Accessibility Levels (C# Reference)

Article • 09/29/2022 • 2 minutes to read

Use the access modifiers, `public`, `protected`, `internal`, or `private`, to specify one of the following declared accessibility levels for members.

Declared accessibility	Meaning
public	Access is not restricted.
protected	Access is limited to the containing class or types derived from the containing class.
internal	Access is limited to the current assembly.
protected internal	Access is limited to the current assembly or types derived from the containing class.
private	Access is limited to the containing type.
private protected	Access is limited to the containing class or types derived from the containing class within the current assembly.

Only one access modifier is allowed for a member or type, except when you use the `protected internal` or `private protected` combinations.

Access modifiers are not allowed on namespaces. Namespaces have no access restrictions.

Depending on the context in which a member declaration occurs, only certain declared accessibilities are permitted. If no access modifier is specified in a member declaration, a default accessibility is used.

Top-level types, which are not nested in other types, can only have `internal` or `public` accessibility. The default accessibility for these types is `internal`.

Nested types, which are members of other types, can have declared accessibilities as indicated in the following table.

Members of	Default member accessibility	Allowed declared accessibility of the member
`enum`	`public`	None

Members of	Default member accessibility	Allowed declared accessibility of the member

`class`	`private`	`public`
		`protected`
		`internal`
		`private`
		`protected internal`
		`private protected`

`interface`	`public`	`public`
		`protected`
		`internal`
		`private` *
		`protected internal`
		`private protected`

`struct`	`private`	`public`
		`internal`
		`private`

* An `interface` member with `private` accessibility must have a default implementation.

The accessibility of a nested type depends on its accessibility domain, which is determined by both the declared accessibility of the member and the accessibility domain of the immediately containing type. However, the accessibility domain of a nested type cannot exceed that of the containing type.

C# Language Specification

For more information, see the C# Language Specification. The language specification is the definitive source for C# syntax and usage.

See also

- C# Reference
- C# Programming Guide
- C# Keywords
- Access Modifiers
- Accessibility Domain
- Restrictions on Using Accessibility Levels
- Access Modifiers
- public
- private
- protected
- internal