Design Patterns in Swift: Creational

INTRODUCTION & PREREQUISITES

Karoly Nyisztor
DEVELOPER
@knyisztor www.leakka.com

Introduction

History

Values and Limitations

Prerequisites

UML Primer

Design Patterns Overview

Creational Design Patterns

Singleton

Prototype

Factory Method

Builder

Abstract Factory

History of Design Patterns

History/Background

Recurring Problems

No Standard Solutions

“Re-Inventing the Wheel”

The Gang of Four

Erich Gamma	**Ralph Johnson**
Richard Helm	**John Vlissides**

Design Patterns

Result of a long evolution process

Proven solutions to recurring problems

Address common software design questions

Design Patterns - Values and Limitations

Benefits of Design Patterns

Reusable

Time Savers

Satisfaction

Future Proof

Less Refactoring

Fewer Bugs

Risks and Limitations

Tough Decisions

Expertise Required

Risk of Delays

Prerequisites

Required Hardware and Software

Mac / OS X El Capitan or later

XCode 8 or later

StarUML 2

UML Primer

Overview

Class Diagrams

UML Relations

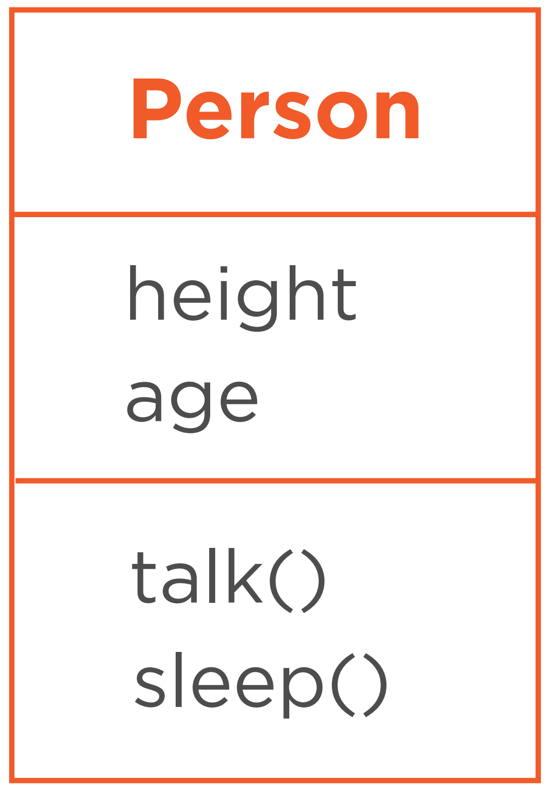
Sequence Diagrams

Class Diagrams

Class Name

Attributes

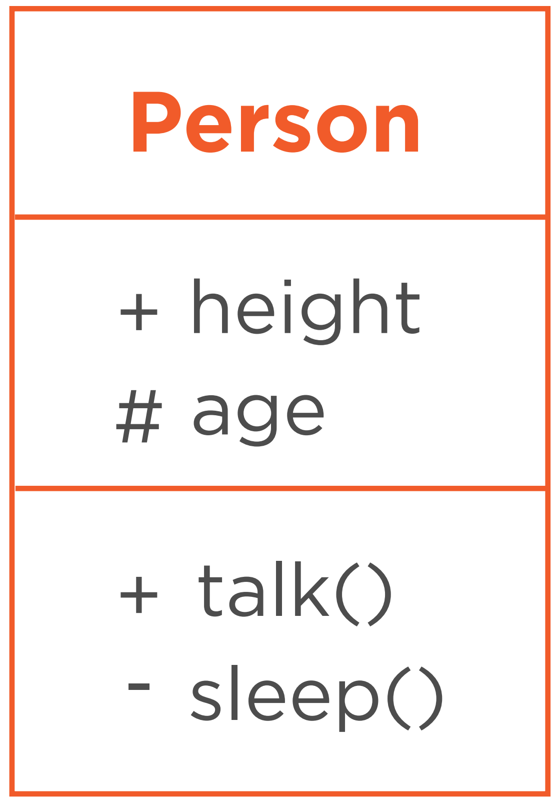
Operations



Public Visibility (+)

Protected Access (#)

Private (-)



