

## Intro

### برنامه‌نویسی شیء‌گرا

در روزهای ابتدایی برنامه‌نویسی، توسعه‌دهندگان از روش‌های خطی و ساده برای نوشتن کد استفاده می‌کردند. این رویکرد در پروژه‌های کوچک و آزمایشی کارآمد بود، اما با گسترش سیستم‌های نرم‌افزاری، مشکلات جدی پدید آمد:

- **افزایش تکرار کد:** قطعه‌کدهای مشابه بارها در بخش‌های مختلف برنامه ظاهر می‌شدند. این مسئله نه تنها زمان توسعه را افزایش می‌داد، بلکه یافتن و اصلاح خطاها را نیز دشوار می‌کرد.
- **مدیریت سخت نیازهای پیچیده:** برخی برنامه‌ها، مانند محاسبات عددی، به راحتی در قالب توابع ریاضی پیاده‌سازی می‌شدند، اما مدل‌سازی سیستم‌هایی مانند یک فروشگاه آنلاین یا یک سامانه بانکی به روش‌های سنتی دشوار بود. برنامه‌نویسان نیاز داشتند که اجزای سیستم را به گونه‌ای ساختار دهند که تغییر و گسترش آن‌ها آسان باشد.

برای حل این مشکلات، روش‌های مختلفی پیشنهاد شد که در نهایت به شکل‌گیری **برنامه‌نویسی شیء‌گرا** انجامید.

### چالش

فرض کنید در حال توسعه نرم‌افزاری برای یک بیمارستان هستید. سیستم شما باید اطلاعات پزشکان، بیماران، پرستاران و اتاق‌ها را مدیریت کند. چگونه این مفاهیم را در قالب کد پیاده‌سازی می‌کنید؟ قبل از ادامه مطالعه، درباره این مسئله فکر کنید.

### شیء و کلاس: ستون‌های اصلی شیء‌گرایی

جهان پیرامون ما از اشیاء مختلفی تشکیل شده است. هر شیء دارای ویژگی‌هایی مانند نام، اندازه و رنگ است و می‌تواند اعمال خاصی انجام دهد. در برنامه‌نویسی شیء‌گرا، این مفهوم به صورت **شیء (Object)** پیاده‌سازی می‌شود.

برای تعریف اشیاء مشابه، از **کلاس (Class)** استفاده می‌کنیم. کلاس در واقع یک نقشه‌ی کلی است که مشخص می‌کند هر شیء چه ویژگی‌هایی دارد و چه کارهایی می‌تواند انجام دهد. سپس از این کلاس‌ها برای ایجاد اشیاء جدید **نمونه‌گیری (Instantiation)** انجام می‌شود.

## مثال

در پایتون، بسیاری از انواع داده‌ای که تاکنون استفاده کرده‌ایم، در حقیقت کلاس‌هایی از پیش تعریف‌شده هستند. به مثال زیر توجه کنید:

```
1 >>> patient_record = dict()
2 >>> patient_record
3 {}
4 >>> type(patient_record)
5 <class 'dict'>
6 >>> patient_record.update({"name": "Ali", "age": 30, "disease": "Flu"})
7 >>> patient_record
8 {'name': 'Ali', 'age': 30, 'disease': 'Flu'}
```

در این قطعه‌کد، `patient_record` یک نمونه از کلاس `dict` است که متدی به نام `update` دارد. ما از جزئیات داخلی این متد آگاه نیستیم، اما به‌سادگی از آن برای افزودن اطلاعات بیمار استفاده می‌کنیم.

برنامه‌نویسی شیء‌گرا به ما کمک می‌کند تا سیستم‌های پیچیده را به مجموعه‌ای از اشیاء کوچک‌تر تقسیم کنیم که هرکدام مسئولیت مشخصی دارند. با این روش، نه‌تنها از تکرار کد جلوگیری می‌شود، بلکه نگهداری و توسعه نرم‌افزار نیز آسان‌تر خواهد شد.

