

# **Отчёт по лабораторной работе**

Исаев Рамазан Курбанович

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Задание</b>	<b>6</b>
<b>3 Теоретическое введение</b>	<b>7</b>
<b>4 Выполнение лабораторной работы</b>	<b>8</b>
4.1 Установка программного обеспечения . . . . .	8
4.2 Практический сценарий использования git . . . . .	10
<b>5 Выводы</b>	<b>17</b>
<b>Список литературы</b>	<b>18</b>

# Список иллюстраций

4.1 Установка git-flow . . . . .	8
4.2 Установка Node.js . . . . .	8
4.3 добавление каталога . . . . .	9
4.4 настройка . . . . .	9
4.5 Делаем первый коммит и выкладываем на github . . . . .	10
4.6 Конфигурация общепринятых коммитов . . . . .	10
4.7 отправка файлов . . . . .	11
4.8 . . . . .	11
4.9 . . . . .	12
4.10 . . . . .	12
4.11 . . . . .	12
4.12 . . . . .	13
4.13 . . . . .	13
4.14 . . . . .	13
4.15 . . . . .	13
4.16 объединение ветки . . . . .	14
4.17 . . . . .	14
4.18 . . . . .	15
4.19 . . . . .	15
4.20 . . . . .	16
4.21 . . . . .	16

# **Список таблиц**

# **1 Цель работы**

Получение навыков правильной работы с репозиториями git.

## **2 Задание**

Выполнить работу для тестового репозитория. Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

## 3 Теоретическое введение

Gitflow Workflow опубликована и популяризована Винсентом Дриссеном.

Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта.

Данная модель отлично подходит для организации рабочего процесса на основе релизов.

Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рабочей среде.

Последовательность действий при работе по модели Gitflow:

Из ветки master создаётся ветка develop.

Из ветки develop создаётся ветка release.

Из ветки develop создаются ветки feature.

Когда работа над веткой feature завершена, она сливаются с веткой develop.

Когда работа над веткой релиза release завершена, она сливаются в ветки develop и master.

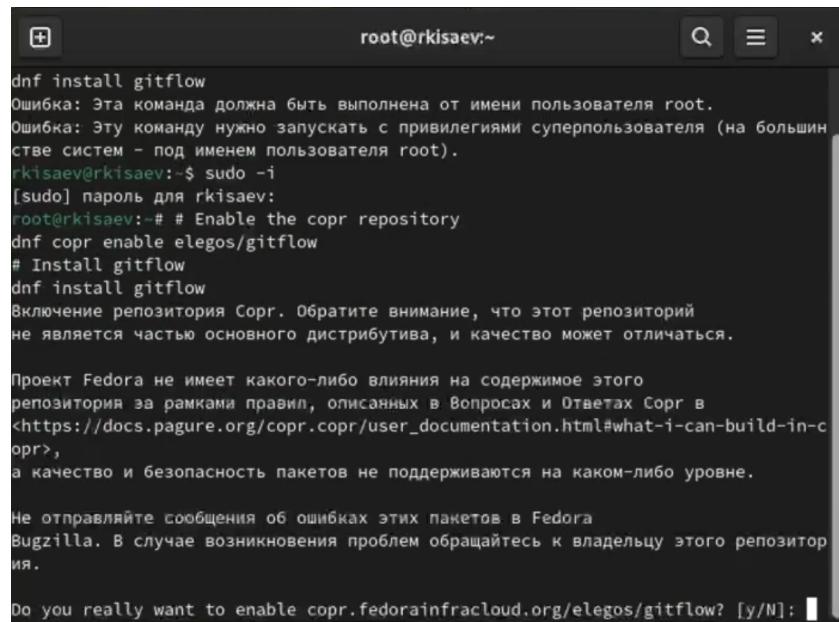
Если в master обнаружена проблема, из master создаётся ветка hotfix.

Когда работа над веткой исправления hotfix завершена, она сливается в ветки develop и master.

