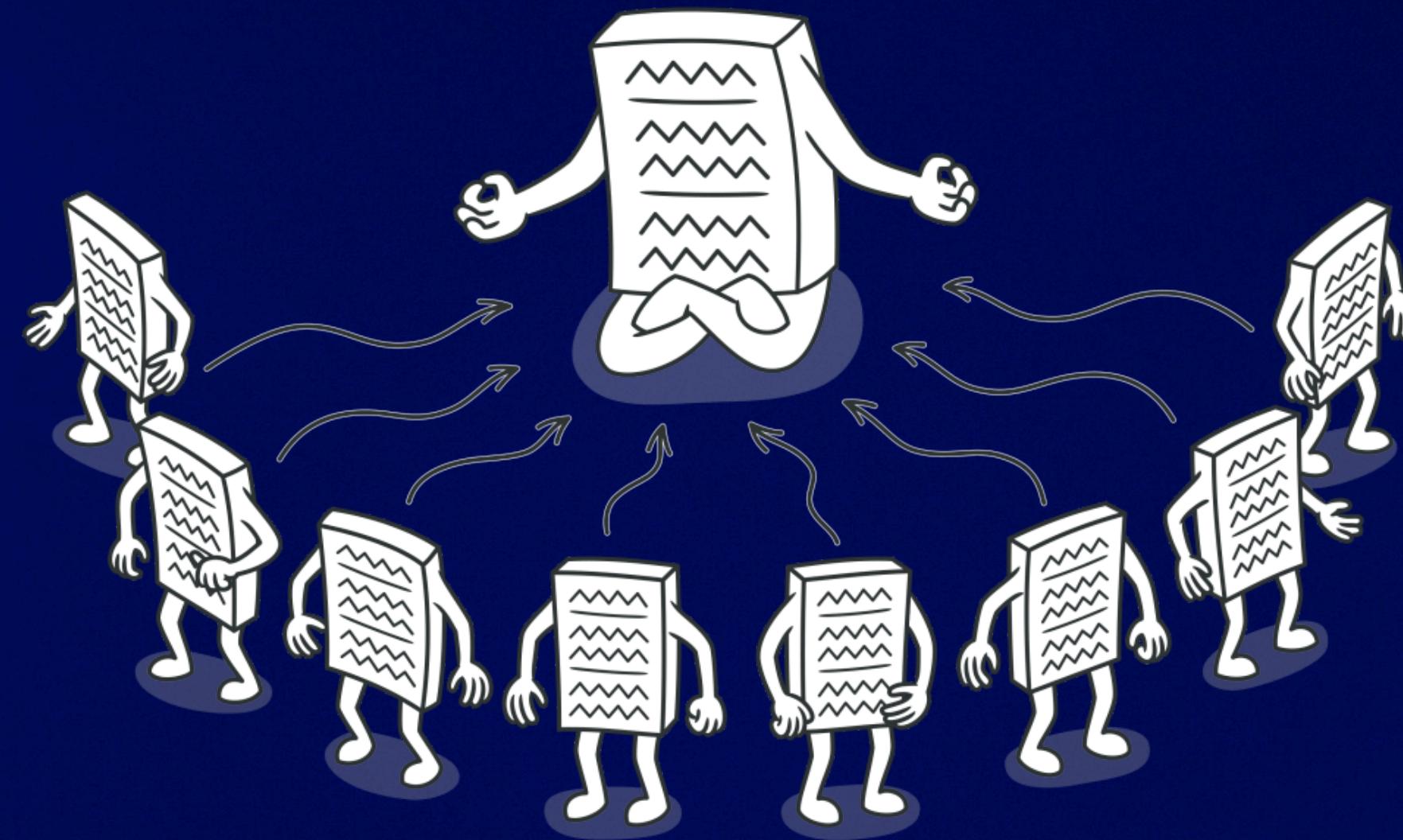


الگو تک نمونه ای یا (SINGLETION PATTERNS)



معرفی

گاهی لازمه فقط یه نسخه از یه شئ توی کل برنامه‌مون وجود داشته باشه.

