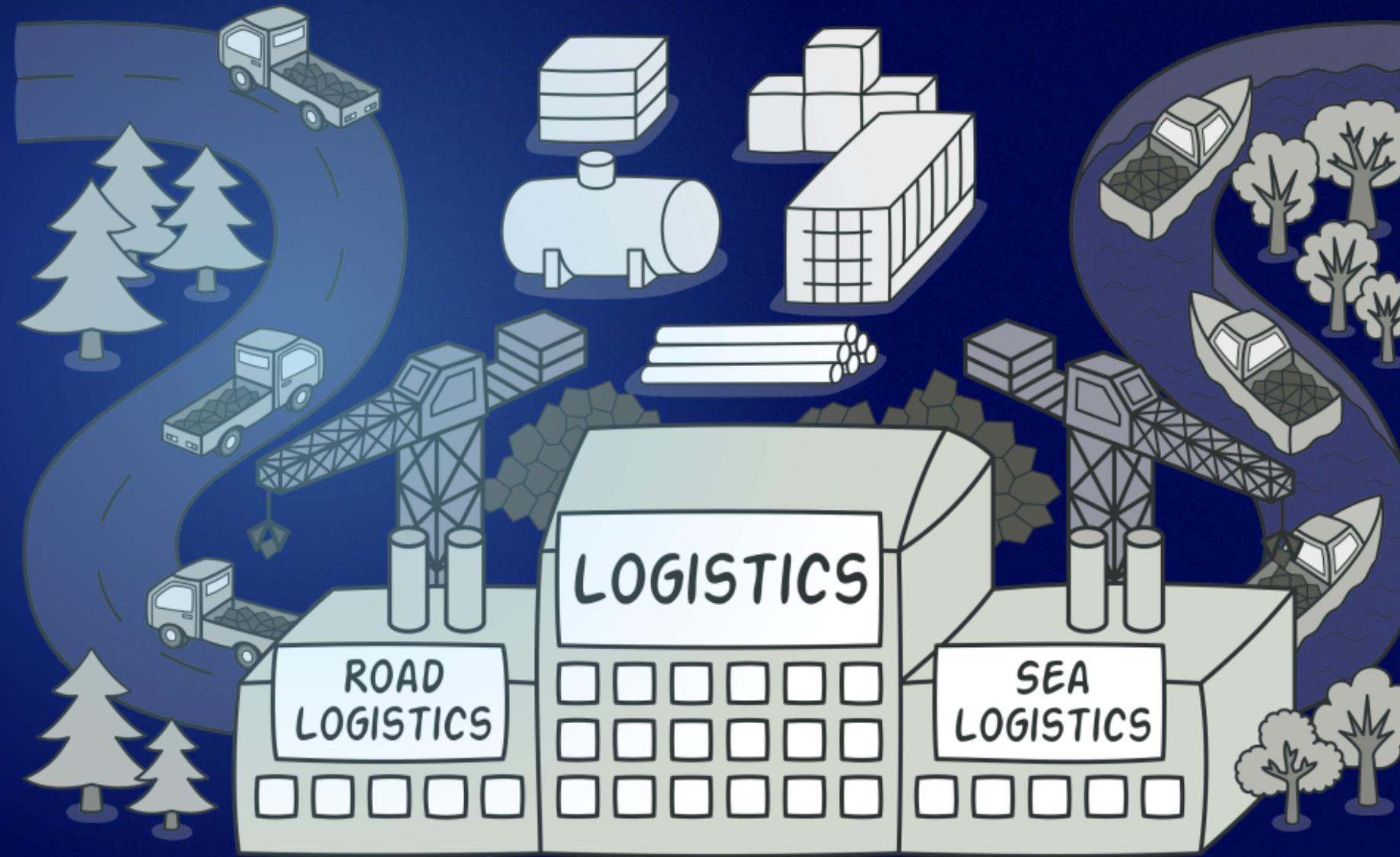


الگو کارخانه ای (FACTORY PATTERN)



معرفی

الگوی کارخانه‌ای بر پایه‌ی اصل وراثت در برنامه‌نویسی شیء‌گرا عمل می‌کند

کلاس والد، متدهای کارخانه‌ای را تعریف من‌کند



زیرکلاس‌ها می‌توانند متدهای والد را بازنویسی من‌کنند



هر زیرکلاس شیء مخصوص به خودش را ایجاد من‌کند



مشکل

فرض کن یک اپلیکیشن مدیریت لجستیک داری
نسخه‌ی اول فقط حمل و نقل با کامیون را پشتیبانی می‌کند.
پس از مدتی، درخواست‌های زیادی دریافت می‌کنی
برای اضافه کردن حمل و نقل دریایی 

راه حل اول:

میای یک کلاس SHIP کنار کلاس TRUCK تعریف می‌کنی.

✖ مشکل اینجاست:

باید کل کد پروژه تغییر کند،

چون همه‌جا به‌طور مستقیم از TRUCK استفاده شده است.

