

# **Шаблон отчёта по лабораторной работе**

**Простейший вариант**

Лупупа Чилеше

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
3.0.1	Основные понятия Git . . . . .	7
3.0.2	Основные команды Git . . . . .	8
<b>4</b>	<b>Выводы</b>	<b>23</b>
	<b>Список литературы</b>	<b>24</b>

## Список иллюстраций

3.1	Enable the copr repository . . . . .	9
3.2	Install gitflow . . . . .	9
3.3	Node.js . . . . .	10
3.4	pnpm . . . . .	11
3.5	pnpm . . . . .	11
3.6	pnpm setup . . . . .	12
3.7	pnpm add -g commitizen . . . . .	12
3.8	pnpm add -g standard-changelog . . . . .	12
3.9	git-extended . . . . .	13
3.10	git commit . . . . .	13
3.11	Конфигурация для пакетов . . . . .	14
3.12	заполнение параметров пакета. . . . .	15
3.13	git flow init . . . . .	15
3.14	git branch . . . . .	16
3.15	git push -all . . . . .	16
3.16	git branch -set-upstream-to=origin/develop develop . . . . .	16
3.17	git flow release start 1.0.0 . . . . .	16
3.18	standard-changelog -first-release . . . . .	17
3.19	git add . . . . .	17
3.20	git flow release finish 1.0.0 . . . . .	17
3.21	git push -all . . . . .	18
3.22	git push -tags . . . . .	18
3.23	gh release create v1.0.0 -F CHANGELOG.md . . . . .	18
3.24	git flow feature start feature_branch . . . . .	19
3.25	git flow feature finish feature_branch . . . . .	19
3.26	git flow release start 1.2.3 . . . . .	20
3.27	standard-changelog . . . . .	20
3.28	git add CHANGELOG.md . . . . .	20
3.29	git flow release finish 1.2.3 . . . . .	21
3.30	git push -all . . . . .	21
3.31	git push -tags . . . . .	22
3.32	gh release create v1.2.3 -F CHANGELOG.md . . . . .	22

## **Список таблиц**

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

## 2 Задание

1. Установка программного обеспечения
2. Установка Node.js
3. Настройка Node.js
4. Общепринятые коммиты

## 3 Теоретическое введение

Git — это распределенная система контроля версий, которая произвела революцию в способах управления и совместной работы разработчиков над программными проектами. Это позволяет нескольким людям одновременно работать над одной базой кода, что упрощает отслеживание изменений, управление версиями и совместную работу в распределенной среде. Чтобы эффективно работать с репозиториями Git, необходимо понимать его основные концепции, команды и рабочие процессы. Целью этого теоретического введения является предоставление учащимся фундаментальных знаний, необходимых для уверенной и эффективной навигации по репозиториям Git.

### 3.0.1 Основные понятия Git

**Репозитории** : сердце любого проекта Git. Репозиторий содержит все файлы, историю, ветки и коммиты, связанные с проектом. **Коммиты** : снимки состояния проекта в определенный момент времени. Каждый коммит уникально идентифицируется и содержит информацию о внесенных изменениях и авторе этих изменений.

**Ветки** : параллельные версии проекта, которые позволяют разработчикам одновременно работать над различными функциями или исправлениями, не затрагивая основную базу кода.

**Слияния** : процесс интеграции изменений из одной ветки в другую, обычно в главную ветку.

**Клонирование** : действие по созданию локальной копии удаленного репозитория для работы.

**Извлечение** : обновление локального репозитория изменениями из удаленного репозитория.

**Pushing** : отправка локальных коммитов в удаленный репозиторий.

### 3.0.2 Основные команды Git

- `git init`: инициализирует новый репозиторий Git.
- `git clone`: создает локальную копию удаленного репозитория.
- `git add`: добавляет изменения в промежуточную область, подготавливая их к фиксации.
- `git commit`: сохраняет изменения в локальном репозитории.
- `git status`: показывает статус изменений в рабочем каталоге и промежуточной области.
- `git ветка`: перечисляет, создает или удаляет ветки.
- `git checkout`: переключает между ветками или коммитами.
- `git merge`: объединяет изменения из одной ветки в другую.
- `git pull`: извлекает изменения из удаленного репозитория и объединяет их в текущую ветку.
- `git push`: отправляет локальные коммиты в удаленный репозиторий. # Выполнение лабораторной работы