# 4 Выполнение лабораторной работы

## 4.1 Установка программного обеспечения

### 1. Установка git-flow



```
root@rkisaev:~ dnf install gitflow
Ошибка: Эта команда должна быть выполнена от имени пользователя root.
Ошибка: Эту команду нужно запускать с привилегиями суперпользователя (на большинстве систем - под именем пользователя root).
rkisaev@rkisaev:~ $ sudo -i
[sudo] пароль для rkisaev:
root@rkisaev:~ # # Enable the copr repository
dnf copr enable elegos/gitflow
# Install gitflow
dnf install gitflow
Включение репозитория Copr. Обратите внимание, что этот репозиторий не является частью основного дистрибутива, и качество может отличаться.

Проект Fedora не имеет какого-либо влияния на содержимое этого репозитория за рамками правил, описанных в Вопросах и Ответах Copr в <https://docs.pagure.org/copr.copr/user_documentation.html#what-i-can-build-in-copr>, а качество и безопасность пакетов не поддерживаются на каком-либо уровне.

Не отправляйте сообщения об ошибках этих пакетов в Fedora Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозитория.

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

Рис. 4.1: Установка git-flow

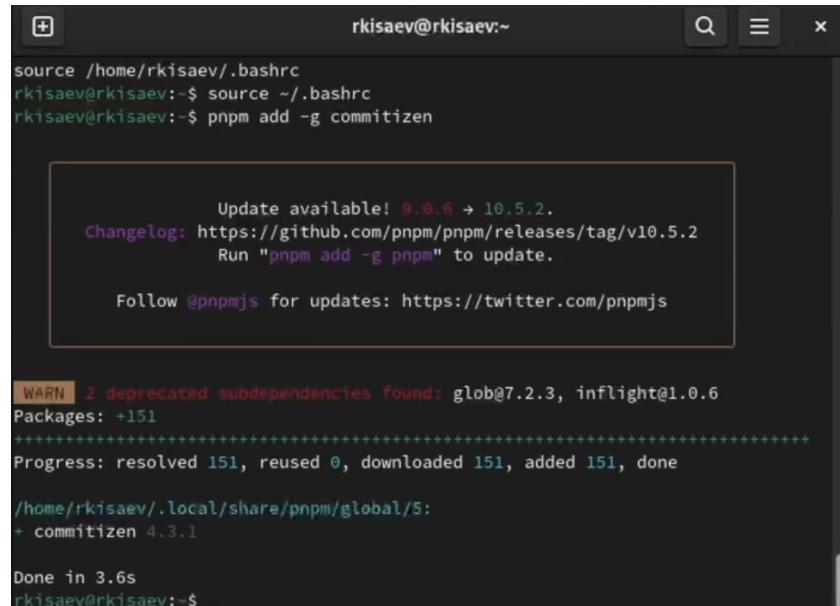
### 2. Установка Node.js

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

#### Установка Node.js

Рис. 4.2: Установка Node.js

### 3. Для работы с Node.js добавим каталог с исполняемыми файлами



```
source /home/rkisaev/.bashrc
rkisaev@rkisaev:~$ source ~/.bashrc
rkisaev@rkisaev:~$ pnpm add -g commitizen

Update available! 9.8.6 → 10.5.2.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.5.2
Run "pnpm add -g pnpm" to update.