مثلاً یه تنظیمات مرکزی، یه LOGGER یا یه اتصال به دیتابیس.

اینجا همون جاییه که الگوی SINGLETION (تک نمونه‌ای) وارد منشه!

معرفی

الگوی **SINGLETON** می‌گه:

«من فقط اجازه می‌دم یه نمونه از این کلاس ساخته بشه.

و هر وقت بھش نیاز داشتی، از یه مسیر مشخص بھش دسترسی داری!»

به زبان ساده: یه شء مشترک برای کل برنامه 😎

مشکل

بعضی وقتا باید مطمئن بشیم فقط یه نمونه از یه کلاس ساخته من شه.

ولی معمولاً[¶] این نیاز دو تا چالش داره

کنترل تعداد نمونه‌ها

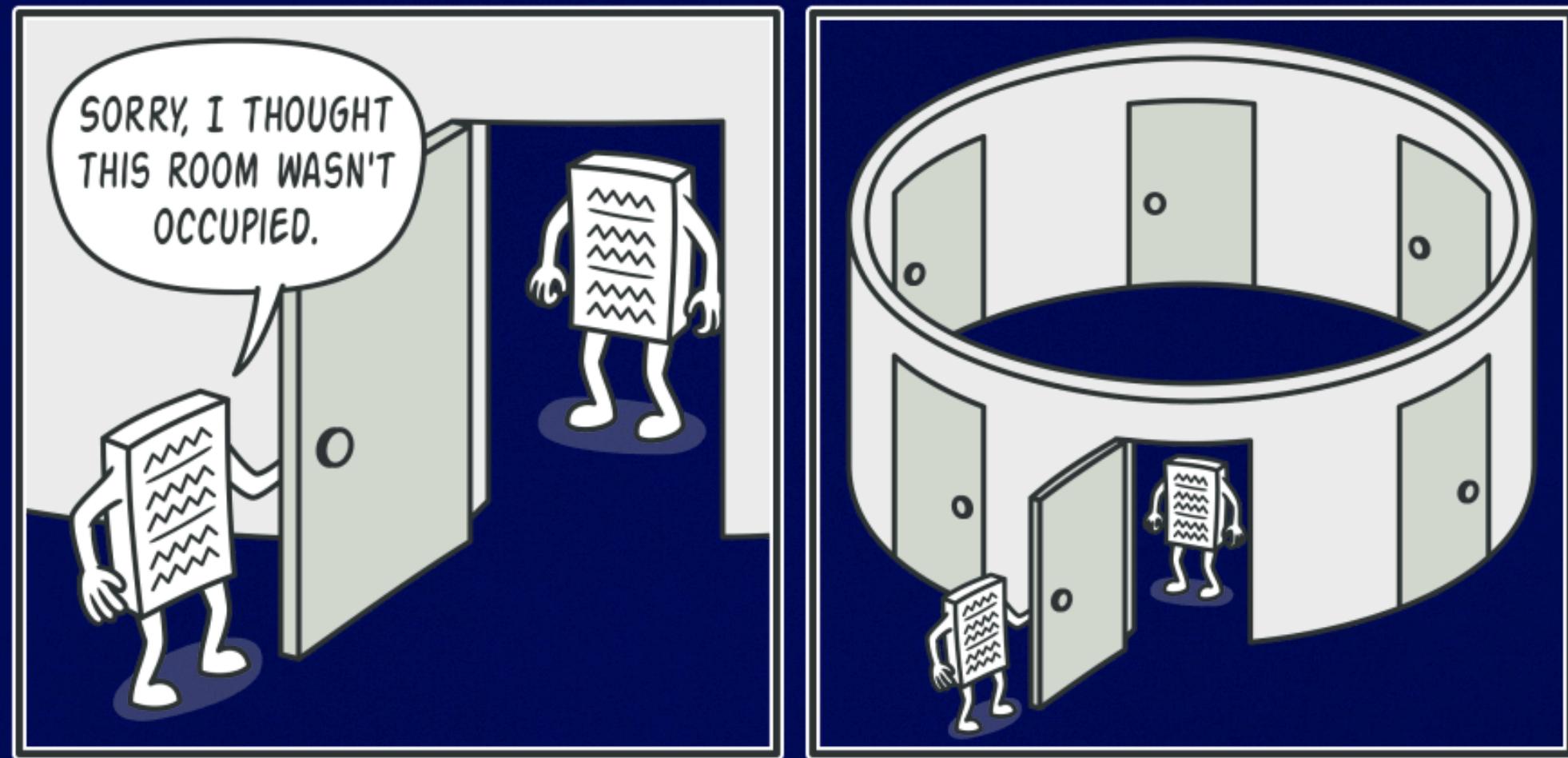
نمی‌خوای ده تا اتصال دیتابیس ساخته بشه! فقط یکی کافیه.

