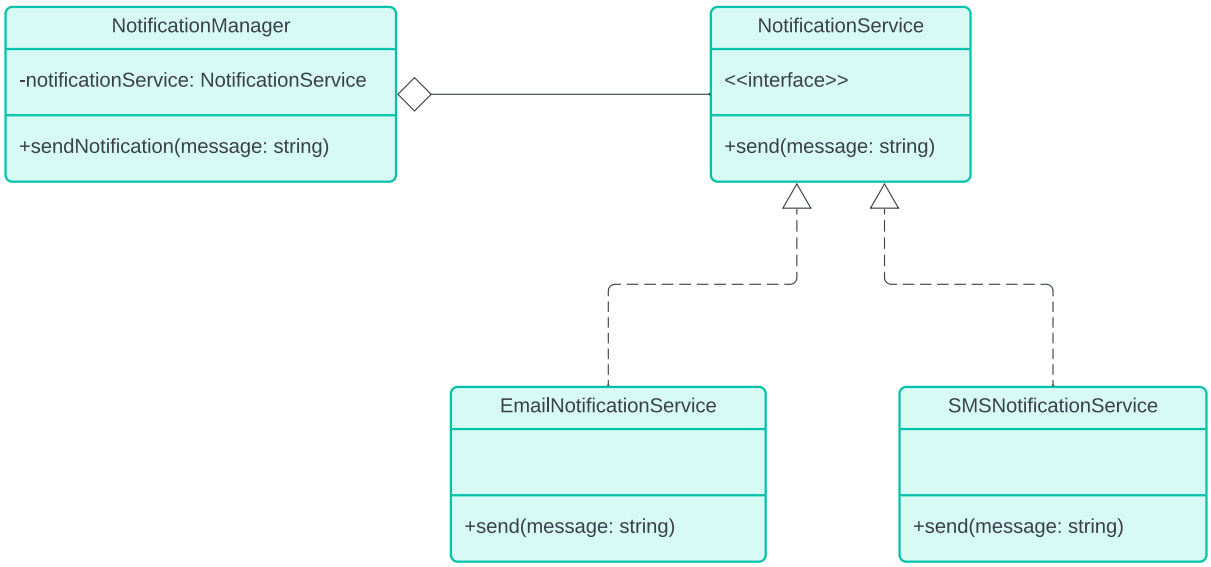
Exercise: Implementing a Notification Service

Your task is to design a flexible notification system for an application. The system should be able to send notifications through different channels, such as email and SMS. You need to implement a solution that allows swapping the notification methods without changing the core application logic.

This is a simulation, so no actual emails or SMS messages will be sent—this exercise focuses purely on the design and usage of Spring's IoC (Inversion of Control) container to manage dependencies.

Class Diagram Overview

Before diving into the implementation, here is a class diagram that represents the final design of your solution. It shows how the various components, such as the `NotificationService` interface, its implementations (`EmailNotificationService` and `SMSNotificationService`), and the `NotificationManager` interact with each other.



Steps

1. **Define the `NotificationService` interface:**
    - The `NotificationService` should have one method, `send(String message)`, to handle sending notifications
2. **Implement multiple notification methods:**
    - Create two classes: one for sending email notifications and one for SMS notifications. Both should implement the `NotificationService` interface.
    - Use a `print` statement in each implementation to simulate sending a notification, for example:
        - Sending email: [message]
        - Sending SMS: [message]
3. **Design a `NotificationManager` class:**
    - The `NotificationManager` should use the `NotificationService` to send a notification. The exact method of sending (email or SMS) should depend on which implementation of `NotificationService` is provided.
4. **Test the notification system:**
    - In your main method, use Spring's `ApplicationContext` to get an instance of `NotificationManager`.
    - Call the `sendNotification("Hello, this is a test message!")` method.
    - The system should print out the correct notification method and message depending on which service (email or SMS) is injected.