Class Association and Aggregation

Class Association and Aggregation

- We discuss how create complex classes from simpler ones
- Distribute responsibilities/functionality among classes (and families of classes)
- I Coupling (loose, tight) between classes
- (no inheritance for the moment)

Aggregation

- ı Aka Whole-Part pattern
- We model an object as consisting of other objects
- I Common technique for assembling objects into bigger ones
- I There can be a fixed number or a variable number of parts in the Whole

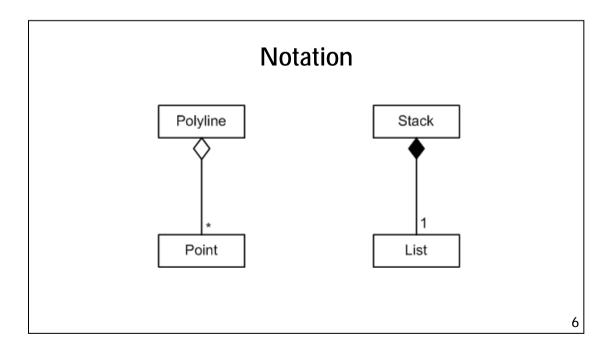
3

Examples (1/2)

- I A line segment consists of two points; a polyline consists of zero or more points
- I A house consists of rooms
- I An area consists of machines
- I An auto consists of wheels etc.

Examples (2/2)

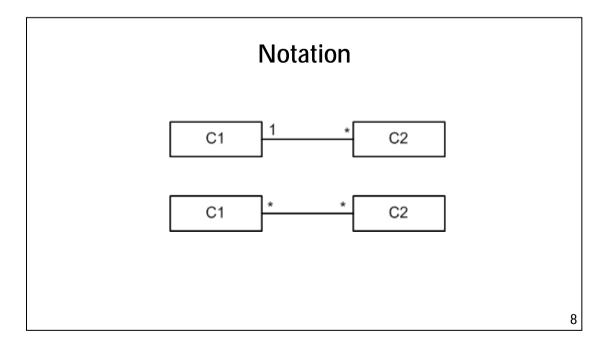
- A portfolio is a collection of assets (bonds, stock)
- I An interest floor is a series of put options
- I An instrument contains a collection of properties (see discussion in MC book)
- A structured product is made up of a portfolio of securities and derivatives
- A portfolio consists of portfolios



Association

- ı 'Loose' relationship between two classes
- Neither class is 'owner' (in contrast to Aggregation)
- The two classes form a new class, in essence
- Multiplicity issues

•



Example

- I A person works for a company
- I An option is based on one or more assets
- I An asset has an asset model
- I An option has a pricing model

C

Combining Models

- It is possible to use inheritance and aggregation to form flexible software systems
- ı Separation of concerns and Single Responsibility Principle (SRP)
- Delegation principle

