GIT

1. Выбираем хостинг для репозиториев <https://github.com>
2. Ветки:

- Новые ветки разработки создаются только от master ветки проекта. Ветвления типа master – ветка – ветка запрещены.

- Каждая ветка разработки должна быть привязана к тикету в задачнике, то есть её имя должно быть идентично имени тикета (например, если тикет называется DEV-2023, то имя ветки будет аналогичным).

- Одну задачу можно дробить на подзадачи. Тогда ветки нужно создавать с номерным префиксом (например, DEV-2023-N, где N – порядковый номер ветки с тем же тикетом).

Порядок:

- После того, как будет выполнен Merge или Pull request, следует удалять source branch.

- Коммиты для себя следует “свернуть” до одного. Комментарий коммита должен иметь в начале имя ветки к которой он относится (например, DEV-2023-4 добавил поле Captcha в форму обратной связи).

1. Настройка среды разработки:

* <https://github.com/leon-500/sf_praktikum2/blob/master/.gitconfig> – ссылка на конфиг
* git config --global user.name "name" - имя, которое будет использоваться при создании коммита
* git config --global user.email “[email@mail.ru](mailto:email@mail.ru)” - email, который будет использоваться при создании коммита
* git config --global core.editor vim - редактор по умолчанию
* git config --global alias.history 'log --graph' - алиас history для команды log –graph
* git clone <https://github.com/leon-500/sf_praktikum2.git> - клонируем содержимое из удалённого репозитория на локальный
* git checkout -b new\_branch\_name – создание новой ветки и переход в неё
* git add somefile или git add . – добавление отдельных файлов или всех файлов в область подготовленных файлов
* git commit -m “comment” - внесение изменений однострочным сообщением
* git checkout master – переключение в основную ветку master
* git merge existing\_branch\_name – объединение указанной ветки с основной
* git push origin branch\_name - отправка изменений в удалённый репозиторий