Null Values

Simon Robinson
Software Developer

@TechieSimon www.SimonRobinson.com

The Two Types of Data

Not-nullable
(Can never be null)

Nullable
(Might be null)

Being explicit about this
helps code robustness

Overview

Null values

- How to use nullable and not-nullable types
  - Different techniques for value and reference types
- Null vs. empty for strings
- Enable nullable reference types on an existing project

Demo

Making structs nullable

- Delivery times app
- Will feature nullable and not-nullable structs

Nullability for Reference Types

Demo

Delivery times app – but with classes

- Will show similar nullability syntax
- But differences in techniques required for classes

The following demo presumes nullable reference types are enabled

Structs and Classes

Nullable structs

`MyStruct?` and `MyStruct` are different types

(`MyStruct?`, under the hood becomes `System.Nullable<MyStruct>`)

Nullable classes

`MyClass?` and `MyClass` are the same type

All classes are intrinsically nullable under the hood

Declaring `MyClass` tells the compiler you don't want to put `null` in this instance...

... and the compiler should warn you if it thinks your code might store `null` in it

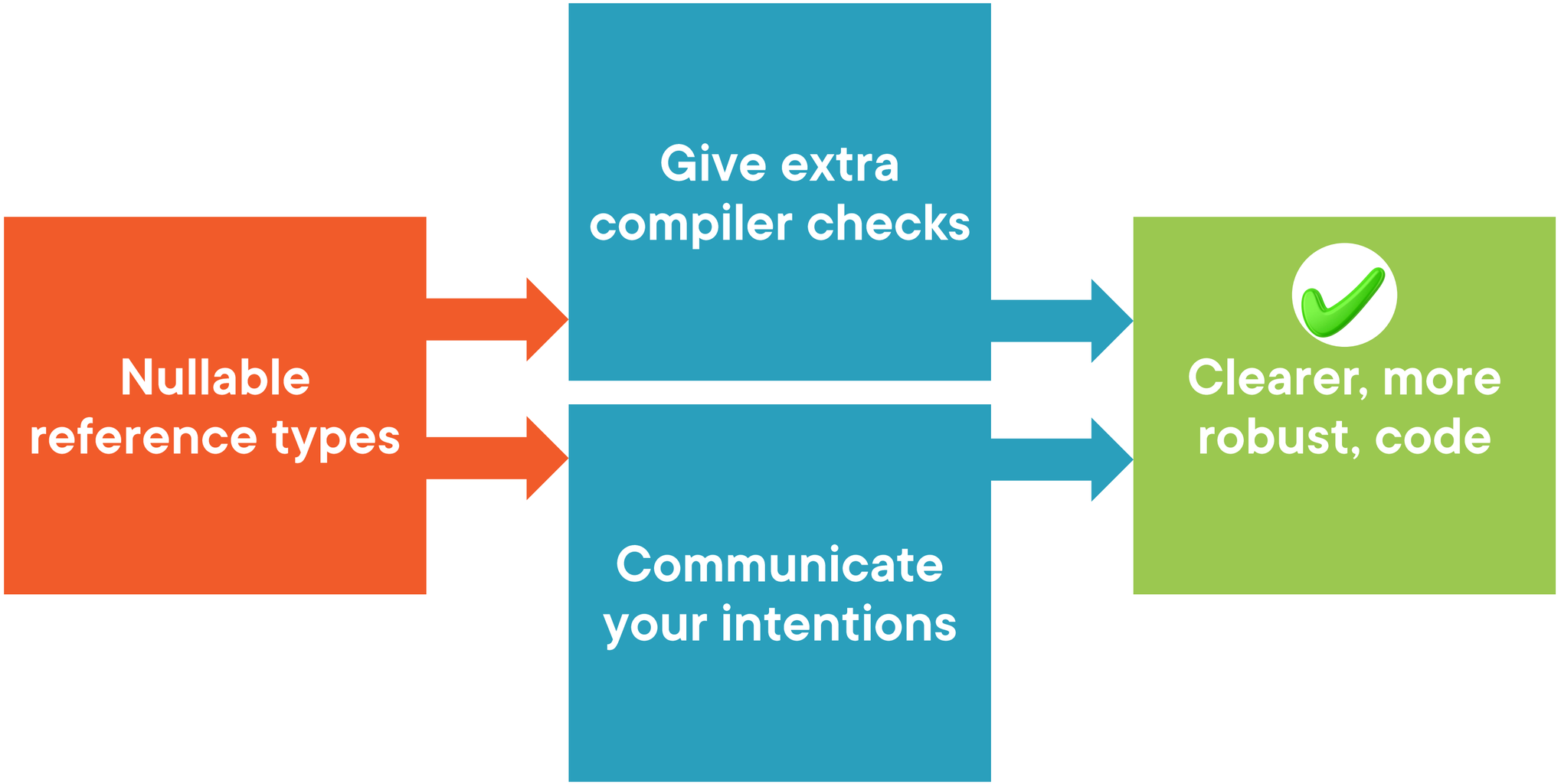
Nulls and Classes: If Null-Checking is Not Enabled

Demo

Delivery times project (again)

- This time, without nullable reference types enabled

Enabling Nullable Reference Types



Null and Empty Strings

Demo

New demo

- Features a string that might lack a usable value
- Shows nullability is more complex for strings

String with No Data

```
string? value = null;
```

```
string value = "";
```

```
string value = "    ";
```

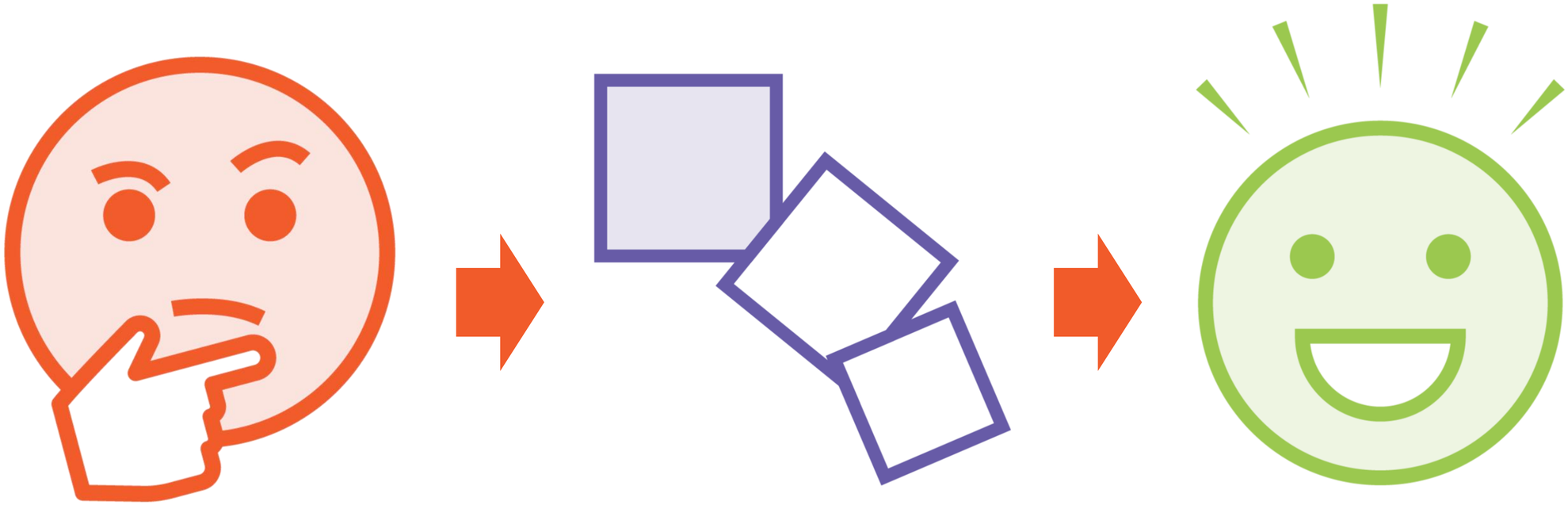
May need to prepare for all these cases

What significance do these values have?

Whitespace might or might not count as data – it's situation-dependant

Example: Preserve whitespace in an XML file

Strings



Think what null/whitespace means

Transform to single form if appropriate

Get fewer bugs

Enabling Nullable Reference Types

Demo

Small library app

- Library of utility functions
- Written without nullable reference type checking
- Will enable nullable reference types
- App has no comments to clarify author's intentions (!)

Summary

Null values

- Use ? to declare an instance as nullable
- For structs, that gives a different type
  - `Value` property to get value
- For classes, depends on enabling nullable reference types
  - Same type, just compiler heuristic checking

Strings

- Think about empty and whitespace strings as well as null

Enable nullable reference types

- Decide which instances are nullable