Object Oriented Analysis & Design
面向对象分析与设计

Lecture_07 通用的职责分配软件原则 GRASP

主讲: 姜宁康 博士

1、GRASP原则一: 创建者 Creator

- **Who should be responsible for creating a new instance of some class**
  由谁来负责创建某个类的新实例（对象）

1.1 GRASP 原则

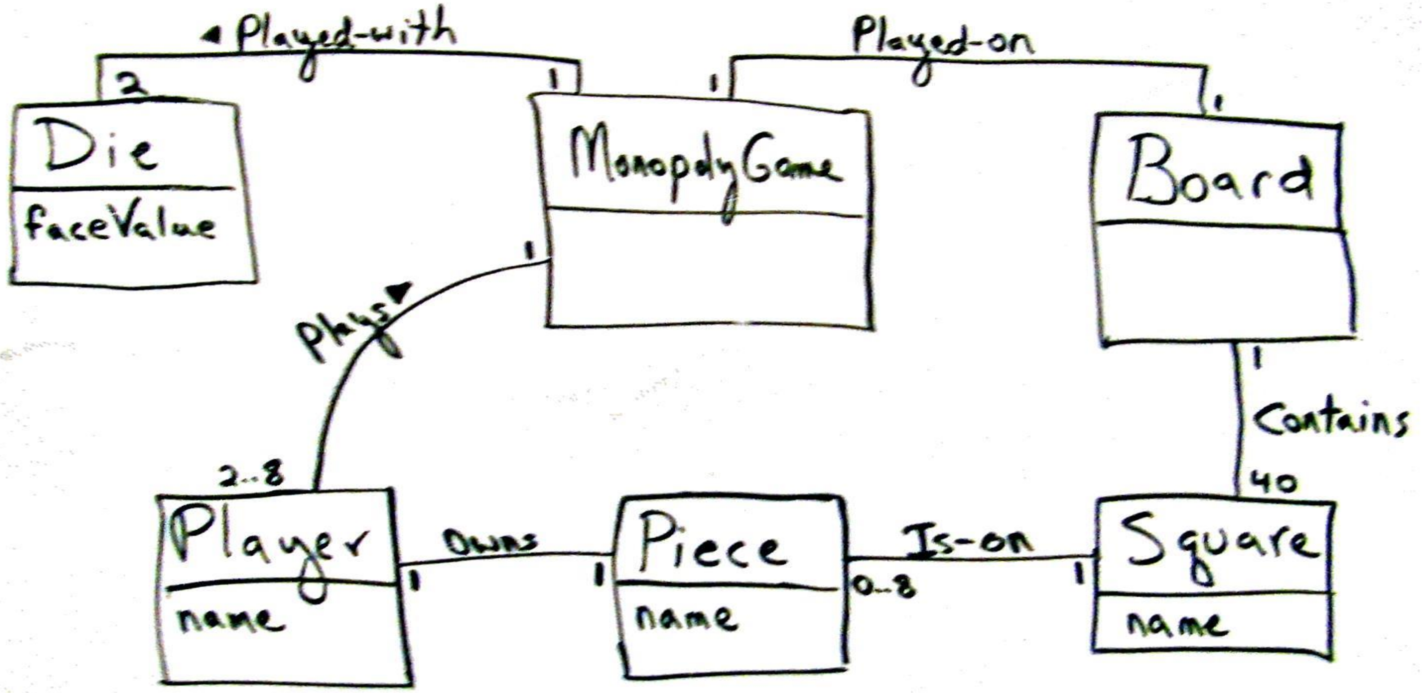
- GRASP
  - General Responsibility Assignment Software Patterns
  - 初学对象技术的同学，在编码或绘制交互图和类图时，应该理解并应用GRASP的内在思想，以便尽快地掌握这些基本原则，它们是设计OO系统的基础
  - GRASP原则可以帮助设计人员理解面向对象设计的本质，并以一种有条理的、理性的、可解释的方式应用这些设计原则 The GRASP patterns are a learning aid to help one understand essential object design, and apply design reasoning in a methodical, rational, explainable ways

1.1 GRASP 原则

- GRASP 原则共9条
  - Creator
  - Information Expert
  - Low Coupling
  - Controller
  - High Cohesion

