Nullable value types and null checking

Agenda

Nullable value types

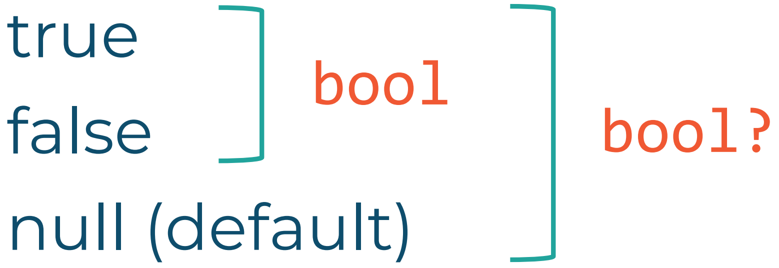
Null checking techniques

Nullable value types

A nullable value type T? represents all values of its underlying value type T and an additional null value

.NET Documentation

Nullable boolean



true

false

null (default)

`bool`

`bool?`

Nullable declaration

```
int? myInteger = null;
bool? myBoolean = true;
DateTime? myDate = DateTime.UtcNow;
```

Shorthand

```
bool? = Nullable<bool>
```

Nullable<T> members

HasValue

Value

GetValueOrDefault()

Conversion

No conversion needed from underlying type to nullable type

Explicit conversion needed for converting nullable type to underlying type

Conversion from T to Nullable<T>

```
bool b1 = true;
bool? b2 = b1;
```

✓

Conversion from Nullable<T> to T

```
bool? b1 = true;
bool b2 = b1;
bool b2 = (bool)b1;
```

⊗

Demo

Print out non nullable properties

Print out nullable properties

Null checking techniques and operators

Null checking operators

Conditional (not specific to null checking)

Coalescing

Null conditional

Conditional operator

```
bool? condition = true;
int myInteger = condition != null ? 10 : 20;
```

Coalescing operator

```
int? i1 = null;
int i2 = i1 ?? -1;
```

Null conditional operator

```
List<Movie> movieList = null;
var title = movieList?[1]?.Title;
```

Demo

Implement null checking

Summary

It is possible to declare nullable value type

The conditional operator allow to check for null values

The coalescing and null conditional operators are useful with variables that can be null