

# **GRASP 2**

## **Software Development**

**NIFR 2021**

# GRASP 2

## Software Development

- Patterns
  - Creator
  - Factory

# GRASP

## Creator

Problem: Who creates object A?

Solution: In general, Assign class B the responsibility to create object A if one, or preferably more, of the following apply:

- Instances of B contain or compositely aggregate instances of A
- Instances of B record instances of A
- Instances of B closely use instances of A
- Instances of B have the initializing information for instances of A and pass it on creation.<sup>[3]</sup>:

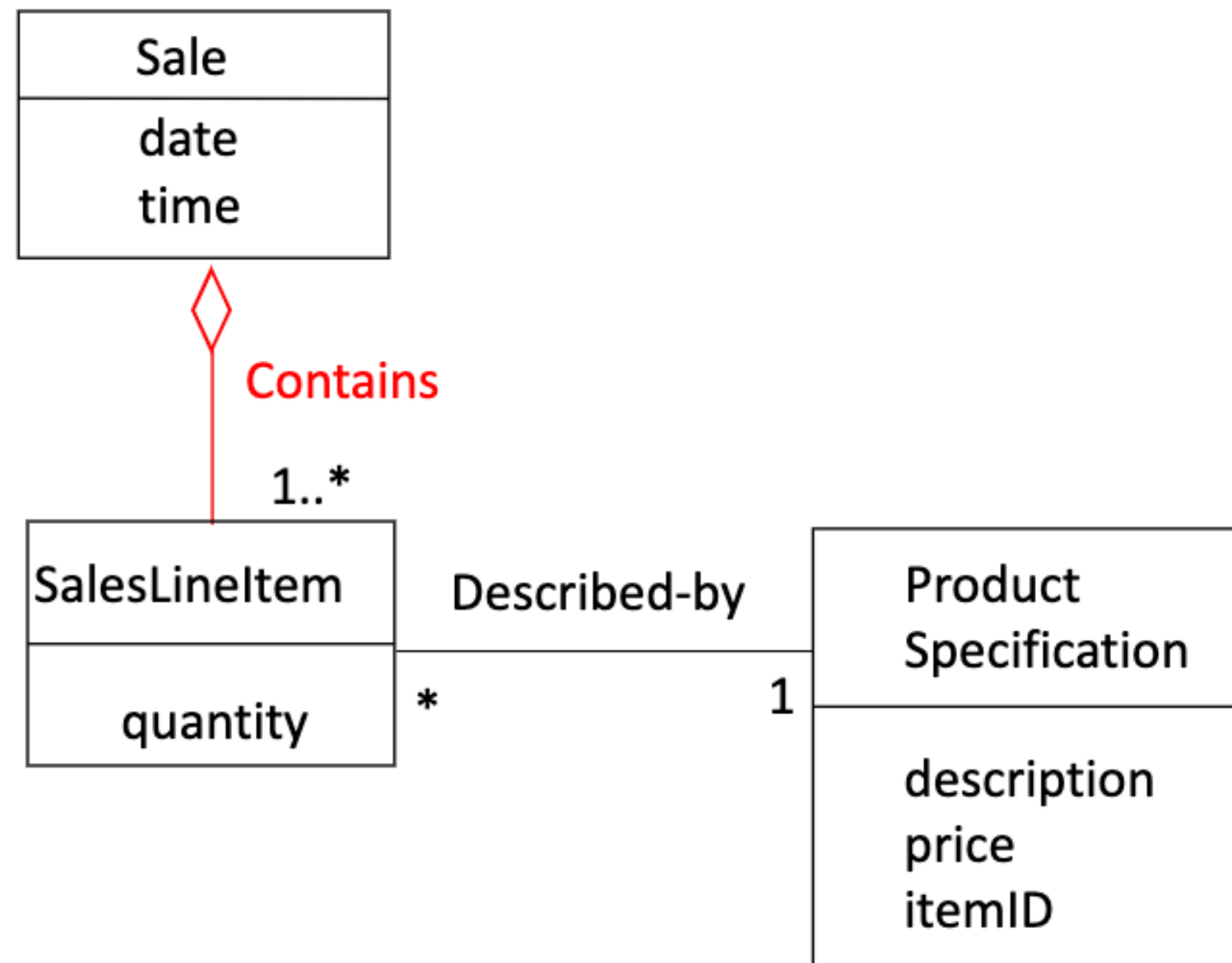
16:16.7

# Creator

```
public class Robot {  
    private int studentid;  
    private RobotArm right;  
    private RobotArm left;  
}
```

Who creates robot arms?