#### 1. Установка программного обеспечения:

Установка git-flow

**Enable the copr repository** `dnf copr enable elegos/gitflow` **Install gitflow** `dnf install gitflow`



```
[lchileshe@lchileshe ~]$ sudo dnf copr enable elegos/gitflow
[sudo] password for lchileshe:
Enabling a Copr repository. Please note that this repository is not part
of the main distribution, and quality may vary.

The Fedora Project does not exercise any power over the contents of
this repository beyond the rules outlined in the Copr FAQ at
https://docs.pagure.org/copr.copr/user\_documentation.html#what-i-can-bu
ild-in-copr>,
and packages are not held to any quality or security level.

Please do not file bug reports about these packages in Fedora
Bugzilla. In case of problems, contact the owner of this repository.

Do you really want to enable copr.fedorainfracloud.org/elegos/gitflow? [
y/N]: Y
Repository successfully enabled.
```

Рис. 3.1: Enable the copr repository

```
[lchileshe@lchileshe ~]$ sudo dnf install gitflow
Copr repo for gitflow owned by elegos 1.4 kB/s | 1.5 kB 00:01
Fedora 39 - x86_64 - Updates 7.9 kB/s | 21 kB 00:02
Fedora 39 - x86_64 - Updates 163 kB/s | 3.6 MB 00:22
Dependencies resolved.
=====
Package Arch Version
Repository Size
=====
Installing:
gitflow x86_64 1.12.3-1.fc34
copr:copr.fedorainfracloud.org:elegos:gitflow 57 k
Transaction Summary
=====
Install 1 Package

Total download size: 57 k
Installed size: 262 k
Is this ok [y/N]: y
Downloading Packages:
gitflow-1.12.3-1.fc34.x86_64.rpm 49 kB/s | 57 kB 00:01
```

Рис. 3.2: Install gitflow

## 2. Установка Node.js

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов.

```
[lchileshe@lchileshe ~]$ sudo dnf install nodejs
[sudo] password for lchileshe:
Last metadata expiration check: 0:05:31 ago on Fri 08 Mar 2024 12:50:38 PM MSK.
Dependencies resolved.
=====
Package                Arch      Version                               Repo      Size
=====
Installing:
nodejs                  x86_64    1:20.10.0-3.fc39                     updates   48 k
Installing dependencies:
nodejs-libs             x86_64    1:20.10.0-3.fc39                     updates   15 M
Installing weak dependencies:
nodejs-docs             noarch    1:20.10.0-3.fc39                     updates   8.1 M
nodejs-full-i18n        x86_64    1:20.10.0-3.fc39                     updates   8.5 M
nodejs-npm              x86_64    1:10.2.3-1.20.10.0.3.fc39           updates   2.2 M

Transaction Summary
=====
Install 5 Packages

Total download size: 34 M
Installed size: 183 M
Is this ok [y/N]: Y
Downloading Packages:
(1/5): nodejs-20.10.0-3.fc39.x86_64.rpm 25 kB/s | 48 kB      00:01
(2/5): nodejs-docs-20.10.0-3.fc39.noarc 73 kB/s | 8.1 MB     01:53
(3/5): nodejs-full-i18n-20.10.0-3.fc39. 71 kB/s | 8.5 MB     02:02
(4/5): nodejs-npm-10.2.3-1.20.10.0.3.fc 100 kB/s | 2.2 MB     00:22
(5/5): nodejs-libs-20.10.0-3.fc39.x86_6 88 kB/s | 15 MB      03:00 =
-----
Total                                191 kB/s | 34 MB     03:03
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Running scriptlet: nodejs-1:20.10.0-3.fc39.x86_64 1/1
  Preparing :                                           1/1
```