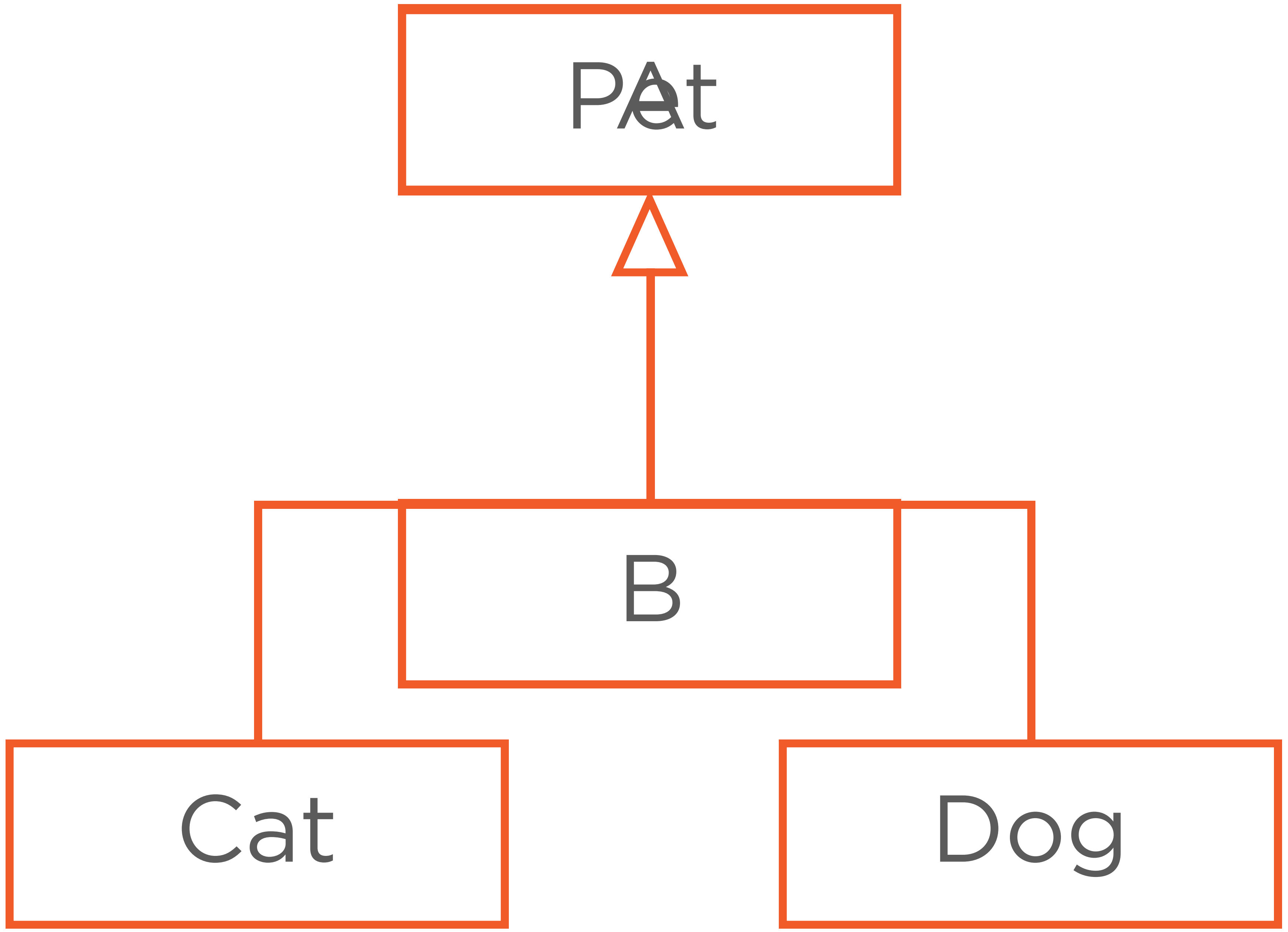
Class diagrams
show the static relationships
between the objects that are
forming the system.

UML Relations

Generalization

“B is an A”