The robot as it contains and aggregates instances of Arms & uses them



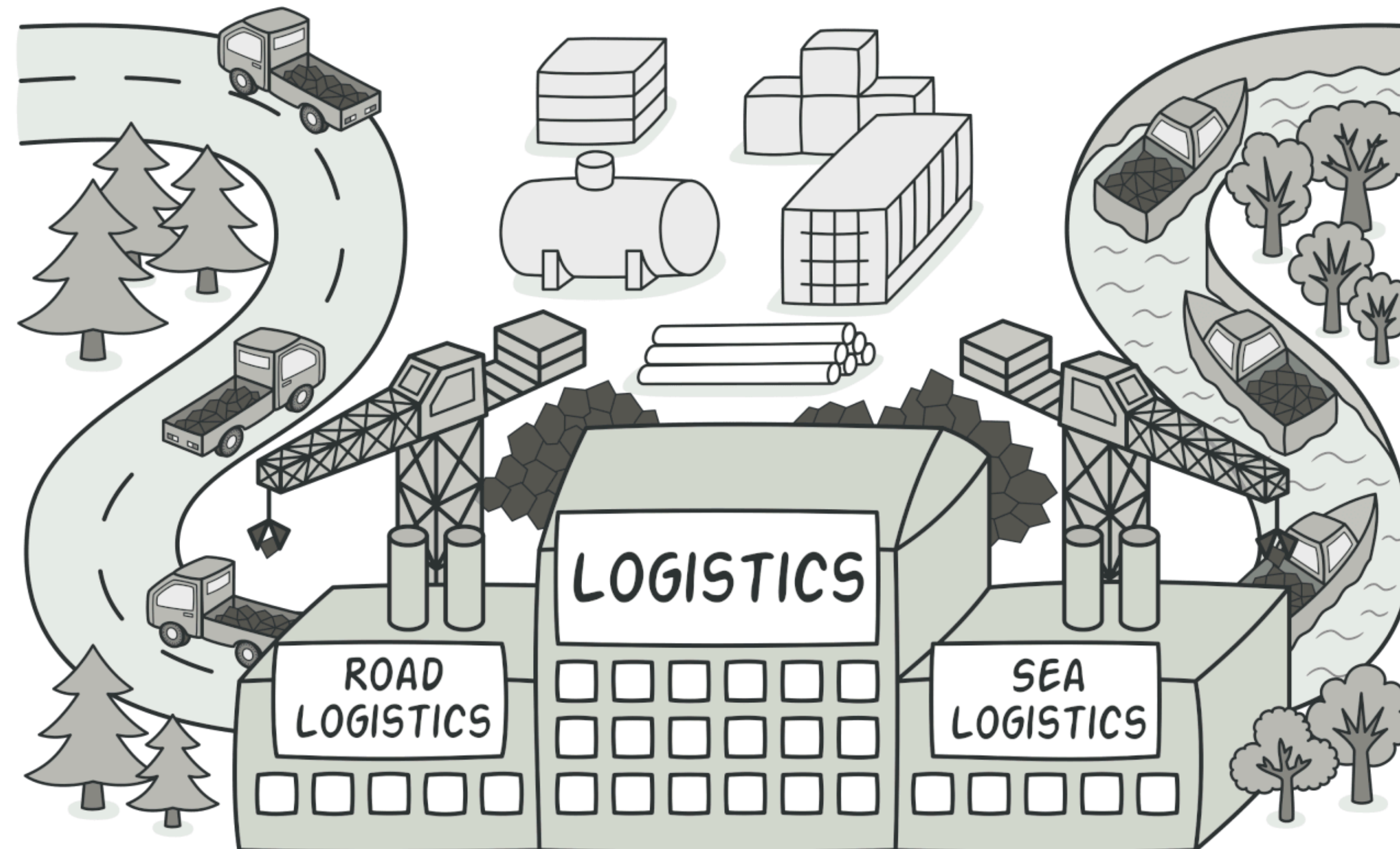
- Who should be responsible for creating a SalesLineItem instance?
- Since a Sale **contains** many SalesLineItem objects, the Creator pattern suggests that Sale is a good candidate.

**Creator supports low coupling**

# Factory pattern

## Intent

**Factory Method** is a creational design pattern that provides an interface for creating objects in a superclass, but allows subclasses to alter the type of objects that will be created.



**Code example:  
Implementing the factory pattern  
for shapes**