Рис. 3.3: Node.js

**Комментарий** :Я установил Node.js с помощью команды `dnf install nodejs`

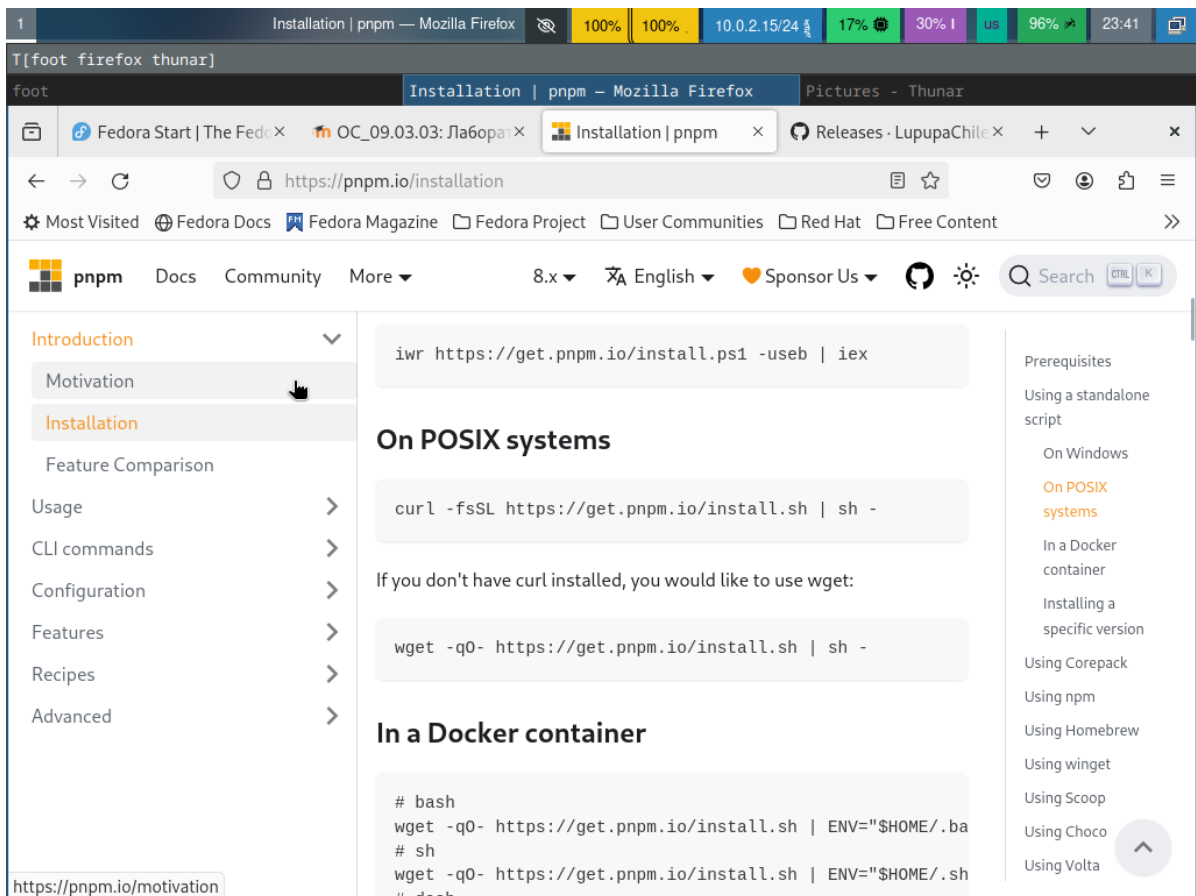


Рис. 3.4: pnpm

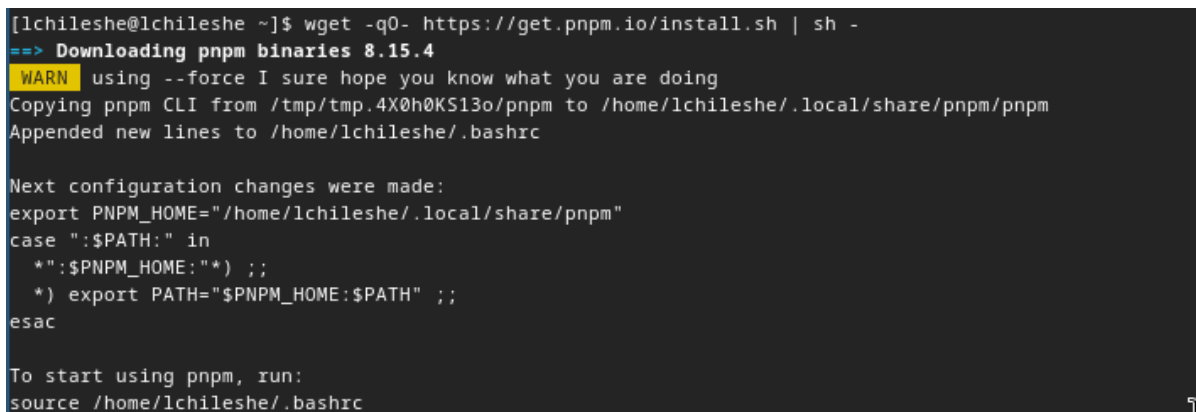


Рис. 3.5: pnpm

**Комментарий :**Я скачал pnpm из интернета

### 3. Настройка Node.js

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH.

```
[lchileshe@lchileshe ~]$  
pnpm setup  
bash: pnpm: command not found  
[lchileshe@lchileshe ~]$ source ~/.bashrc  
[lchileshe@lchileshe ~]$ pnpm setup  
No changes to the environment were made. Everything is already up to date.  
[lchileshe@lchileshe ~]$  
  
source ~/.bashrc
```

Рис. 3.6: pnpm setup

#### 4. Общепринятые коммиты

- commitizen: Данная программа используется для помощи в форматировании КОММИТОВ.

```
[lchileshe@lchileshe ~]$ pnpm add -g commitizen  
Packages: +152  
+++++  
Downloading registry.npmjs.org/typescript/5.4.2: 5.82 MB/5.82 MB, done  
Progress: resolved 152, reused 0, downloaded 152, added 152, done  
  
/home/lchileshe/.local/share/pnpm/global/5:  
+ commitizen 4.3.0  
  
Done in 58.5s
```

Рис. 3.7: pnpm add -g commitizen

- standard-changelog Данная программа используется для помощи в создании ЛОГОВ.

```
[lchileshe@lchileshe ~]$ pnpm add -g standard-changelog  
Packages: +56  
+++++  
Progress: resolved 208, reused 152, downloaded 56, added 56, done  
  
/home/lchileshe/.local/share/pnpm/global/5:  
+ standard-changelog 5.0.0  
  
Done in 14.8s
```

Рис. 3.8: pnpm add -g standard-changelog

- Практический сценарий использования git

