customer(CID23511);
```

```
Point blank = new Point (3, 4);
blank = null;
```



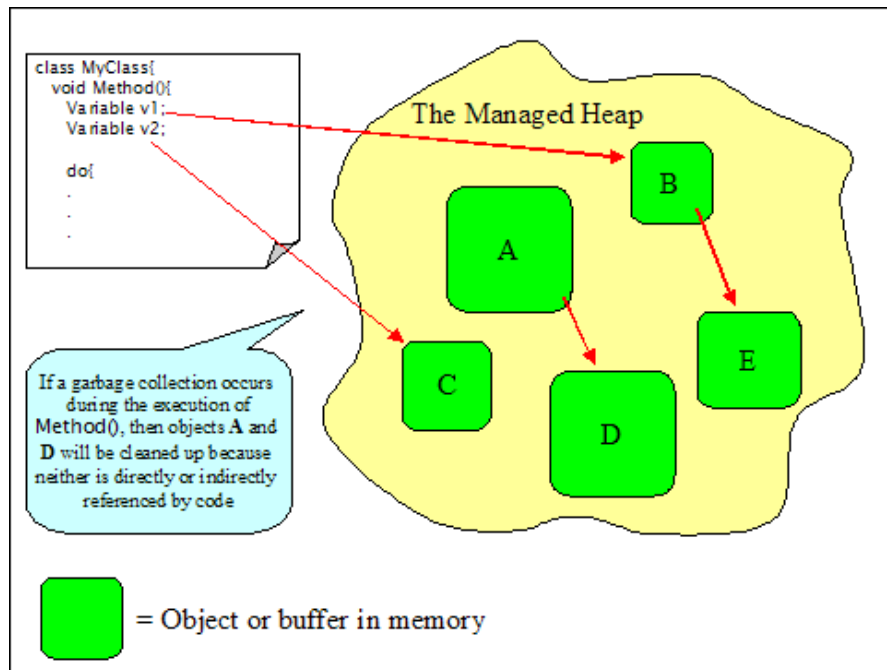
oggetto candidato
al GC: blank infatti punta all'oggetto **null**



Si pensi ad un'applicazione, parzialmente definita in figura, dove l'oggetto A non è più referenziato perché si è usciti dallo *scope* (*ambito di visibilità* della variabile che conteneva tale riferimento) o in conseguenza, ad esempio, dell'uso di un **metodo conclusivo** con sintassi:

`nomeOgg.close();`

In tal caso l'oggetto A ed in conseguenza anche D (che risulta indirettamente non più referenziato) sono **marcati** come aree di memoria *inaccessibili*.



Se accade che l'automatismo del **garbage collection** si attivi durante l'esecuzione del metodo `Method()` - all'interno del quale sono referenziati due oggetti della classe `Variable` - gli oggetti **marcati** come aree di memoria *inaccessibili* vengono eliminati, liberando spazio in memoria.

