**TDD Stub Examples**

**Summary of Stubbing Usage**  
For every example, a hand-crafted stub class replaces real external dependencies, providing controlled, test-specific behavior. This enables deterministic, focused logic testing of the service in JUnit 5, where the unit interacts with external dependencies whose responses are simulated by the stub object. Each test method's comments explain its precise role and the stub's use within.

**Example 1: User Notification Filtering**

**Task Requirement:**  
Implement a NotificationService that filters notifications for a given user based on their user settings. The filtering logic should skip notifications marked as "Do Not Disturb" in settings. User settings come from a dependency UserSettingsProvider. Create unit tests using a hand-crafted stub for UserSettingsProvider.

**Test Class**

import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
import java.util.\*;  
  
class NotificationServiceTest {  
 // Purpose: Verifies filtering out "DND" notifications according to stubbed settings.  
 @Test  
 void filtersNotificationsBasedOnUserSettings() {  
 // Stub simulates user settings (with "Do Not Disturb" enabled).  
 UserSettingsProviderStub stub = new UserSettingsProviderStub(true);  
  
 NotificationService service = new NotificationService(stub);  
  
 List<String> allNotifications = Arrays.asList("Meeting at 10am", "DND: Lunch Break", "DND: Focus Time", "General Update");  
 List<String> filtered = service.filterNotifications("alice", allNotifications);  
  
 assertEquals(Arrays.asList("Meeting at 10am", "General Update"), filtered,   
 "Should filter out notifications marked DND when 'Do Not Disturb' is enabled");  
 }  
  
 // Purpose: Ensures all notifications are shown if "Do Not Disturb" is off in stub.  
 @Test  
 void doesNotFilterIfDndOff() {  
 UserSettingsProviderStub stub = new UserSettingsProviderStub(false);  
 NotificationService service = new NotificationService(stub);  
  
 List<String> allNotifications = Arrays.asList("Meeting at 10am", "DND: Lunch Break");  
 List<String> filtered = service.filterNotifications("alice", allNotifications);  
  
 assertEquals(allNotifications, filtered, "All notifications should be shown when DND is off");  
 }  
}  
  
// Stub Explanation:  
// 'UserSettingsProviderStub' is used instead of the real settings provider.   
// It's a simple Java class that allows the test to simulate different user settings.  
// Where: Used directly in test methods, injected into service.  
// How: Constructor parameter determines simulated DND setting.  
  
// STUB: Hardcoded to simulate settings  
class UserSettingsProviderStub implements UserSettingsProvider {  
 private final boolean doNotDisturb;  
  
 public UserSettingsProviderStub(boolean doNotDisturb) {  
 this.doNotDisturb = doNotDisturb;  
 }  
 @Override  
 public boolean isDoNotDisturbEnabled(String username) {  
 return doNotDisturb;  
 }  
}

**Service and Stub Implementation**

// Service to be tested  
class NotificationService {  
 private final UserSettingsProvider settingsProvider;  
  
 public NotificationService(UserSettingsProvider settingsProvider) {  
 this.settingsProvider = settingsProvider;  
 }  
  
 public List<String> filterNotifications(String username, List<String> notifications) {  
 boolean doNotDisturb = settingsProvider.isDoNotDisturbEnabled(username);  
 if (!doNotDisturb) {  
 return notifications;  
 }  
 List<String> filtered = new ArrayList<>();  
 for (String note : notifications) {  
 if (!note.startsWith("DND:")) {  
 filtered.add(note);  
 }  
 }  
 return filtered;  
 }  
}  
  
// Interface for collaboration  
interface UserSettingsProvider {  
 boolean isDoNotDisturbEnabled(String username);  
}