  - Polymorphism
  - Indirection
  - Pure Fabrication
  - Protected Variations

1.2 Mini Exercise 1



Monopoly iteration-1 domain model

- Who will create Square object in Monopoly ? Why?
- Thinking steps
  - Have no design model, so start with Domain Model
  - LRG: Low representational gap — build design model

1.3 GRASP rule1: Creator (创建者)

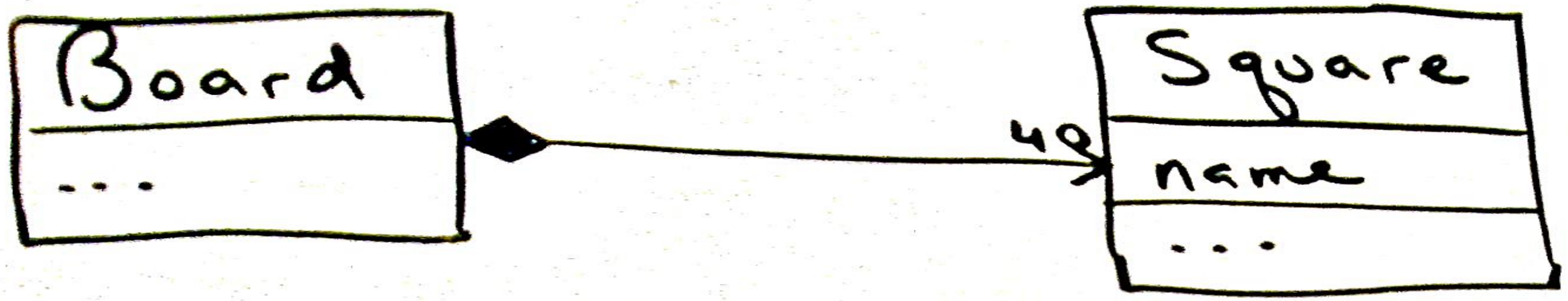
- **Name: Creator**
- **Problem:**
  - **Who should be responsible for creating a new instance of some class?**
- **Solution:**
  - **Assign class B the responsibility to create an instance of class A if one of these is true *(the more the better)* :**
  - **1. B "contains" or compositely aggregates A**
  - **2. B records A**
  - **3. B closely uses A**
  - **4. B has the initializing data for A that will be passed to A when it is created (B is an expert with respect to A)**

如果有一个以上的选项适用，通常首选聚集或包含A的类

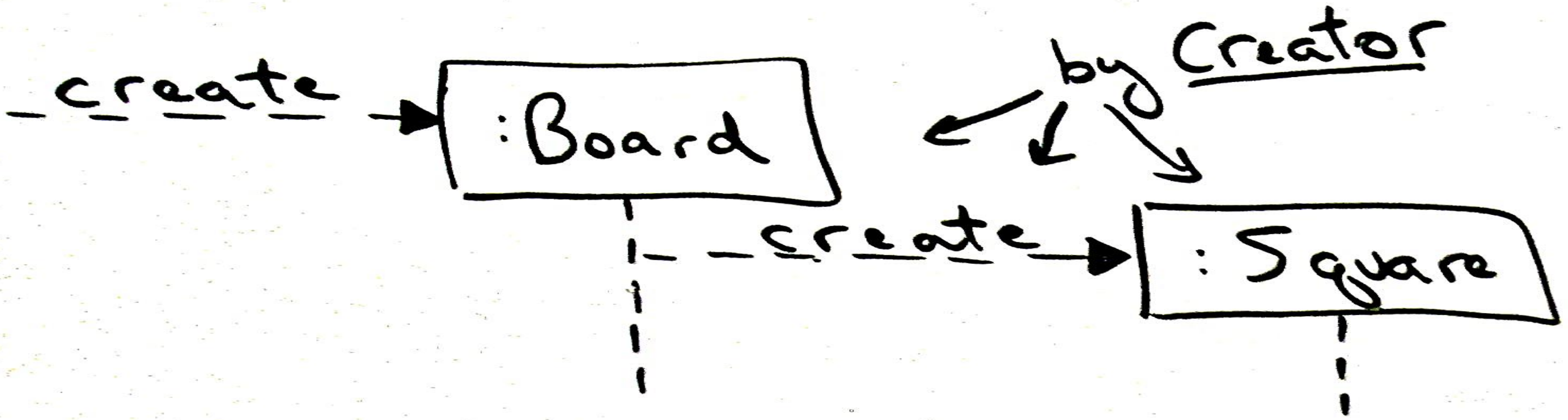
- **Note:**
  - **B and A refer to software objects, not domain model objects**

