SWEN30006
Software Modelling and Design

Object-Oriented Analysis and Design

Larman Chapter 1

Time is a great teacher, but unfortunately it kills all its pupils.

—Hector Berlioz

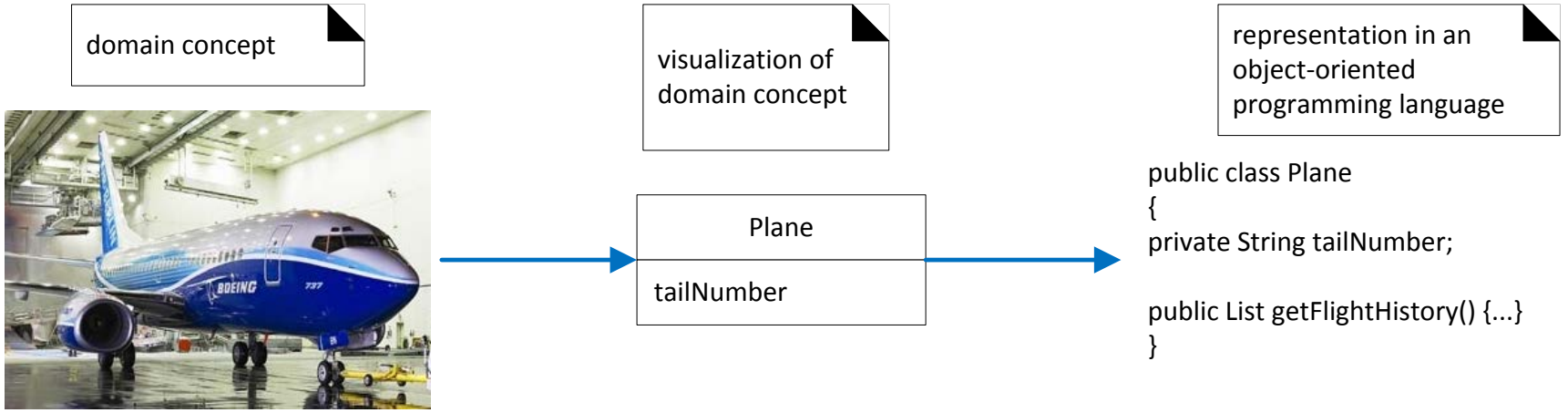
What will you learn?

- UML vs. Thinking in Objects
- OOD: Principles and Patterns
- Use Cases
- Iterative Development, Agile Modelling, and an Agile Unified Process (UP)
- Case Studies

What is Analysis and Design?

- ***Analysis***: investigation of the problem & requirements.
- ***Object-oriented analysis***: emphases finding and describing objects and concepts in the problem *domain*.
- ***Design***: a conceptual solution to a problem that meets the requirements.
- ***Object-oriented design***: emphasises defining software objects and their collaboration.
- ***Implementation***: a concrete solution to a problem that meets the requirements.
- **Object-oriented implementation**: implementation in object-oriented languages and technologies.

Object-Orientation Emphasizes Representation of Objects.



In simple terms, we want to understand the domain, represent it, define a solution and implement it, all in terms of objects.

Goals and Outcomes

You should be able to:

- Apply principles and patterns to create better designs, especially
  - skillfully assign responsibilities to S/W objects
  - with emphasis on the nine GRASP principles
- Iteratively follow a set of common analysis and design activities (based on Agile/UP)
- Create frequently used models in UML

