Challenge

Your Turn!

```
-   Write an `@Audit`  class decorator that logs when a drone instance is created.
```

- Apply it to a new `AuditDrone` class.
- Implement a parameter decorator `@ValidateEmail` that could be used to mark a method parameter for email validation.

Challenge

Your Turn!

- Implement a `PromotionSystem` observer that announces special offers to customers when a drink is served.
- Add it to the `DrinkOrder` notification list and test it.
- Create a Factory that produces different types of `Report` objects (`PDFReport` , `HTMLReport`) based on a string input.

Challenge

- Implement a new gateway class `BankTransferGateway` that logs payment processing.
- Use it with `PaymentProcessor` to process a payment.
- Write a mock gateway that simulates failure (`return false`) and test error handling.
- Define an interface `EmailService` and implement two classes: `SmtpEmailService` and `MockEmailService` . Inject the desired service into a `NotificationSender` class using constructor injection.