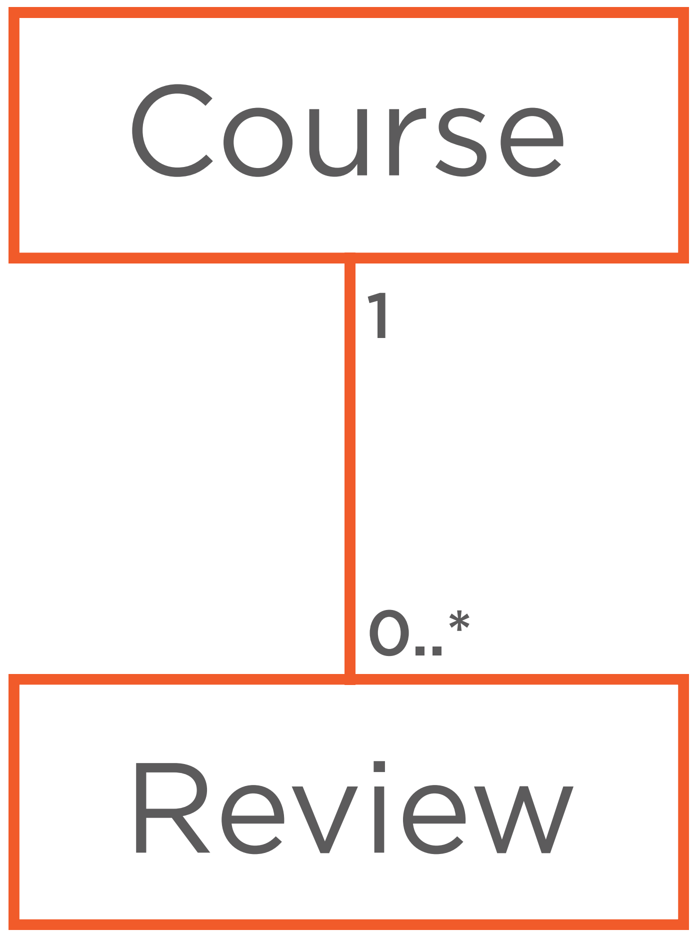
One or More Children



Association

Reference

Multiplicity

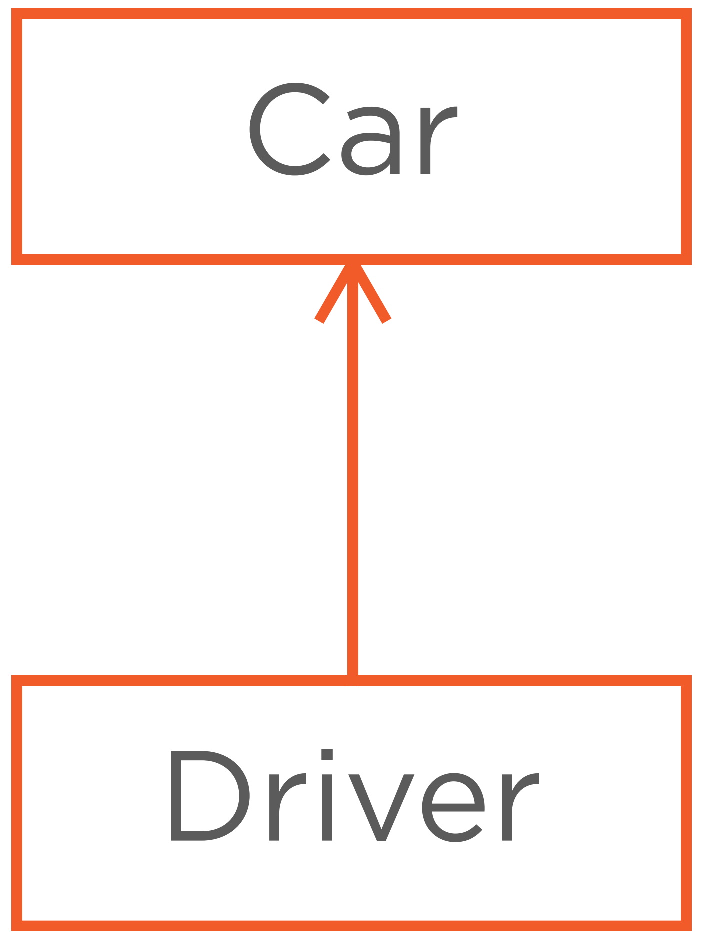


Multiplicity

0	- no instances
0..1	- zero or exactly 1 instance
1	- exactly one instance
0..*	- zero or more instances
*****	- zero or more instances
1..*	- one or more instances