## Создание репозитория git

### Подключение репозитория к github

я создал репозиторий на GitHub. Для примера назовём его git-extended.

```
[lchileshe@lchileshe ~]$ git clone --recursive git@github.com:LupupaChileshe/git-extended.git
Cloning into 'git-extended'...
warning: You appear to have cloned an empty repository.
```

Рис. 3.9: git-extended

Делаем первый коммит и выкладываем на github:

```
[lchileshe@lchileshe git-extended]$ touch README.md
[lchileshe@lchileshe git-extended]$ ls
README.md
[lchileshe@lchileshe git-extended]$ git add .
[lchileshe@lchileshe git-extended]$ git commit -m "first commit"
[main (root-commit) 176fa73] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
[lchileshe@lchileshe git-extended]$ git remote add origin git@github.com:LupupaChileshe/git-extended.git
error: remote origin already exists.
[lchileshe@lchileshe git-extended]$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 869 bytes | 869.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:LupupaChileshe/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Рис. 3.10: git commit

- Конфигурация общепринятых коммитов

### Конфигурация для пакетов Node.js

```
[lchileshe@lchileshe git-extended]$ pnpm init
Wrote to /home/lchileshe/git-extended/package.json

{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Рис. 3.11: Конфигурация для пакетов

Необходимо заполнить несколько параметров пакета.

Название пакета.

Лицензия пакета. Список лицензий для npm: <https://spdx.org/licenses/>. Предлагается лицензия CC-BY-4.0.