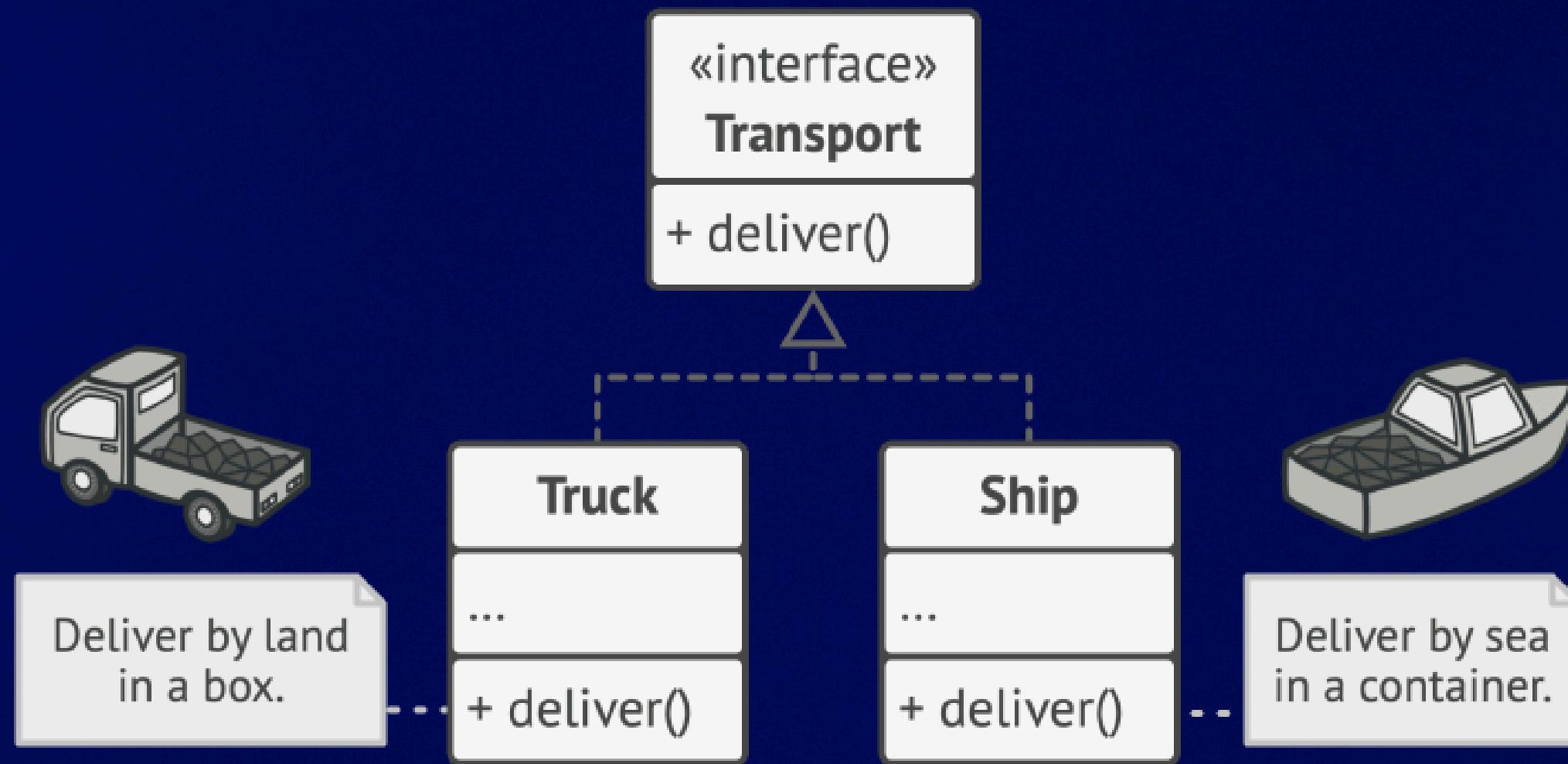
راه حل دوم:

استفاده از FACTORY METHOD ✓

- ۱- تعریف یک کلاس والد به نام TRANSPORT
- ۲- ایجاد زیرکلاس‌ها: TRUCK و SHIP
- ۳- بازنویسی متدها در هر زیرکلاس

برای ساخت شیء مخصوص به خودش

راه حل دوم:



کی استفاده کنیم

📌 زمانی که نوع شیء مشخص نیست

اگر از قبل نمی‌دانی چه شیئی باید ساخته شود، FACTORY METHOD امکان تصمیم‌گیری در زمان اجرا را فراهم می‌کند مثل مثال بالا

📌 صرفه‌جویی در منابع سیستم

وقتی ساخت دوباره شیء پرهزینه است، می‌توان از اشیاء موجود دوباره استفاده کرد و حافظه و زمان اجرا را بهینه کرد. کلاس TRANSPORT برای بقیه حمل و نقل ها هم استفاده می‌شود مانند حمل و نقل هوایی

نکات کلیدی:

- ♦ بسیاری از طراحی‌ها با FACTORY METHOD شروع می‌شوند و به مرور به سمت BUILDER یا ABSTRACT FACTORY می‌روند.
- ♦ در واقع یک نمونه‌ی خاص از FACTORY METHOD هم هست. TEMPLATE METHOD

جمع بندی

FACTORY METHOD یعنی:

- ✓ یک الگوی ساده اما قدرتمند
- ✓ برای ایجاد انعطاف در ساخت اشیاء.
- ✓ هر جا که به گسترش پذیری نیاز داشته باشی،
- ✓ این الگو می تواند ناجی تو باشد