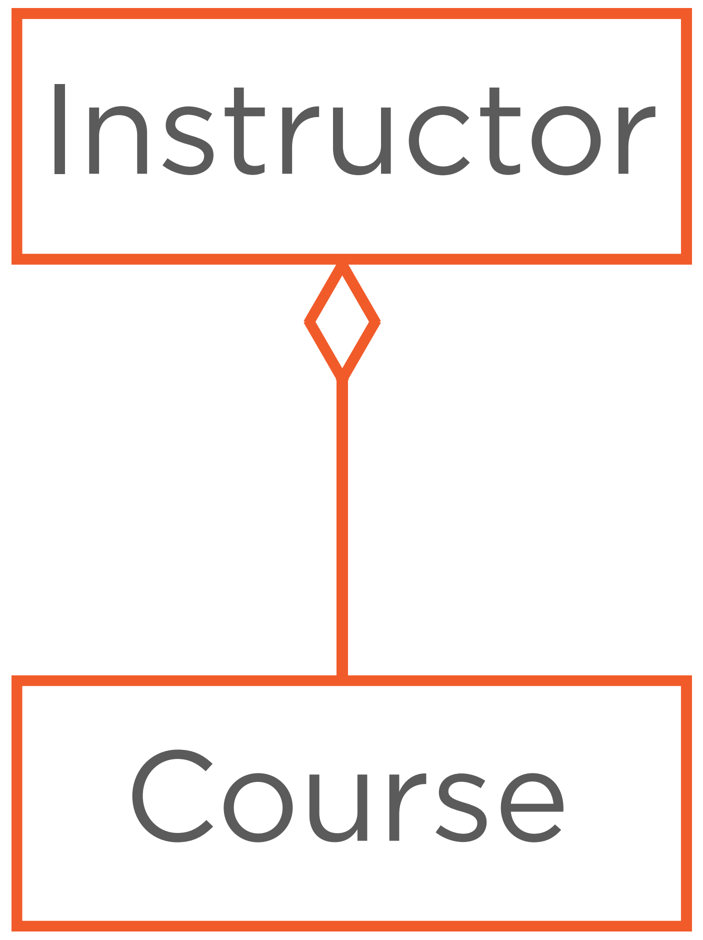
Navigability

One-way



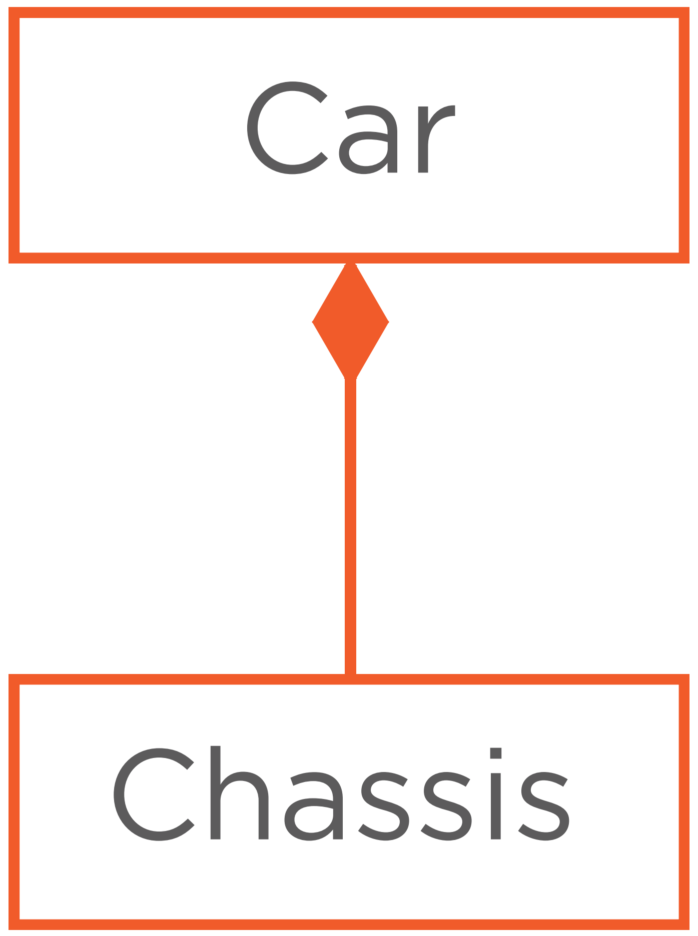
Aggregation

“has-a”



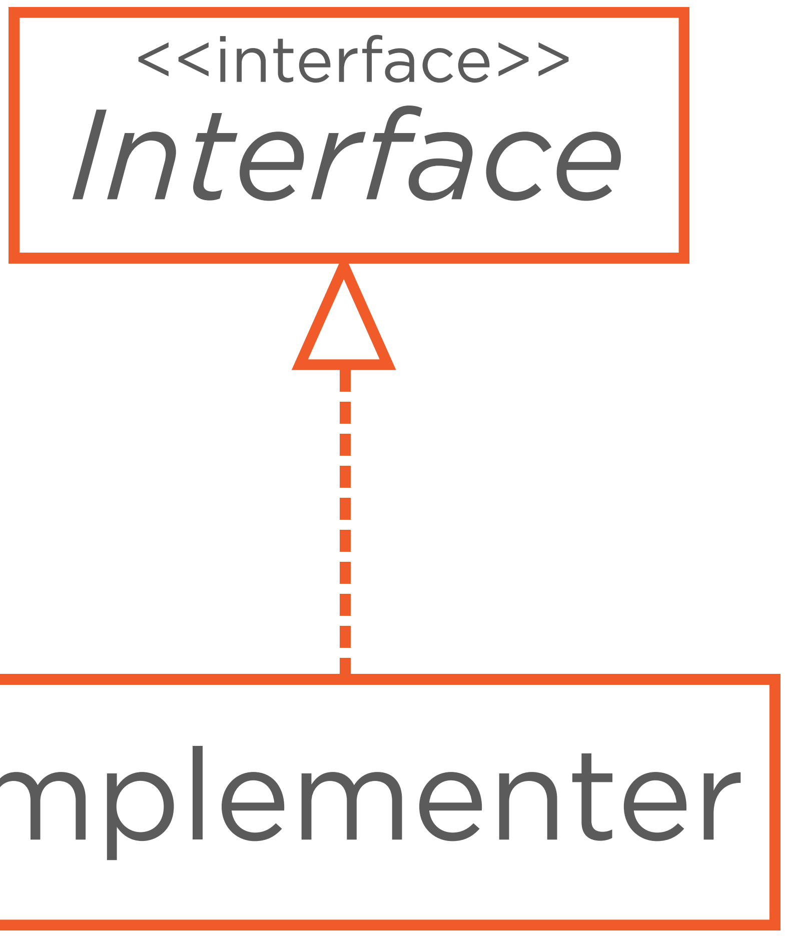
Composition

“part-of”