Сконфигурируем формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов:

```
"config": { "commitizen": { "path": "cz-conventional-changelog" } }
```

Таким образом, файл package.json приобретает вид:

```
{ "name": "git-extended", "version": "1.0.0", "description": "Git repo for
educational purposes", "main": "index.js", "repository": "git@github.com:username/git-
extended.git", "author": "Name Surname username@gmail.com", "license": "CC-
BY-4.0", "config": { "commitizen": { "path": "cz-conventional-changelog" } }
}
```

- Добавим новые файлы:

```
git add .
```

- Выполним коммит:

```
git cz
```

```

[lchileshe@lchileshe git-extended]$ nano package.json
[lchileshe@lchileshe git-extended]$ git add .
[lchileshe@lchileshe git-extended]$ git cz
cz-cli@4.3.0, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: feat:      A new
feature
? What is the scope of this change (e.g. component or file name): (press
  enter to skip)
? Write a short, imperative tense description of the change (max 94
  chars):
  (10) added file
? Provide a longer description of the change: (press enter to skip)

? Are there any breaking changes? No
? Does this change affect any open issues? No
[main 1581d65] feat: added file
 1 file changed, 15 insertions(+)
  create mode 100644 package.json
[lchileshe@lchileshe git-extended]$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.15 KiB | 1.15 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:LupupaChileshe/git-extended.git
 176fa73..1581d65  main -> main

```

Рис. 3.12: заполнение параметров пакета.

- Конфигурация git-flow

Инициализируем git-flow

```

[lchileshe@lchileshe git-extended]$ git flow init

Which branch should be used for bringing forth production releases?
  - main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
Hooks and filters directory? [/home/lchileshe/git-extended/.git/hooks]

```

Рис. 3.13: git flow init

- Проверьте, что Вы на ветке develop:

```
[lchileshe@lchileshe git-extended]$ git branch
* develop
main
```

Рис. 3.14: git branch

- Загрузите весь репозиторий в хранилище

```
[lchileshe@lchileshe git-extended]$ git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/LupupaChileshe/git-extended/pull/new/develop
remote:
To github.com:LupupaChileshe/git-extended.git
 * [new branch]      develop -> develop
```

Рис. 3.15: git push --all

- Установите внешнюю ветку как вышестоящую для этой ветки:

```
[lchileshe@lchileshe git-extended]$
git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 3.16: git branch --set-upstream-to=origin/develop develop

- Создадим релиз с версией 1.0.0

```
[lchileshe@lchileshe git-extended]$ git flow release start 1.0.0
Switched to a new branch 'release/1.0.0'

Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.0.0'
```

Рис. 3.17: git flow release start 1.0.0



- Создадим журнал изменений

```
[lchileshe@lchileshe git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
```

Рис. 3.18: standard-changelog --first-release

- Добавим журнал изменений в индекс

```
[lchileshe@lchileshe git-extended]$ git add CHANGELOG.md
[lchileshe@lchileshe git-extended]$ git commit -am 'chore(site): add changelog'
[release/1.0.0 6e7680a] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
```

Рис. 3.19: git add

- Зальём релизную ветку в основную ветку

```
[lchileshe@lchileshe git-extended]$ git flow release finish 1.0.0
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Merge made by the 'ort' strategy.
CHANGELOG.md | 9 ++++++++
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
Already on 'main'
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Merge made by the 'ort' strategy.
CHANGELOG.md | 9 ++++++++
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
Deleted branch release/1.0.0 (was 6e7680a).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'main'
- The release was tagged 'v1.0.0'
- Release tag 'v1.0.0' has been back-merged into 'develop'
- Release branch 'release/1.0.0' has been locally deleted
- You are now on branch 'develop'
```

Рис. 3.20: git flow release finish 1.0.0

- Отправим данные на github

`git push --all`

```
[lchileshe@lchileshe git-extended]$ git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.80 KiB | 1.40 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:LupupaChileshe/git-extended.git
   1581d65..479ab4f  develop -> develop
   1581d65..65b48aa  main -> main
```

Рис. 3.21: `git push --all`

`git push --tags`

```
[lchileshe@lchileshe git-extended]$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 161 bytes | 161.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:LupupaChileshe/git-extended.git
 * [new tag]         v1.0.0 -> v1.0.0
```

Рис. 3.22: `git push --tags`

- Создадим релиз на github. Для этого будем использовать утилиты работы с github