Follow @pnpmjs for updates: https://twitter.com/pnpmjs

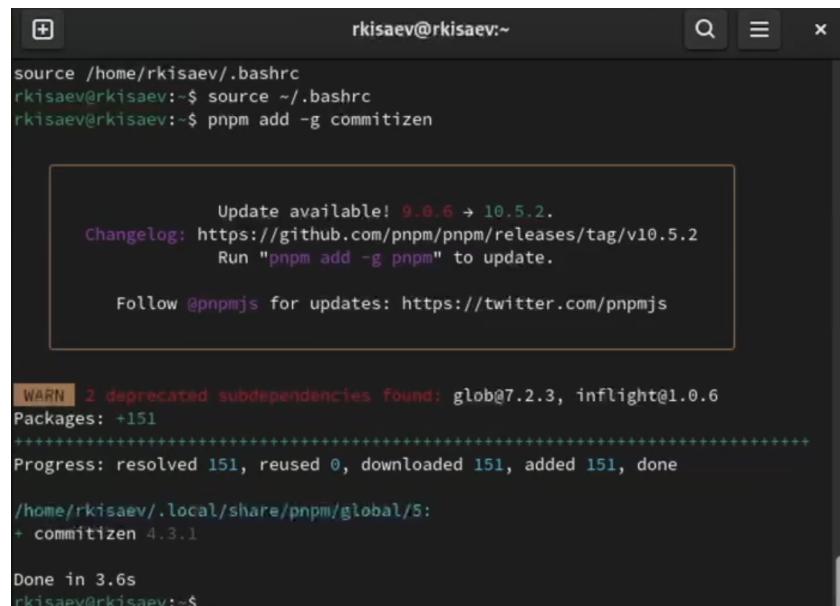
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Packages: +151
Progress: resolved 151, reused 0, downloaded 151, added 151, done

/home/rkisaev/.local/share/pnpm/global/5:
+ commitizen 4.3.1

Done in 3.6s
rkisaev@rkisaev:~$
```

Рис. 4.3: добавление каталога

### 4. Настраиваем commitizen, standard-changelog



```
source /home/rkisaev/.bashrc
rkisaev@rkisaev:~$ source ~/.bashrc
rkisaev@rkisaev:~$ pnpm add -g commitizen

Update available! 9.8.6 → 10.5.2.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.5.2
Run "pnpm add -g pnpm" to update.

Follow @pnpmjs for updates: https://twitter.com/pnpmjs

WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Packages: +151
Progress: resolved 151, reused 0, downloaded 151, added 151, done

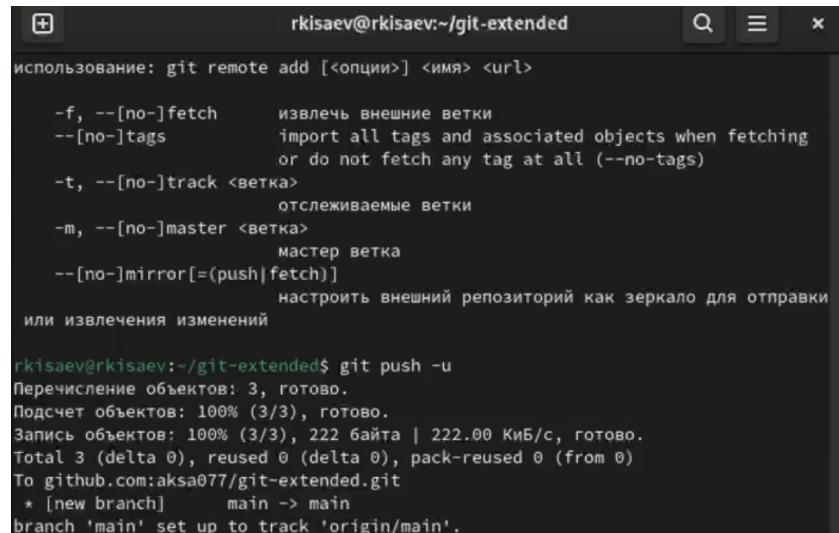
/home/rkisaev/.local/share/pnpm/global/5:
+ commitizen 4.3.1

