

Nullable types

Nullable types are constructed using the `?` type modifier. For example, `int?` is the nullable form of the predefined type `int`. A nullable type's underlying type must be a non-nullable value type.

The type specified before the `?` modifier in a nullable type is called the **underlying type** of the nullable type. The underlying type of a nullable type can be any non-nullable value type or any type parameter that is constrained to non-nullable value types (that is, any type parameter with a `struct` constraint). The underlying type of a nullable type cannot be a nullable type or a reference type.

A nullable type can represent all values of its underlying type plus an additional null value.

The syntax `T?` is shorthand for `System.Nullable<T>`, and the two forms can be used interchangeably.



Add Class *Nullable* type to C# profile.

```
class class1 {  
}  
int? a = null;  
System.Nullable a = null;  
}
```

Reversed UML model:

