Implementing Interfaces in Different Modules

Richard Warburton

@richardwarburto www.monotonic.co.uk

Module Outline

Correctly Implementing an Interface

Liskov Substitution Principle

Hiding Implementation

Tips and tricks to hide implementation

Concrete Example

Implementing the repository pattern

Liskov Substition Principle

References to interfaces must be able to use any implementation of an implementing class

```
interface Rectangle {
  void setWidth(int width);
  void setHeight(int height);
  …
}
```

Example Rectangle

```
class Square implements Rectangle {
…
}
```

Liskov Substitution Principle Breakage

A square is a rectangle in natural language

Doesn’t implement an is-a relationship – can’t set width and height independently

General Principles

Preconditions

An implementation should work wherever its parent does

Postconditions

An implementation should do whatever is expected of its parent

Hide Implementation

Access Control

Any methods not on the interface can be private/package scoped

Packages

Use packages to hide implementation: package scoping

Fields

Interfaces don’t expose instance fields

Demo

Completing the Repository

SQL Implementation

- Use a relational database

CSV Implementation

- Flat file storage of comma separated values

Summary

You've learned what it takes to successfully implement an interface!

Hide implementation details

- Access control
- Package scoping

Can you over-abstract though?