```
[lchileshe@lchileshe git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/LupupaChileshe/git-extended/releases/tag/v1.0.0
```

Рис. 3.23: `gh release create v1.0.0 -F CHANGELOG.md`

### 3.0.2.1 Работа с репозиторием git

Разработка новой функциональности

Создадим ветку для новой функциональности

```
[lchileshe@lchileshe git-extended]$ git flow feature start feature_branch
Switched to a new branch 'feature/feature_branch'

Summary of actions:
- A new branch 'feature/feature_branch' was created, based on 'develop'
- You are now on branch 'feature/feature_branch'

Now, start committing on your feature. When done, use:

    git flow feature finish feature_branch
```

Рис. 3.24: git flow feature start feature\_branch

- Далее, продолжаем работу с git как обычно.
- По окончании разработки новой функциональности следующим шагом следует объединить ветку feature\_branch с develop

```
[lchileshe@lchileshe git-extended]$ git flow feature finish feature_branch
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Already up to date.
Deleted branch feature/feature_branch (was 479ab4f).

Summary of actions:
- The feature branch 'feature/feature_branch' was merged into 'develop'
- Feature branch 'feature/feature_branch' has been locally deleted
- You are now on branch 'develop'
```

Рис. 3.25: git flow feature finish feature\_branch

### 3.0.2.2 Создание релиза git-flow

Создадим релиз с версией 1.2.3:

```
[lchileshe@lchileshe git-extended]$ git flow release start 1.2.3
Switched to a new branch 'release/1.2.3'

Summary of actions:
- A new branch 'release/1.2.3' was created, based on 'develop'
- You are now on branch 'release/1.2.3'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.2.3'
```

Рис. 3.26: git flow release start 1.2.3

- Создадим журнал изменений

```
[lchileshe@lchileshe git-extended]$ nano package.json
[lchileshe@lchileshe git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md
```

Рис. 3.27: standard-changelog

- Добавим журнал изменений в индекс

```
[lchileshe@lchileshe git-extended]$ git add CHANGELOG.md
[lchileshe@lchileshe git-extended]$ git commit -am 'chore(site): update changelog'
[release/1.2.3 68be02e] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
```

Рис. 3.28: git add CHANGELOG.md

- Зальём релизную ветку в основную ветку

```
[lchileshe@lchileshe git-extended]$ git flow release finish 1.2.3
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Merge made by the 'ort' strategy.
  CHANGELOG.md | 4 ++++
  package.json | 2 +-
  2 files changed, 5 insertions(+), 1 deletion(-)
Already on 'main'
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Merge made by the 'ort' strategy.
  CHANGELOG.md | 4 ++++
  package.json | 2 +-
  2 files changed, 5 insertions(+), 1 deletion(-)
Deleted branch release/1.2.3 (was 68be02e).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'main'
- The release was tagged 'v1.2.3'
- Release tag 'v1.2.3' has been back-merged into 'develop'
- Release branch 'release/1.2.3' has been locally deleted
- You are now on branch 'develop'
```

Рис. 3.29: git flow release finish 1.2.3

- Отправим данные на github

git push --all

```
[lchileshe@lchileshe git-extended]$ git push --all
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.78 KiB | 1.39 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:LupupaChileshe/git-extended.git
  479ab4f..a1ca712 develop -> develop
  65b48aa..a9d66d7 main -> main
```

Рис. 3.30: git push --all

git push --tags

```
[lchileshe@lchileshe git-extended]$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 160 bytes | 160.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:LupupaChileshe/git-extended.git
 * [new tag]          v1.2.3 -> v1.2.3
[lchileshe@lchileshe git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/LupupaChileshe/git-extended/releases/tag/v1.2.3
[lchileshe@lchileshe git-extended]$
```

Рис. 3.31: git push –tags

- Создадим релиз на github с комментарием из журнала изменений

```
[lchileshe@lchileshe git-extended]$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/LupupaChileshe/git-extended/releases/tag/v1.2.3
```

Рис. 3.32: gh release create v1.2.3 -F CHANGELOG.md

## 4 Выводы

Я получил навыки корректной работы с git-репозиториями.

## **Список литературы**