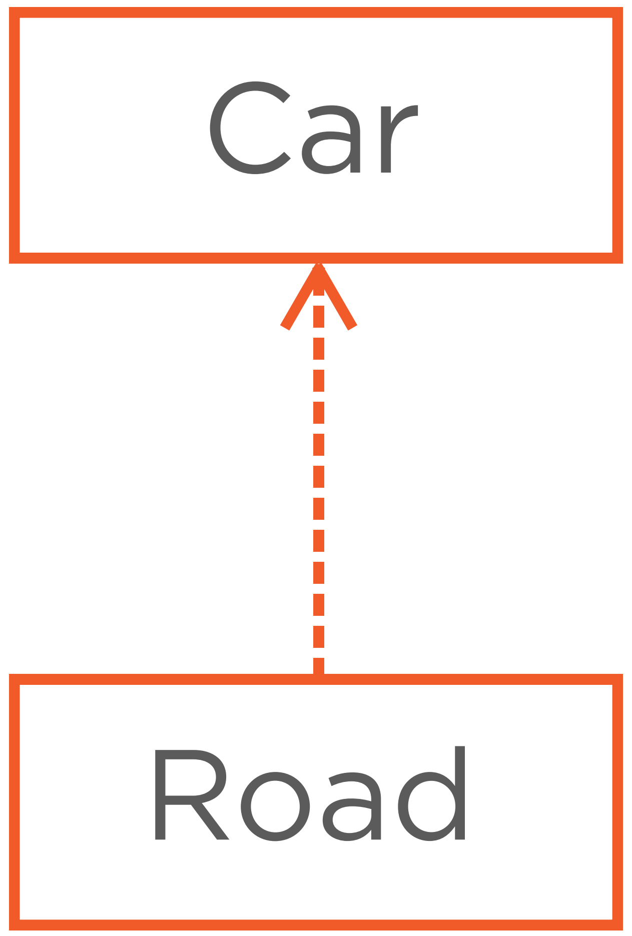
Realization

Implement Behaviour

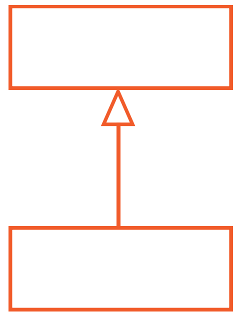


Dependency

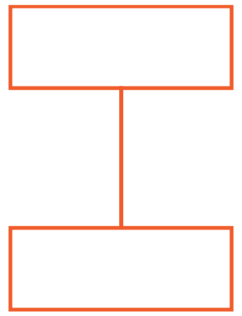
Weak Relations



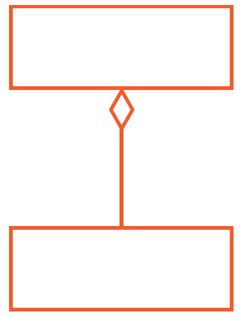
UML Relationhips