Done in 3.6s
rkisaev@rkisaev:~$
```

Рис. 4.4: настройка

## 4.2 Практический сценарий использования git

- Подключение репозитория к github. Создаем репозиторий Git, для примера назовём его git-extended.



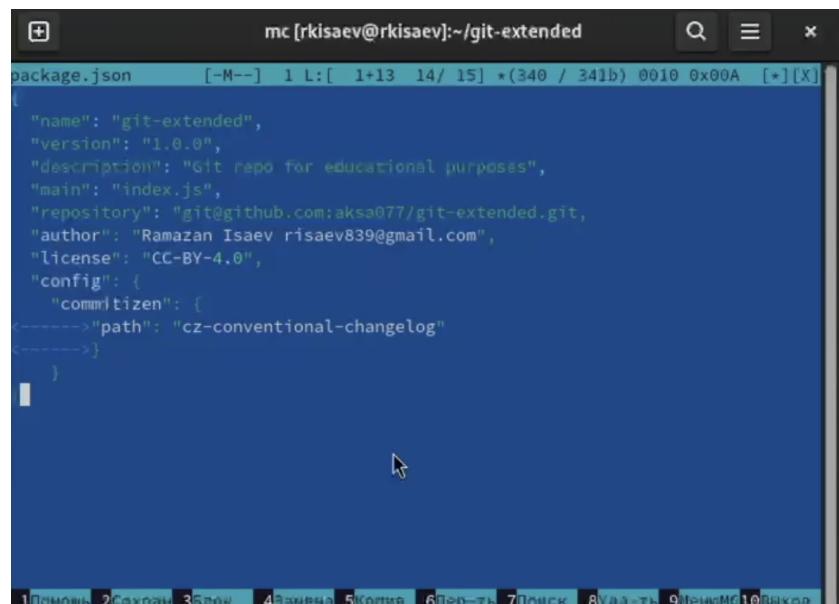
```
rkisaev@rkisaev:~/git-extended
использование: git remote add [<опции>] <имя> <url>

-f, --[no-]fetch      извлечь внешние ветки
--[no-]tags          import all tags and associated objects when fetching
                     or do not fetch any tag at all (--no-tags)
-t, --[no-]track <ветка>
                     отслеживаемые ветки
-m, --[no-]master <ветка>
                     мастер ветка
--[no-]mirror[=(push|fetch)]
                     настроить внешний репозиторий как зеркало для отправки
или извлечения изменений

rkisaev@rkisaev:~/git-extended$ git push -u
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 222 байта | 222.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:aksa077/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Рис. 4.5: Делаем первый коммит и выкладываем на github

