

# Các vấn đề trong bài toán

## "99 Bottles of Beer"

Xem xét các vấn đề bất cập cho **99 Bottles of Beer on the Wall**

Bài toán này dựa trên một bài hát dân gian tiếng Anh có tên là **"99 Bottles of Beer on the Wall"**. Nội dung bài hát (và cũng là nội dung của bài toán) như sau:

- Bắt đầu với 99 chai bia trên tường.
- Mỗi lần, bạn hát một đoạn như sau:

99 bottles of beer on the wall, 99 bottles of beer.  
Take one down and pass it around, 98 bottles of beer on the wall.

- Sau đó lặp lại với 98, 97, ... cho đến khi còn 0 chai:

No more bottles of beer on the wall, no more bottles of beer.  
Go to the store and buy some more, 99 bottles of beer on the wall.

Giải pháp đầu tiên

```
class BottlesOfBeer:
    def song(self):
        return self.verses(99, 0)

    def verses(self, hi, lo):
        return '\n'.join(self.verse(n) for n in range(hi, lo - 1, -1))

    def verse(self, n):
        return (
            f'{"No more" if n == 0 else n} bottle{"s" if n != 1 else ""}'
            f' of beer on the wall, '
            f'{"no more" if n == 0 else n} bottle{"s" if n != 1 else ""} of beer.\n' +
            (f'Go to the store and buy some more, ' if n == 0 else f'Take {"one" if n != 1 else "it"} down and pass it
            around, ') +
            f'{"99" if n - 1 < 0 else ("no more" if n - 1 == 0 else n - 1)} bottle{"s" if n - 1 != 1 else ""}'
        )
```

```
f' of beer on the wall.\n'
```

```
)
```

Đoạn code trên thực hiện một thủ thuật gọn gàng. Nó có thể cô đọng đến mức không ai có thể hiểu được trong khi vẫn giữ lại rất nhiều sự trùng lặp. Mã này khó hiểu vì nó không nhất quán và trùng lặp, và vì nó chứa các khái niệm ẩn mà nó không đặt tên. Và nhúng rất nhiều logic vào chuỗi câu thơ.

## Sự nhất quán(Consistency)

Các toán tử ba ngôi đang gây nhầm lẫn. Phần lớn sử dụng dạng đơn giản, như ở dòng 10:

```
"No more" if n == 0 else n
```

Và một số lại lồng toán tử ba ngôi bên trong một toán tử ba ngôi khác như

```
f'{"99" if n - 1 < 0 else ("no more" if n - 1 == 0 else n - 1)} bottle{"s" if n - 1 != 1 else ""}'
```

Việc lồng nhiều toán tử ba ngôi như vậy khiến mã khó đọc hơn đối với con người; kiểu viết này làm **tăng chi phí bảo trì** mà **không mang lại lợi ích gì**.

## Lặp lại (Duplication)

Đoạn code này lặp lại cả dữ liệu và logic. Việc lặp lại chuỗi như `"of beer"` hay `"on the wall"` thì dễ thấy và dễ hiểu, nhưng lặp lại logic thì khó hiểu hơn nhiều — đặc biệt khi nó được nhúng bên trong chuỗi.

Ví dụ, logic để thêm chữ `"s"` vào `"bottle"` xuất hiện tới **3 lần**. Hai chỗ đầu giống hệt nhau:

```
"s" if n != 1 else ""
```

Nhưng đến chỗ thứ ba thì lại khác đi chút:

```
"s" if n - 1 != 1 else ""
```

Sự lặp lại này cho thấy: **có những ý tưởng tiềm ẩn trong code** (như việc chia số nhiều, hay phân biệt giữa `n` và `n - 1`) nhưng lại **chưa được đặt tên rõ ràng**. Code như vậy khiến người đọc phải tự đoán xem từng phần đang thực sự làm gì.

## Tên biến và hàm (Names)

Điều dễ nhận ra nhất trong hàm `verse` là: **hầu như không có cái tên nào cả**. Mọi logic đều được nhúng trực tiếp vào trong chuỗi string.

Mỗi đoạn logic đó đều mang một ý nghĩa nào đó — nhưng lại **không được đặt tên**, khiến bạn phải tự hình dung trong đầu từng phần đang làm gì.

Nếu muốn code dễ hiểu hơn, đừng bắt người đọc phải đoán. Thay vì nhét logic vào chuỗi, hãy tách nó ra thành các phương thức có tên rõ ràng. Khi đó, `verse` chỉ cần gọi các hàm đó để lấp đầy nội dung — đơn giản, dễ đọc và dễ bảo trì.