

## Inheritance

TODO(This is a stub)

### Overriding

A callable declaration (that is, a property or member function declaration) inside a classifier declaration is said to be *overridable* if:

- Its visibility (and the visibility of its getter and/or setter) is not **private**;
- It is declared as **open**, **abstract** or **override** (interface methods and properties are implicitly **abstract**).

A callable declaration inside a classifier declaration *subsumes* a corresponding declaration of the base classifier type if:

- Its return type is a subtype of the return type of the corresponding declaration;
- Its formal parameter types are supertypes of the types of corresponding parameters from the corresponding declaration;
- Its name is the same as the name of the corresponding declaration.

TODO(visibility games)

If the declaration of the base classifier type is overridable and the declaration of the derived classifier type subsumes it and it has an **override** modifier, it is **overriding** the base declaration.

If the base declaration is not overridable and/or the deriving declaration does not have an **override** modifier, it is not permitted and should result in a compile-time error.

Note: Kotlin does not have a concept of full shadowing of declarations.

Platforms may introduce additional cases of both *overridability* and *subsumption* of declarations, as well as limit the overriding mechanism due to internal representation limitations.

TODO(...)