مشکل

دسترسی سراسری (GLOBAL ACCESS)

من خوای از هرجای برنامه به اون شیء دسترسی داشته باشی،
اما نمیخوای متغیر GLOBAL بسازی که نامن باشه.

مشکل



راه حل اشتباه

یه راه وسوسه‌انگیز استفاده از متغیرهای GLOBAL هست.

اما...

هر کس منونه مقدارش رو تغییر بده و برنامه رو به هم بربیزه 😊
نتیجه؟ کلی باگ و بی‌نظمی!

راه حل درست

الگوی **SINGLETON** می‌گه: "خودم مراقبم فقط یه نمونه ساخته بشه!"

- ♦ سازنده (PRIVATE) رو خصوصی می‌کنیم (CONSTRUCTOR)
- ♦ یه متده (GetInstance) می‌سازیم
- ♦ اون متده چک می‌کنه اگه شی ساخته نشده بود، بسازه
- ♦ بعد همیشه همون رو برمن‌گردونه
- ♦ نتیجه: همیشه یه شی واحد داری — هیچ تکراری!

مثال

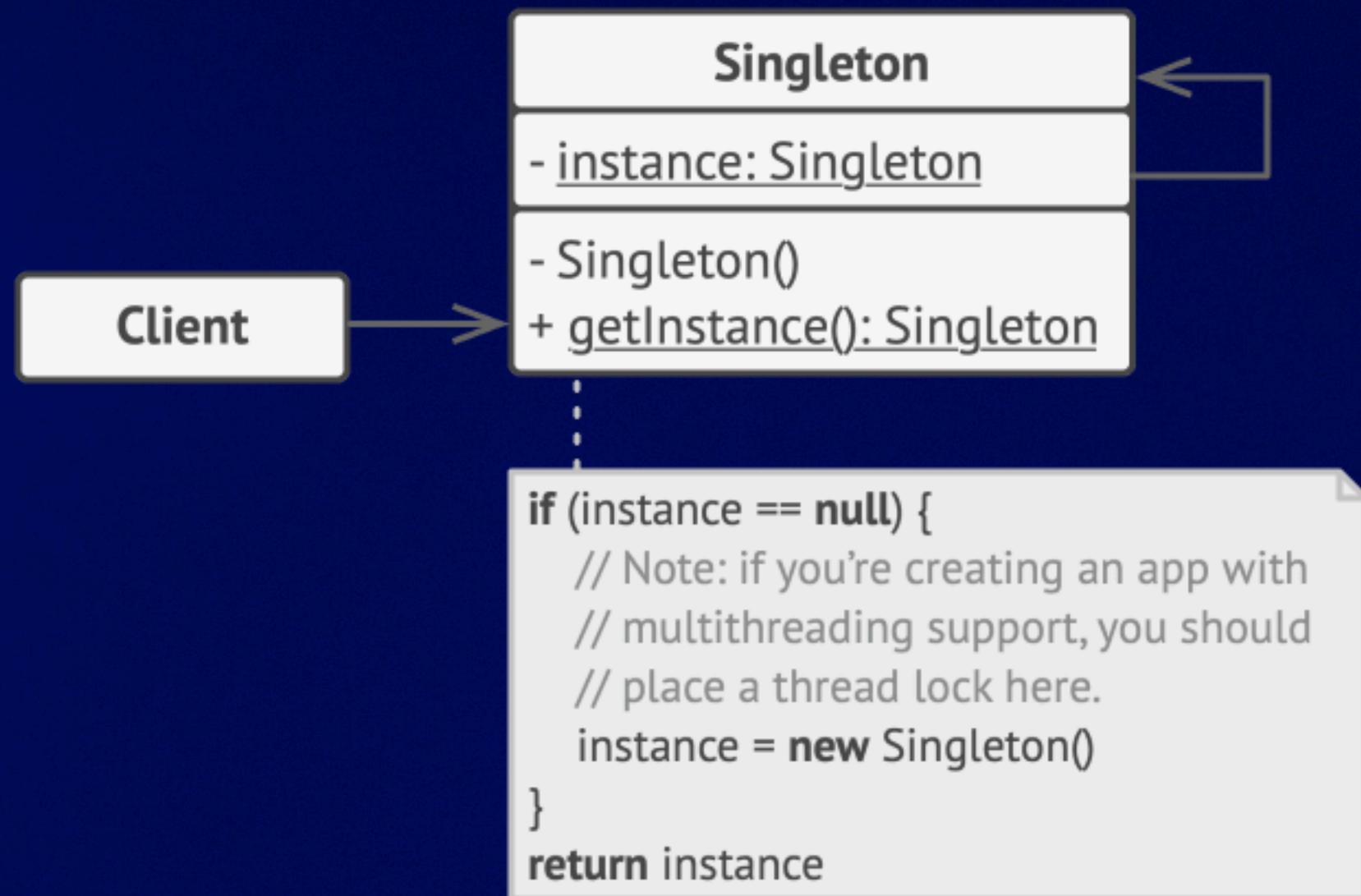
تصور کن رئیس یه شرکت فقط یکیه 

هر کس که بخواهد تصمیم مهم بگیره باید با همون یه نفر صحبت کنه.

نه من تویی رئیس جدید بسازی، نه من تویی اون رو جایگزین کنی!

SINGLETON دقیقاً همین کار رو تویی کد انجام من ده.

ساختار



کی استفاده کنیم

✓ وقتی فقط یه شی باید وجود داشته باشه:

- تنظیمات برنامه (CONFIG)

- اتصال به دیتابیس