1.4 Creator: example Monopoly

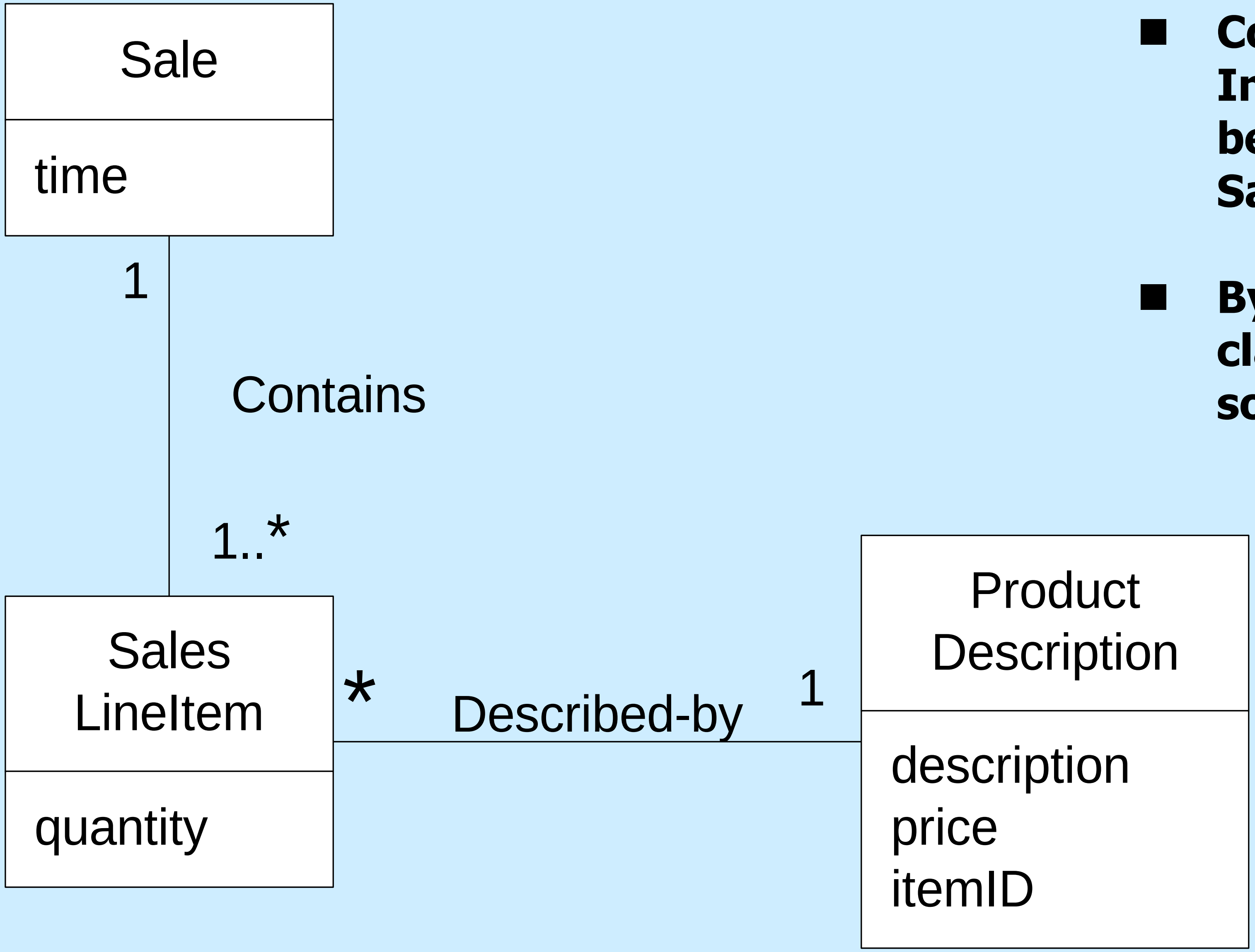
- Board create Square
  - Because of 'Board' contains 'Square'



Who create 'Board'?