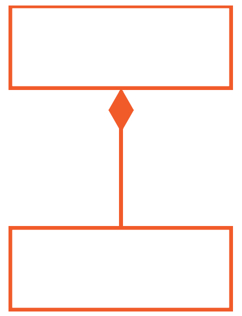
Generalization



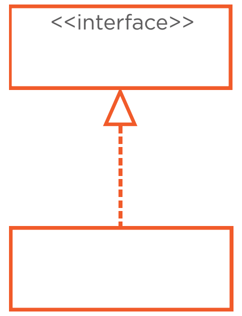
Association



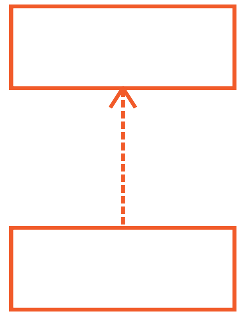
Aggregation



Composition



Realization



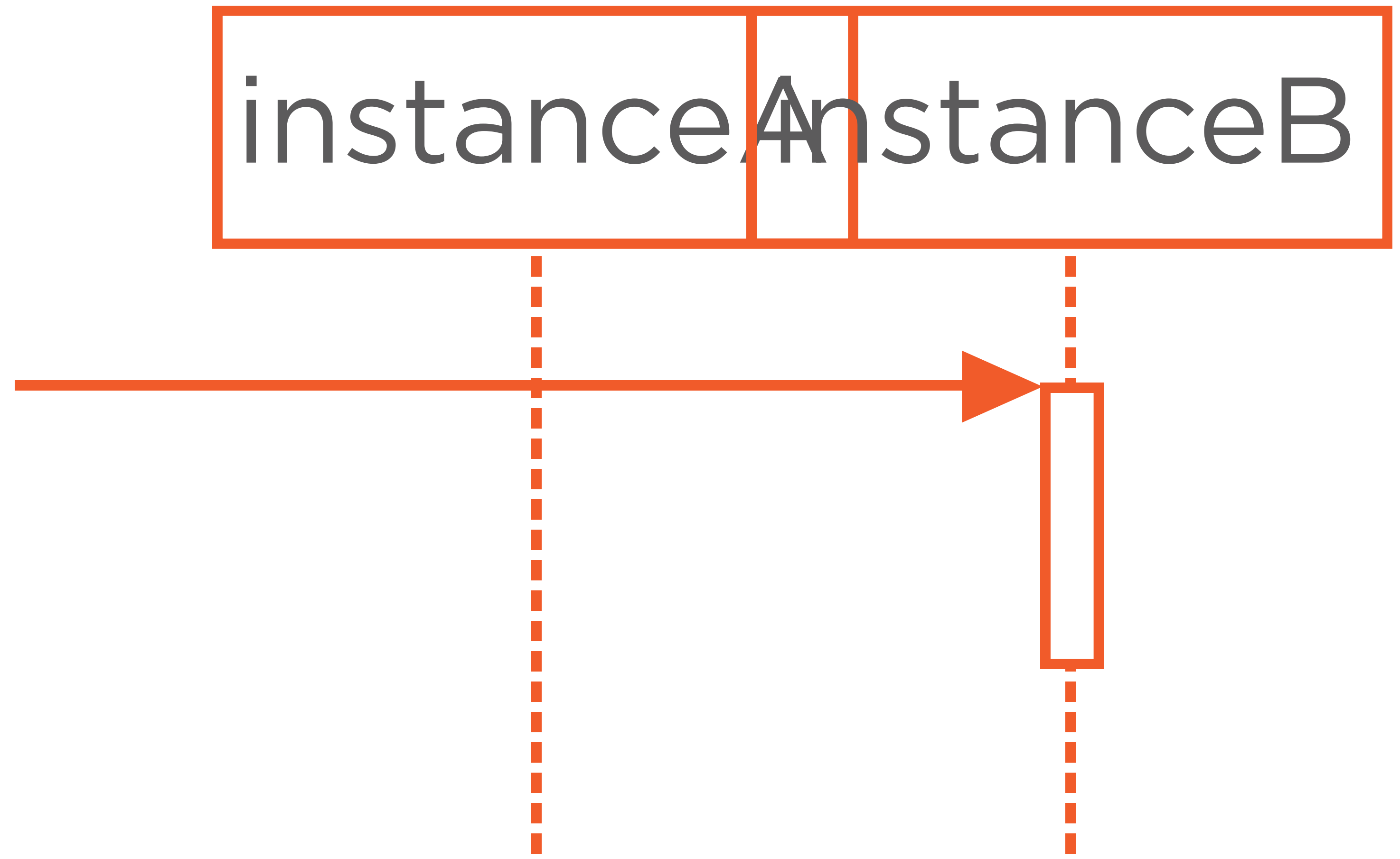
Dependency

Sequence Diagrams

Object Lifeline