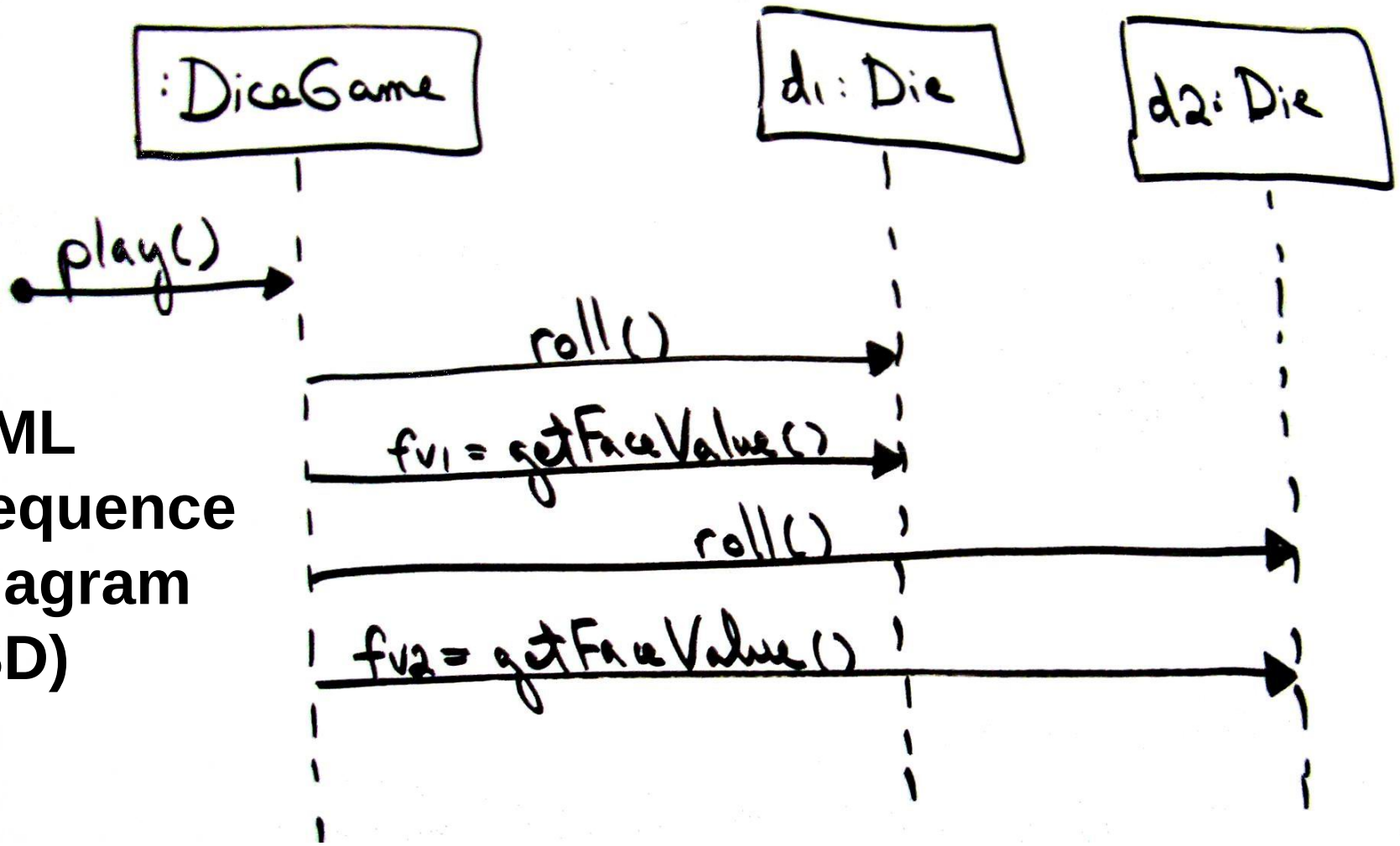
A Short Example: Dice Game (7 wins)

- Define Use Cases
  - *Play a Dice Game: Player requests to roll the dice. System presents results: if the dice face value totals seven, the player wins; otherwise, player loses.*
- Define a Domain Model
- Assign Object Responsibilities and Draw Interaction Diagrams
- Define Design Class Diagrams