- CACHE یا LOGGER •

✓ وقتی من خوای کنترل بهتری روی GLOBALها داشته باشی

رابطه با سایر الگوها

PROTOTYPE و ABSTRACT FACTORY, BUILDER ◆

من تون خودشون به صورت SINGLETION پیاده‌سازی بشن تا فقط یه نمونه ازشون وجود داشته باشه.

نکات کلیدی

GLOBAL VARIABLE دسترسی جهانی امن‌تر از SINGLETON می‌ده. 

ولی اگه زیاد استفاده پشه، می‌تونه باعث وابستگی زیاد (COUPLING) بشه. 

در تست‌نویسی (UNIT TESTING) هم گاهی در درس درست می‌کنه! 

نسل‌های مختلف پیاده‌سازی SINGLETION

* نسخه‌های متنوع :SINGLETION

(وقتی نیاز شد ساخته منشئ) LAZY INITIALIZATION .

(از اول ساخته منشئ) EAGER INITIALIZATION .

(مناسب برنامه‌های چندنفری) THREAD-SAFE SINGLETION .

جمع بندی

الگوی **SINGLETON** یعنی فقط یه ریس تو سازمان داریم!

هرکس بخواه تصمیم بگیره یا گزارش بده، باید از طریق همون یه نفر انجامش بده.

اگه هر بخش برای خودش یه ریس داشت، کل شرکت به هم منجیخت! 😅

هم همین‌طوره: یه شئ مركزي وجود داره که همه بهش دسترسی د

ارن، ولی فقط یکی ازش ساخته منشه.

نتیجه: 🧠

نظم، کنترل، و هماهنگی در کل سیستم! ✓