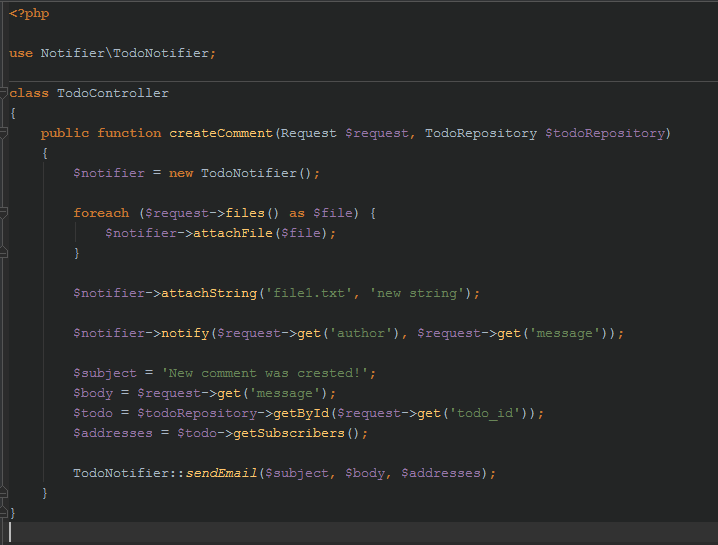
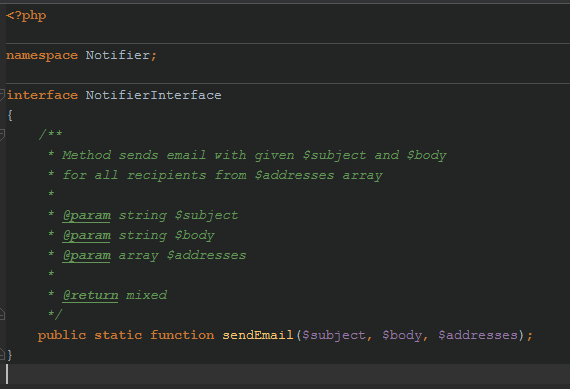
1. **Для чего предназначен TodoNotifier класс?**

Класс TodoNotifier является частью некого трекера задач. Он выполняет следующие функции:

1. рассылка электронных писем любым наборам адресатов с любым содержимым;
2. создание новых комментариев к задаче;
3. сохранение файлов для комментариев в папке для временных файлов на сервере.
4. **Дайте пример того, как этот класс может быть использован**



1. **Как бы мог выглядеть NotifierInterface? Какие методы он может объявлять?**



Я намеренно не включил в интерфейс метод notify, хотя по названию это напрашивается, потому что в данном классе метод notify выполняет совершенно не ту функцию, которая должна быть у класса Notifier.

1. **Каковы потенциальные проблемы с этим кодом? Что подозрительно? Как можно их исправить?**

Мои замечания к коду:

1. *Класс нарушает принцип единственной обязанности.*

TodoNotifier – подразумевается, что этот класс отвечает за уведомления подписчикам задачи на ее различные события. По факту, через метод sendMail этого класса можно посылать любые сообщения любым адресам, к задаче этот метод ни как не привязан. Метод notify же вообще занимается совершенно другой работой – он сохраняет новый комментарий к задаче.

Так же этот класс умеет сохранять новые файлы к комментариям и изменять существующие (методы attachString и attachFile).

1. *Раскрытие лишних деталей реализации.*

Все 3 свойства объявлены публичными, лучше сделать их приватными и добавить геттеры-сеттеры, если это необходимо. Задавать начальное значение null свойствам не нужно.

1. *Использование глобальных переменных.*

Невозможно быть уверенным, что в глобальной переменной хранится именно то, чего ожидает программист. Передавать smtpHost каждый раз при вызове метода с его параметрами также не хочется – это лишние детали реализации, которые должны быть скрыты для сущностей высшего уровня, использующих этот метод. Так как smtp host меняется крайне редко, хорошей практикой будет поместить его в конфигурационный файл и брать оттуда при выполнении метода sendMail.

1. *Зависимость от класса phpmailer.*

В методе sendMail присутствует явная зависимость от класса phpmailer. Это, как минимум, мешает покрытию метода тестами.