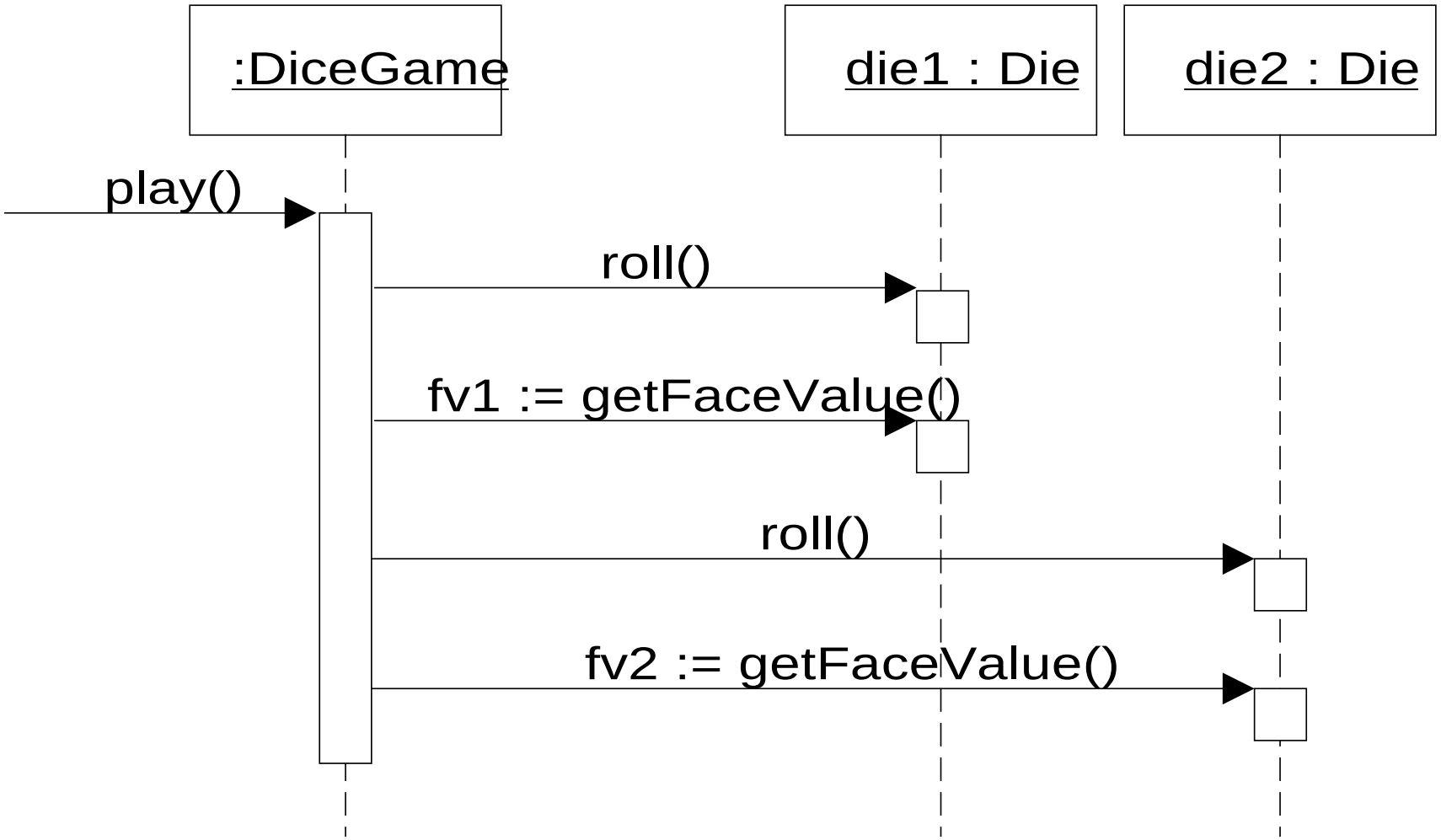
Partial Domain Model of the Dice Game

Messages between S/W Objects

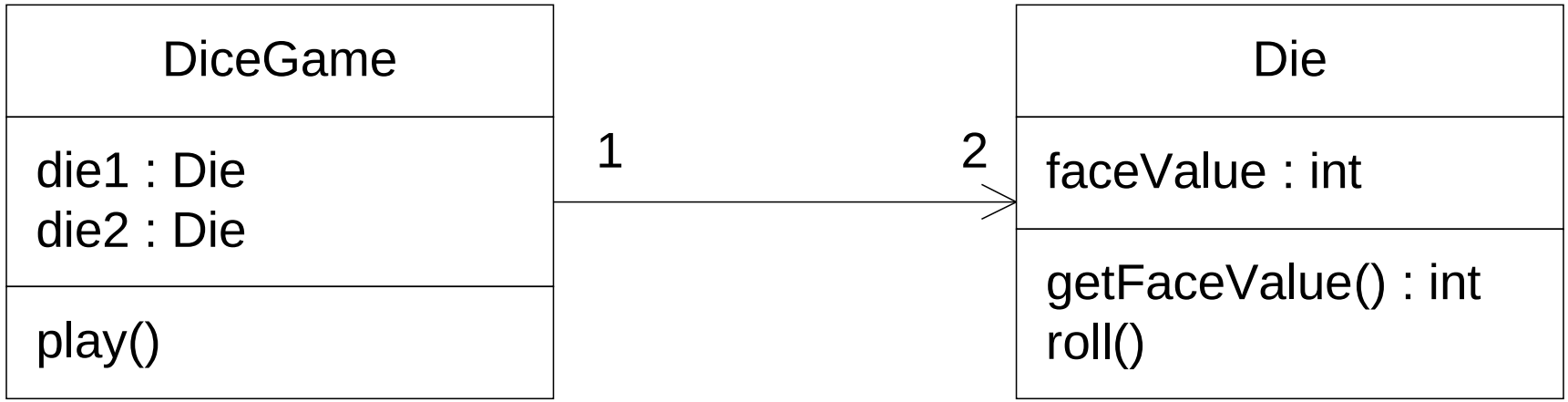
UML Sequence Diagram (SD)



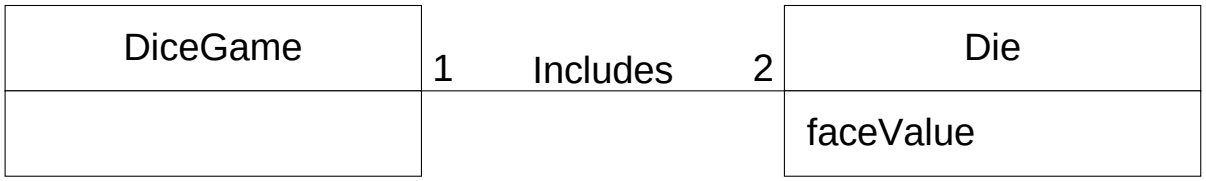
SD (again): UML Tool Version



Partial Design Class Diagram