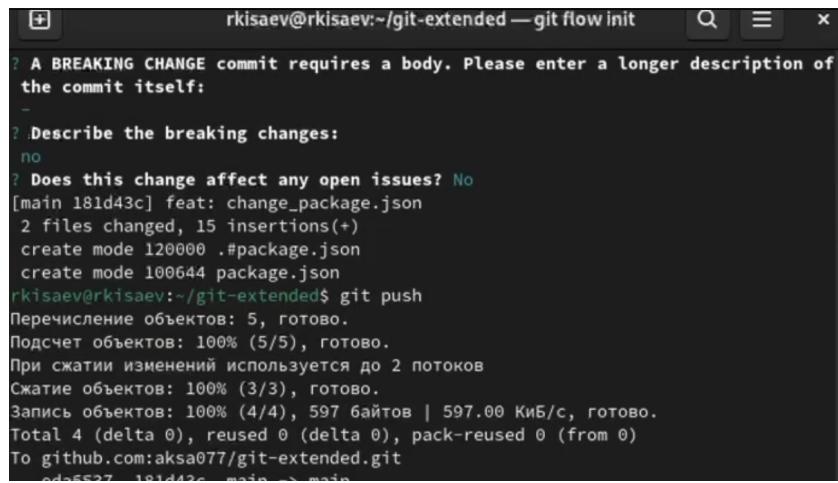
- Конфигурация общепринятых коммитов. Для этого добавим в файл package.json команду для форматирования коммитов.



```
mc [rkisaev@rkisaev]:~/git-extended
package.json [-M--] 1 L:[ 1+13 14/ 15] *(340 / 341b) 0010 0x00A [*][X]
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:aksa077/git-extended.git",
  "author": "Ramazan Isaev risaev839@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.6: Конфигурация общепринятых коммитов

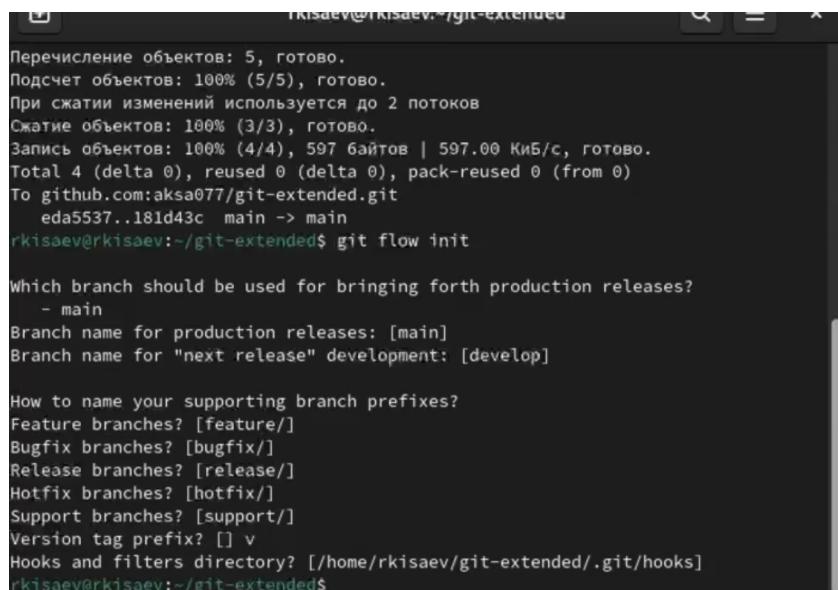
3. Добавляем файлы, выполняем коммит, отправляем на Github.



```
? A BREAKING CHANGE commit requires a body. Please enter a longer description of the commit itself:  
-  
? Describe the breaking changes:  
no  
? Does this change affect any open issues? No  
[main 181d43c] feat: change_package.json  
2 files changed, 15 insertions(+)  
create mode 120000 .#package.json  
create mode 100644 package.json  
rkisaev@rkisaev:~/git-extended$ git push  
Перечисление объектов: 5, готово.  
Подсчет объектов: 100% (5/5), готово.  
При сжатии изменений используется до 2 потоков  
Сжатие объектов: 100% (3/3), готово.  
Запись объектов: 100% (4/4), 597 байтов | 597.00 КИБ/с, готово.  
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To github.com:aksa077/git-extended.git  
    eda5537..181d43c main -> main
```

Рис. 4.7: отправка файлов

4. Инициализируем git-flow. Префикс для ярлыков установим в v. Загружаем весь репозиторий в хранилище. Установим внешнюю ветку как вышестоящую для этой ветки, создадим релиз с версией 1.0.0



```
rkisaev@rkisaev:~/git-extended$  
Перечисление объектов: 5, готово.  
Подсчет объектов: 100% (5/5), готово.  
При сжатии изменений используется до 2 потоков  
Сжатие объектов: 100% (3/3), готово.  
Запись объектов: 100% (4/4), 597 байтов | 597.00 КИБ/с, готово.  
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To github.com:aksa077/git-extended.git  
    eda5537..181d43c main -> main  
rkisaev@rkisaev:~/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/] [feature/]  
Bugfix branches? [bugfix/] [bugfix/]  
Release branches? [release/] [release/]  
Hotfix branches? [hotfix/] [hotfix/]  
Support branches? [support/] [support/]  
Version tag prefix? [] v  
Hooks and filters directory? [/home/rkisaev/git-extended/.git/hooks] [/home/rkisaev/git-extended/.git/hooks]  
rkisaev@rkisaev:~/git-extended$
```

Рис. 4.8: .

