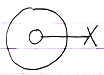
|  |
| --- |
| Circle Language Spec: Black Boxes |

## Notations of Private

Private access connectors get a different notation in different situations.

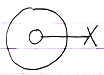
#### Private in a Definition

In a definition the private access connector is a cross:



This is because access is restricted.

Private Set in definition:



Private Get in definition:



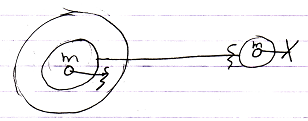
#### Private in a Friend Reference

Inside a friend reference, a Private member becomes accessible, so the access connector it is displayed using a wavy line:



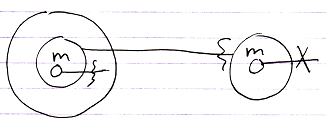
Which expresses *accessible*, but only to a select group of objects.

The following diagram expresses a member with Private Set accessible from within the befriended:



The circle on the right is the definition, so the its member is displayed with a private access connector there (a cross). The access connector within the Friend is different. It is drawn with a wavy line, expressing ‘access conditionally’. In that situation the Private access connector is also called a Friend access connector.

The following diagram expresses a member with Private *Get* accessible from within the befriended:

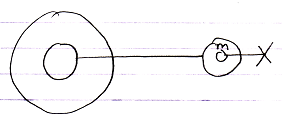


You can differentiate the Friend *access connector* from Friend *declarations* because an access connector has nothing at the end of its line.

#### Private in a Normal Reference

In normal references, private members are *not shown at all*.

The following diagram shows a member with Private Set not accessible or visible from in a normal object:



And that is where the complexity *hiding* kicks in. The circle on the right is still the definition, so it does show the Private member while the normal reference on the left does not show the Private members of its related object.

The following diagram shows a member with Private *Get* not accessible or visible from within a normal reference:

