



## ۱ بازسازی

تعاریف زیادی از "کد تمیز"<sup>۱</sup> وجود دارد؛ اما احتمالاً یکی از بهترین تعاریف‌ها متعلق به "بیارنه استراستروپ"<sup>۲</sup> خالق و توسعه‌دهنده‌ی زبان ++C است. وی ر تعریف خود از یک کد تمیز، دو مورد زیر را به عنوان معیارهای اساسی یک کد تمیز بر می‌شمارد:

- منطق و الگوریتم کد باید آن‌قدر واضح و قابل فهم باشد که اشکالات و نقص‌های جزئی نتوانند از چشم برنامه‌نویس و آزمونگر کد دور بمانند؛ ضمن این که وضوح کد باید به حدی بالا باشد که برنامه‌نویس را از نوشتن کامنت<sup>۳</sup> بی‌نیاز کند.
- کارایی<sup>۴</sup> برنامه نوشته‌شده باید در بهینه‌ترین<sup>۵</sup> شکل ممکن باشد تا بعدها برنامه‌نویس دیگری به بهانه‌ی بهینه‌سازی<sup>۶</sup> برنامه‌ی سابق با ایجاد تغییرات نادرست سبب نامنظم‌شدن و کثیف‌شدن کد نشود.

در عمل، در اکثر مواقع شما بعد از یک طراحی نسبتاً خوب و پیاده‌سازی آن، برای مدتی طولانی از آن کد برا هدف خود استفاده می‌کنید و در طول این مدت تغییرات و قابلیت‌های زیادی را به آن می‌افزایید.

پس از مدتی نه‌چندان طولانی، این تغییرات باعث می‌شوند که شما دیگر عملکرد کد را به‌وضوح متوجه نشوید و به تبع آن، توانایی تغییر و ارتقای کد را نیز از دست می‌دهید. همین زنجیره‌ی اتفاقات به ظاهر ساده در تاریخچه‌ی نسبتاً کوتاه توسعه‌ی نرم‌افزاری باعث نابودشدن شرکت‌های بسیاری در این عرصه شده است.

حال با توجه به خطرات و مشکلاتی که یک کد کثیف به همراه دارد، باید راه‌حلی برای رفع کثیف‌بودن کد و جلوگیری از ایجاد آن ارائه دهیم. شما در این تمرین کامپیوتری با روند بازسازی<sup>۷</sup> کد آشنا می‌شوید.

بازسازی عملیاتی است که در طی آن ساختار یک نرم افزار به صورتی تغییر و بهبود می‌یابد که بدون از دست‌رفتن کارایی‌ها و تغییر رابط کاربری<sup>۸</sup> برنامه، ساختار درونی کد به طرز قابل توجهی تمیزتر می‌شود.

بنیادی‌ترین مفهوم یاری‌کننده‌ی یک برنامه‌نویس در طی عملیات بازسازی شناخت عناصری است که باعث کثیف شدن کدها می‌شوند و به اصطلاح به آن‌ها Smells Code گفته می‌شود.

وظیفه‌ی شما در این تمرین بازسازی کد خودتان در تمرین اول درس است؛ بنابراین **خوانایی و تمیزبودن کد** در این تمرین بیشترین اهمیت را دارد. در ادامه توضیحاتی درباره‌ی بازسازی کد ارائه می‌شود. پیشنهاد می‌کنیم که ابتدا تا پایان صورت تمرین را مطالعه کنید و سپس شروع به بازسازی کد خود کنید.

<sup>1</sup>Clean Code

<sup>2</sup>Bjarne Stroustrup

<sup>3</sup>Comment

<sup>4</sup>Performance

<sup>5</sup>Optimal

<sup>6</sup>Optimization

<sup>7</sup>Refactoring

<sup>8</sup>Interface

## ۲ کد تمیز

عواملی در کد وجود دارند که ممکن است باعث کثیف شدن آن شوند؛ در ادامه برخی از این عوامل توضیح داده شده‌اند. توجه کنید که در انتهای این تمرین نمره شما فقط بر اساس عوامل زیر سنجیده می‌شود و به ازای هر یک از موارد زیر که در کد شما وجود داشته باشد نمره‌ی شما کاسته خواهد شد. ساختار کلی کد و طراحی شما نباید تغییر کند و فقط ساختار درونی آن می‌تواند تغییر کند. این عوامل خلاصه‌ای از کتاب Code Clean هستند و عبارت انتهای هر عامل فصل و شماره‌ای آن عامل را در کتاب نشان می‌دهد.

### ۱.۲ نام‌گذاری (فصل ۲)

○

### ۲.۲ توابع (فصل ۳)

○

### ۳.۲ فرمتینگ

○

## ۴.۲ Smells and Heuristics (فصل ۱۷)

### ۱.۴.۲ کامنت‌ها

○ در این تمرین کامنت‌گذاری به هیچ نحوی قابل قبول نیست.

### ۲.۴.۲ کامنت‌ها

○ در این تمرین کامنت‌گذاری به هیچ نحوی قابل قبول نیست.