```
rkisaev@rkisaev:~/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/rkisaev/git-extended/.git/hooks]
rkisaev@rkisaev:~/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/aksa077/git-extended/pull/new/develop
remote:
To github.com:aksa077/git-extended.git
 * [new branch]      develop -> develop
rkisaev@rkisaev:~/git-extended$
```

Рис. 4.9: .

```
Everything up-to-date
rkisaev@rkisaev:~/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
rkisaev@rkisaev:~/git-extended$ git flow release start 1.0.0
Переключились на новую ветку «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. 4.10: .

5. Создадим журнал изменений Добавим журнал изменений в индекс Зальем релизную ветку в основную Отправим данные на Github. Для этого будем использовать утилиты работы с github

```
rkisaev@rkisaev:~/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
```

Рис. 4.11: .

```
rkisaev@rkisaev:~/git-extended$ git add CHANGELOG.md
rkisaev@rkisaev:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 355dbe3] chore(site): add changelog
 1 file changed, 14 insertions(+)
  create mode 100644 CHANGELOG.md
rkisaev@rkisaev:~/git-extended$
```

Рис. 4.12: .

```
rkisaev@rkisaev:~/git-extended$ git flow release finish 1.0.0
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
  CHANGELOG.md | 14 ++++++
  1 file changed, 14 insertions(+)
  create mode 100644 CHANGELOG.md
Уже на «main»
Ваша ветка опережает «origin/main» на 2 коммита.
  (используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
Fatal: Tagging failed. Please run finish again to retry.
rkisaev@rkisaev:~/git-extended$
```

Рис. 4.13: .

```
rkisaev@rkisaev:~/git-extended$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 615 байтов | 615.00 КиБ/с, готово.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:aksa077/git-extended.git
  181d43c..55e5141 main -> main
 * [new branch]      release/1.0.0 -> release/1.0.0
rkisaev@rkisaev:~/git-extended$ git push --tags
Everything up-to-date
rkisaev@rkisaev:~/git-extended$
```

Рис. 4.14: .

```
rkisaev@rkisaev:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/aksa077/git-extended/releases/tag/v1.0.0
rkisaev@rkisaev:~/git-extended$
```

Рис. 4.15: .

6. Создадим ветку для новой функциональности. Следующим шагом следует объединить ветку feature\_branch с develop

```
rkisaev@rkisaev:~/git-extended$ git flow feature start feature_branch
Переключились на новую ветку «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

rkisaev@rkisaev:~/git-extended$ git flow feature finish feature_branch
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Уже актуально.
Ветка feature/feature_branch удалена (была 06dc33c).

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. 4.16: объединение ветки

7. Создадим релиз с версией 1.2.3 Обновим номер версии в файле package.json. Установим ее в 1.2.3.

```
rkisaev@rkisaev:~/git-extended$ git flow release start 1.2.3
Переключились на новую ветку «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. 4.17: .

```
rkisaev@rkisaev:~/git-extended$ cat package.json
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:aksa077/git-extended.git",
  "author": "Ramazan Isaev rishaev839@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.18: .

8. Создадим журнал изменений Добавим журнал изменений в индекс Зальем релизную ветку Отправим данные на Github Создадим релиз на Github с комментарием из журнала изменений

```
rkisaev@rkisaev:~/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
```

Рис. 4.19: .

```
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +---
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Уже на «main»
Ваша ветка опережает «origin/main» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 +---
package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Ветка release/1.2.3 удалена (была 96cdb44).

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. 4.20: .

```
rkisaev@rkisaev:~/git-extended$ git push --all
- 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'

rkisaev@rkisaev:~/git-extended$ git push --tags
rkisaev@rkisaev:~/git-extended$
```

Рис. 4.21: .

## **5 Выводы**

Мы получили навыки правильной работы с репозиториями git.

## **Список литературы**