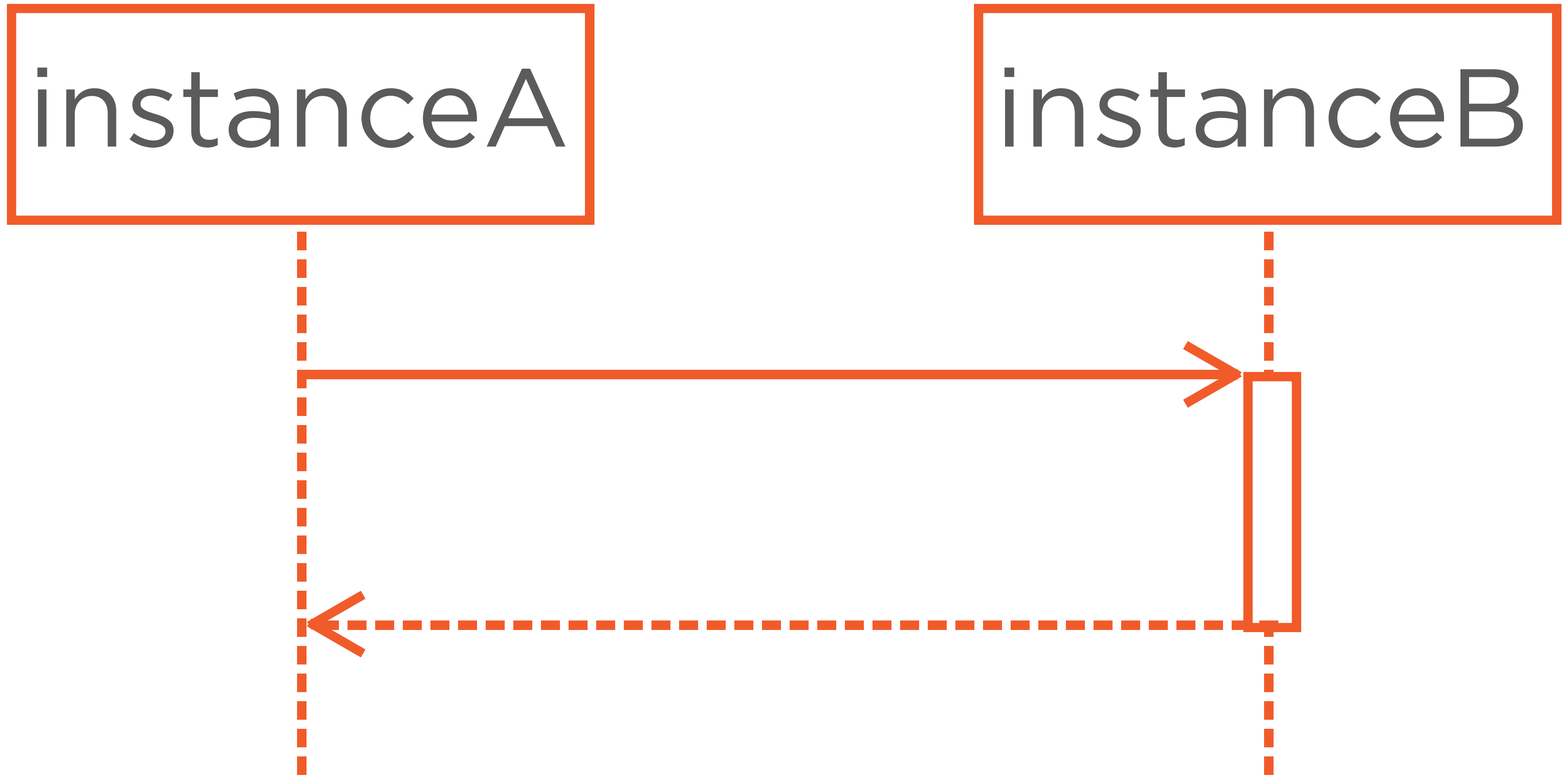
Message

Execution Occurrence

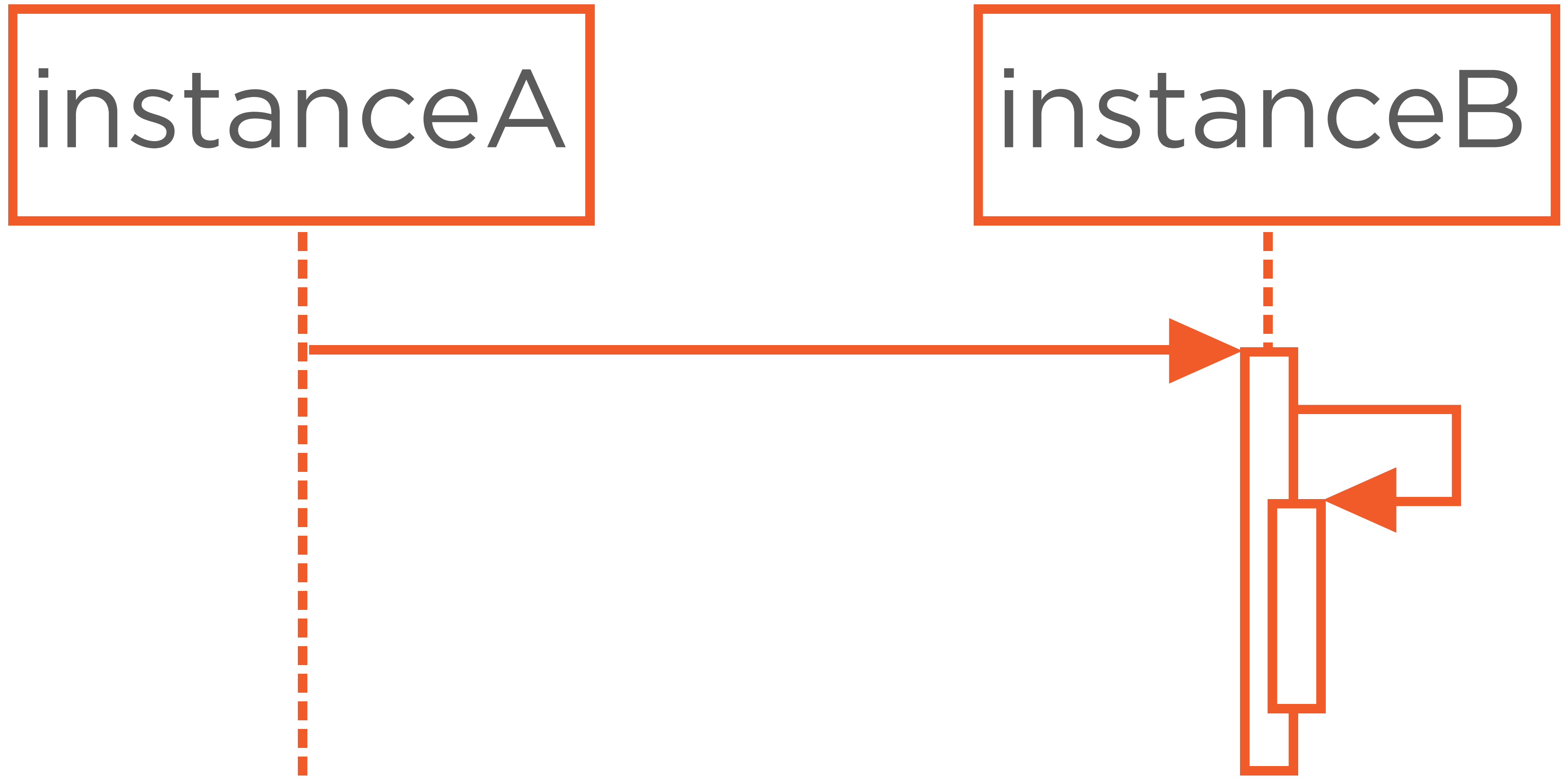


Asynchronous Messages

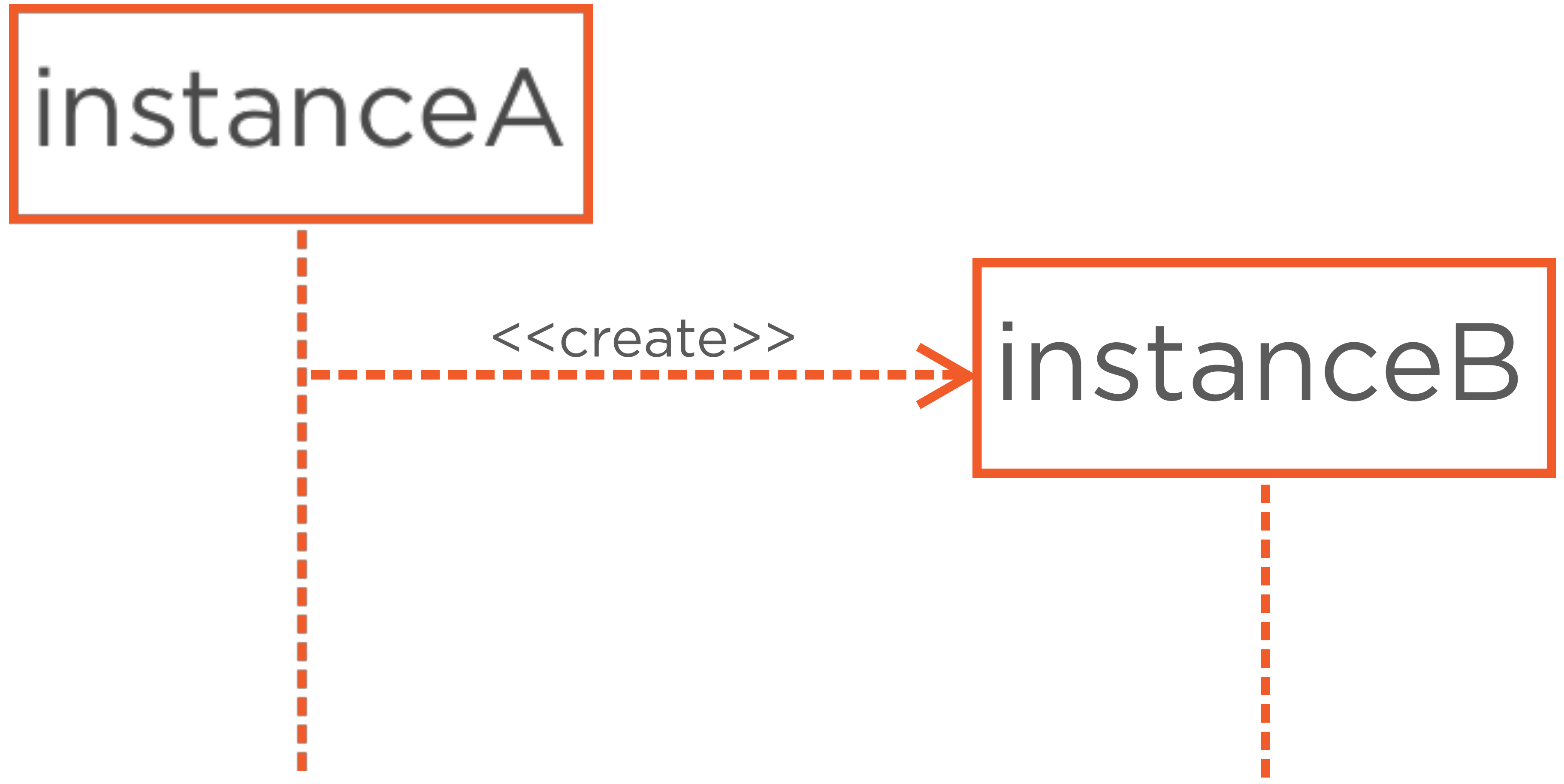
Async Return Messages



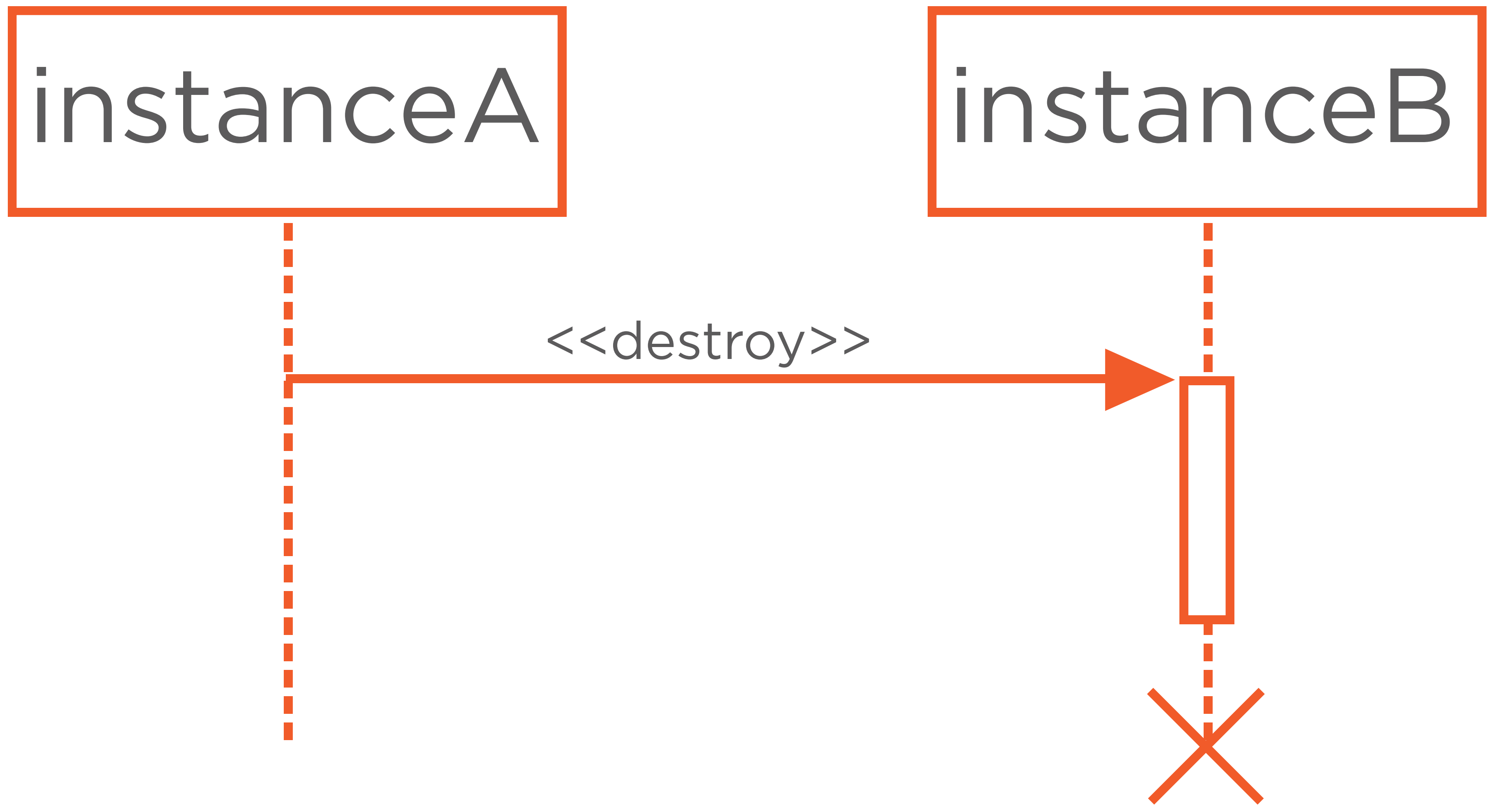
Message to Self



Lifeline Creation



Lifeline Termination



Sequence diagrams document the behavior of your system

Design Patterns: Classification

Categories

Creational Patterns

Structural Patterns

Behavioral Patterns

This course is about Creational Design Patterns in Swift