1. *баг в строке*

$mailer->Subject = $subject ? : 'Email notification';

Либо дописываем $subject после знака «?», либо убираем пробел между «?» и «:» (оператор Элвиса).

1. *Несоблюдение некоторых правил psr.*

Согласно правилам PSR нужно обрамлять тело цикла или условного выражения в фигурные скобки всегда, даже если тело состоит из одной строки.

1. *Произвольный порядок методов.*

Хороший класс должен читаться как газетная статья. В таком коде идут сначала публичные методы, а приватные строятся сразу за тем публичным методом, который его первым использует. В рассматриваемом классе все перемешано.

1. *Потенциальная возможность получить ошибку «undefined index».*

Метод getPostTask – при взятии из конфига параметров есть риск получить ошибку undefined index. Перед взятием стоит проверить существование этих индексов в массиве.

1. *Явные зависимости от классов TodoService и TodoServiceClient.*

Такие зависимости мешают покрытию кода тестами, а также могу вызвать трудности при изменении, например, конструкторов этих классов. Хорошим решением будет вынести TodoService в параметры конструктора класса.

1. *Несогласованность в терминологии.*

Где-то используется слово “todo”, где-то “task”. Нужно выбрать один термин и везде его придерживаться. Лучше бы использовать “task”, так как todo зарезервировано IDE для особых комментариев.

1. *Метод attachString перезаписывает содержимое файлов.*

Название можно перевести как «прикрепить строку», но по факту метод не прикрепляет переданную строку к требуемому файлу, он перезаписывает этот файл с этой строкой (метод file\_put\_contents). Возможно, стоит добавить флаг FILE\_APPEND.

1. *Утечка памяти через папку tmp.*

Свойство tmpFiles использется для удаления файлов при создании комментария в методе notify. Но файлы, добавляемые через attachFile в этот массив не сохраняются, сохраняются только изменяемые в методе attachString файлы. Значит, некоторые файлы будут скапливаться в папке tmp, пока свободное место не кончится.

1. *Методe createAuthrntication в классе TodoNotifier не место.*
2. *Использование устаревших функций mysql\_connect и mysql\_query. Функции mysql\_fetch\_result вообще не существует.*

Лучше воспользоваться MySQLI или PDO.

1. *Опасность sql-инъекции.*

Нужно использовать PDO::prepare.

1. *Невыразительные названия переменных в createAuthrntication.*
2. *Используется текущее подключение к БД(которого может и не быть), которое не закрывается после исполнения метода.*

Раз mysql\_connect вызывается без аргументов, значит оно использует уже существующее подключение к БД. В конце метода не вызывается mysql\_close. Это не лучшее решение, программист не может контролировать, какое именно подключение к БД будет использоваться при выполнении этого метода.

1. *Баг в методе notify – strpos может вернуть 0 как позицию вхождения строки “test”, что расценится условным оператором как false.*

Необходимо использовать строгое сравнение (!== false).

1. **Комментарии к рефакторингу**

Я подготовил две версии. Первую версию я делал в условиях того, что я не имею право менять внешнее поведение класса TodoNotifier. Я не стал убирать из него методы, которые в нем быть не должны, не менял видимость свойств, не устанавливал правила строгой типизации и не вызывал никаких новых исключений помимо тех, что уже есть. Просто постарался закрыть некоторые уязвимости и улучшить читаемость кода.

Вторая версия – то, как я бы поменял код, если бы у меня была возможность найти и исправить все места, где класс TodoNotifier используется. Я переименовал класс в TaskNotifier, в нем есть один метод – notifyAboutNewComment. В дальнейшем, туда можно добавить и другие методы уведомлений по задаче (например, notifyAboutStatusChange). Метод использует класс Notifier через интерфейс NotifierInterface. У этого класса есть метод sendEmail. Этот класс, в дальнейшем, может использоваться и в других класса-уведомителях (например, NewsNotifier). Также есть возможность добавить в класс Notifier методы для передачи уведомлений по другим каналам (sendSms, sendTelegram и т.д.). Все методы с созданием комментариев и работой с файлами я бы раскидал по другим, более подходящим, сущностям.

Я работал исходя из предположения, что код будет выполняться на версии не ниже php7.