

# Профессиональный практикум программной инженерии

Лабораторная работа №1 «Изучение базовых принципов работы с Git»

---

## Цель лабораторной работы

Познакомиться с основами использования системы контроля версий Git.

## Ход работы

1. Зарегистрироваться на github.com.
2. Получить доступ к проекту у преподавателя.
3. Создать каталог проекта.
4. Создать базовую структуру репозитория.
5. Начать разработку кода.
6. Сделать не менее пяти фиксаций, касающихся добавления файлов и редактирования кода.

## Содержание отчета

1. Титульный лист.
2. Вывод команды `git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short`
3. Вывод команды `git diff` для одной из ревизий, касающейся редактирования кода.

# Профессиональный практикум программной инженерии

Лабораторная работа №2 «Работа с ветками в Git»

---

## Цель лабораторной работы

Познакомиться с основами использования веток в системе контроля версий Git.

## Ход работы

1. На основе master создать ветку lab2.
2. Переключить рабочую копию на созданную ветку.
3. Сделать не менее пяти фиксаций, касающихся добавления файлов и редактирования кода.
4. Переключить рабочую копию на master.
5. Влить все изменения из ветки с помощью команды merge.
6. Удалить ветку.

## Содержание отчета

1. Титульный лист.
2. Вывод команды `git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short,` касающийся работы с веткой

# Профессиональный практикум программной инженерии

## Лабораторная работа №3 «Работа с параллельными ветками»

---

### Цель лабораторной работы

Научиться работать с параллельными ветками в Git.

### Ход работы

1. Создать ветку lab3.1  
`git checkout -b lab3.1`
2. Внести не менее трех коммитов в ветку 3.1
3. Проверить лог ревизии  
`git log --pretty=format:@"%h %ad | %s%d [%an]" --graph --date=short`
4. На основе master создать ветку lab3.2  
`git checkout master`  
`git checkout -b lab3.2`
5. Внести не менее трех коммитов в ветку 3.2
- 6.
7. Проверить лог ревизии  
`git log --pretty=format:@"%h %ad | %s%d [%an]" --graph --date=short`
8. Переключиться на ветку master  
`git checkout master`
9. Влить обе ветки  
`git merge lab3.1`  
`git merge lab3.2`
10. Посмотреть лог ревизии  
`git log --pretty=format:@"%h %ad | %s%d [%an]" --graph --date=short`
11. Удалить обе ветки  
`git branch -D lab3.1`  
`git branch -D lab3.2`

## **Содержание отчета**

1. Титульный лист.
2. Вывод команды `git log` для ветки `lab3.1`
3. Вывод команды `git log` для ветки `lab3.2`
4. Вывод команды `git log` для `trunk` после вливания всех веток

# Профессиональный практикум программной инженерии

Лабораторная работа №4 «Разрешение конфликтов объединения веток»

---

## Цель лабораторной работы

Научиться разрешать в Git конфликтные ситуации, возникающие после объединения веток.

## Ход работы

1. На основе master создать ветку lab4
2. Переключиться на эту ветку
3. Внести изменения и сделать и зафиксировать их в репозитории
4. Переключиться на master
5. Внести другие изменения в тот же файл в те же строки и сделать коммит
6. Влить ветку lab4 и получить конфликт
7. Разрешить конфликт и закоммитить результирующий код
8. Удалить ветку lab4

## Содержание отчета

1. Титульный лист.
2. Вывод команды `git status`, демонстрирующую конфликтную ситуацию
3. Содержимое конфликтующего файла (достаточно только строк, в которых произошел конфликт)

# Профессиональный практикум программной инженерии

Лабораторная работа №5 «Создание самодокументирующегося кода»

---

## Цель лабораторной работы

Научиться добавлять в программный код специальным образом оформленные докблок-комментарии, для последующей автоматической генерации API Reference.

## Ход работы

1. Во всех файлах с исходным кодом добавить докблоки ко всем классам, методам, свойствам и функциям.
2. Закоммитить изменения в репозиторий.

## Примеры докблоков

Докблок для класса:

```
/**
 * Транспортное средство
 *
 * Класс описывает абстрактное транспортное
 * средство с базовыми характеристиками.
 *
 * @author      John Doe
 * @version     1.0.1
 * @copyright   GNU Public License
 * @todo        Реализовать все методы
 */
class Vehicle {}
```

Докблок для свойства:

```
/**
 * Производитель транспортного средства
 *
 * Используем только простое символьное
 * имя производителя. Если будет
 * необходима детализация, создадим
 * класс Brand
 *
 * @var         string    $brand
 */
public $brand;
```

Докблок для метода/функции:

```
/**
 * Работа со свойством {@link $brand}
 *
 * Если аргумент определен, то устанавливается новое
 * значение свойства и возвращается указатель на
 * объект, иначе возвращается текущее значение
 * свойства.
 *
 * @param string $brand Производитель
 * @return mixed Возвращает текущее значение
 *               свойства или указатель на
 *               объект
 */
public function brand($brand=NULL);
```

## Содержание отчета

Лабораторная работа не подразумевает наличие бумажного отчета. Код проверяется непосредственно в репозитории.

# Профессиональный практикум программной инженерии

Лабораторная работа №6 «Создание руководства программиста»

---

## **Цель лабораторной работы**

Познакомиться с форматом оформления документации DocBook.

## **Ход работы**

1. Установить один из XML-редакторов (по желанию).
2. Оформить краткое описание разработанного продукта (не менее двух абзацев).
3. Рассмотреть с примерами кода не менее трех сценариев использования вашего приложения.
4. Проверить корректность созданного DocBook файла с помощью средств редактора или одного из онлайн валидаторов, например <http://www.mashupsoft.com/docbooks/validator/>.

## **Содержание отчета**

1. Титульный лист.
2. Файл DocBlock.



# Професіональний практикум програмної інженерії

## Список літератури

---

1. <https://try.github.io/levels/1/challenges/1>
2. JavaScript – полезны материалы / <http://habrahabr.ru/post/152048/>
3. Сборник статей о разработке программных продуктов / Дж. Спольски . – <http://local.joelonsoftware.com/wiki/Russian>
4. Сайт сообщества разработчиков / <http://habrahabr.ru/>
5. Право інтелектуальної власності: Академ. курс: Підручник для студентів вищих навч. закладів / За ред. О.А.Підпригори: -К.: Концерн „Видавничий Дім „Ін Юре””, 2004. -672 с. Антонов В.М.
6. Інтелектуальна власність і комп’ютерне авторське право. –К.: КНТ, 2005. -520 с.
7. Официальный сайт сообщества GNU / <http://www.gnu.org/>