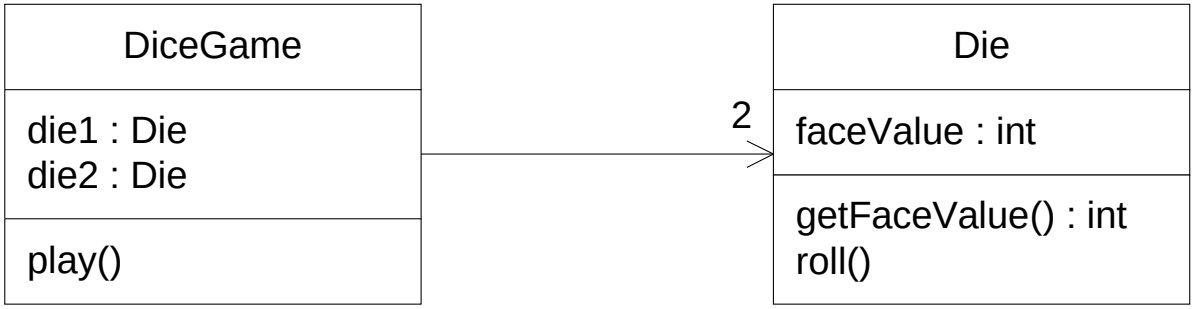


Different Perspectives with UML



Conceptual Perspective (domain model)

Raw UML class diagram notation used to visualize real-world concepts.



Specification or Implementation Perspective (design class diagram)

Raw UML class diagram notation used to visualize software elements.