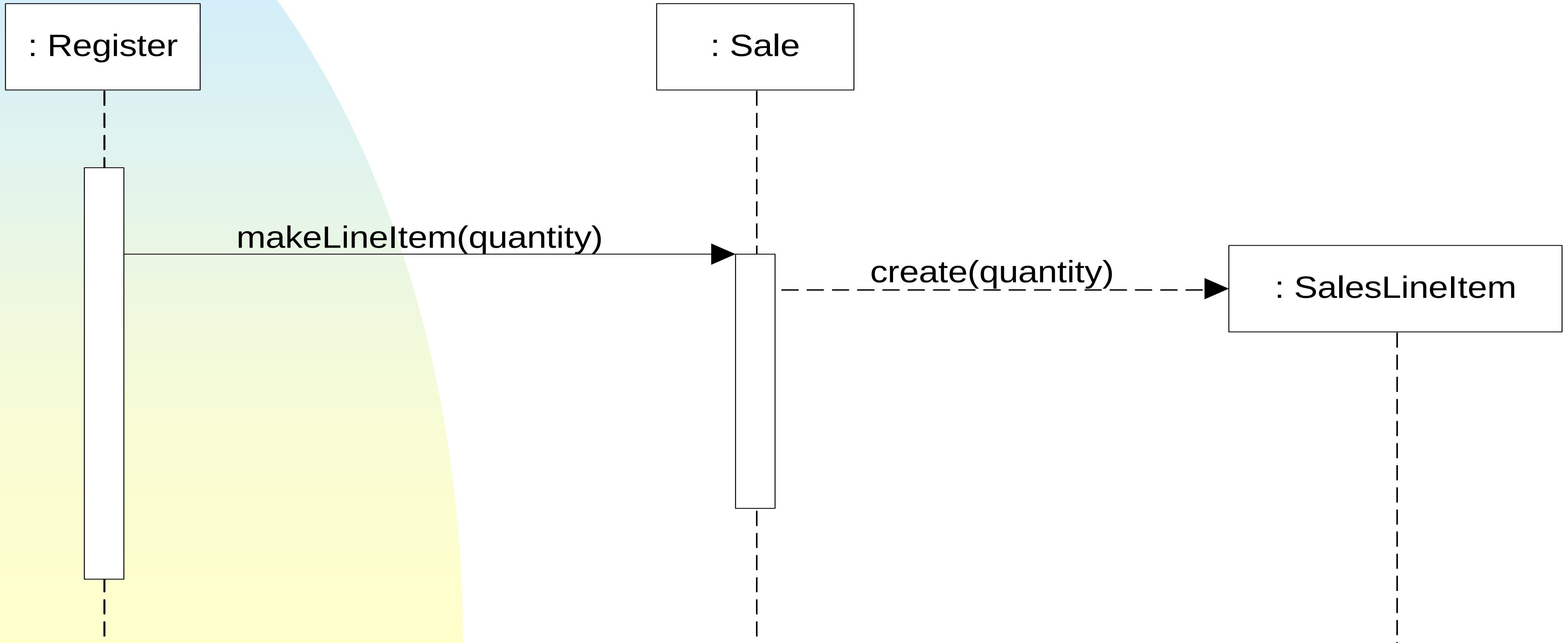
1.5 Creator : POS example



- Consider the partial domain model : In the POS application, who should be responsible for creating a SalesLineItem instance?
- By Creator, we should look for a class that aggregates, contains, and so on, SalesLineItem instances

1.5 Creator : POS example

- Since a Sale contains (in fact, aggregates) many SalesLineItem objects, the Creator pattern suggests that Sale is a good candidate to have the responsibility of creating SalesLineItem instances
- This leads to a design of object interactions as shown :



1.6 Discuss: Creator — When Not to Use

- When want to reuse existing instances for performance purposes (caching)
- conditionally creating an instance from one of <u>a family of similar classes </u>based upon some external property value
  - GOF pattern
- Delegate responsibility further down
- Other more complex situations

1.6 Discuss: Creator — Benefits

- Existing associations means created class is in any case visible to creator
- High cohesion
- Does not increase coupling

- 本讲结束