

Телеграф

Цели практической работы

- Научиться работать с классами и объектами на практике.
- Реализовать методы для «Телеграфа».
- Закрепить навыки работы с Git.

Для выполнения этого задания нужно иметь базовые навыки работы в Git. Если вы ещё не познакомились с ним, сейчас самое время изучить курс «Система контроля версий Git».

Создайте пустой публичный репозиторий в GitLab для дальнейшей работы в нём. Чтобы вам не было слишком сложно в самом начале, мы будем хранить тексты в массиве. В следующих занятиях мы заменим массив на базу данных.

Что нужно сделать

Мы продолжаем разрабатывать наш аналог «Телеграфа». Опишем класс для описания сущности «текст». Добавим в класс поля и методы, необходимые для хранения, редактирования, удаления и вывода текста на экран.

Для успешной реализации программы вам потребуется познакомиться со встроенными функциями PHP для работы с файловой системой:

- функция `file_put_contents` — для записи текста в файл;
- функция `file_get_contents` — для чтения текста из файла.
- функция `file_exists` — для проверки, существует ли файл с заданным именем.

1. Создайте класс **TelegraphText**.

2. Добавьте поля `$text`, `$title`, `$author`, `$published` и `$slug`.

- **`$title`** — заголовок текста;
- **`$text`** — текст;
- **`$author`** — имя автора;
- **`$published`** — дата и время последнего изменения текста;
- **`$slug`** — уникальное имя для файла, в котором будет храниться текст (например, `test_text_file`).

3. Опишите конструктор для класса **TelegraphText**. При создании объекта данного класса будет передаваться имя автора и уникальное имя файла. В конструкторе задайте соответствующие значения полям **`$author`**, **`$slug`** и **`$published`**. Присвойте текущее время и дату полю **`$published`**.

- 4.Опишите метод **storeText** для записи текста в файл. Создайте ассоциированный массив с ключами **text**, **title**, **author**, **published** и присвойте соответствующим элементам значения полей **\$title**, **\$text**,**\$author** и **\$published**. Сериализуйте массив с помощью встроенной функции **serialize** и запишите его в файл. Имя файла хранится в поле **\$slug**.
- 5.Опишите метод **loadText** для загрузки текста из файла. Имя файла хранится в поле **\$slug**. По аналогии с методом **storeText** десериализуйте содержимое файла (если файл не пуст) в массив, а затем присвойте полям **\$title**, **\$text**, **\$author** и **\$published** значения соответствующих элементов массива. Метод должен возвращать **\$text**.
- 6.Опишите метод **editText** для редактирования текста. Метод должен принимать текстовые параметры: заголовок и текст. Метод назначает новые значения полям **\$text** и **\$title**.
- 7.Создайте объект класса **TelegraphText**, передав необходимые для конструктора параметры.
- 8.Вызовите методы **editText**, а затем **storeText**.
- 9.Вызовите метод **loadText** и выведите значение, которое вернёт этот метод, на экран.
- 10.Если всё работает правильно, сделайте коммит ваших изменений с помощью **Git** и отправьте коммит в репозиторий.

Советы и рекомендации

- Старайтесь писать в **commit message** то, что отражает смысл вашего кода.
- Обращайте внимание на доступ к полям объекта внутри методов (используйте **\$this->**).

Критерии оценки

Принято:

- в задании используются указанные инструменты и соблюдены условия;
- код корректно отформатирован по стандартам программирования на PHP;
- скрипт запускается, выводит различные данные на экран, не вызывает ошибок.

На доработку: выполнены не все обязательные пункты задания или есть ошибки.

Как отправить работу на проверку

Отправьте коммит, содержащий код задания, на ветку **master** в вашем репозитории и пришлите его URL (URL MergeRequest'a) через форму ниже. Репозиторий должен быть **public**.