

Основы программной инженерии (ПИ)

Системы контроля версий. Часть2

План лекции:

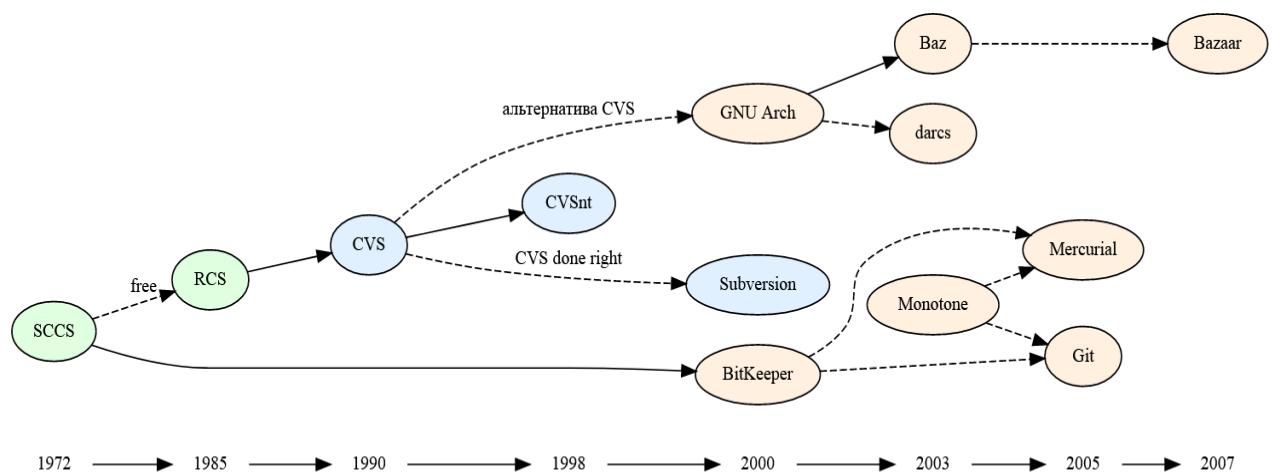
- ветвления в Git;
- создание, слияние веток в Git;
- конфликты при слиянии веток в Git;
- основы работы с удаленными репозиториями;
- совместная работа над проектом (коллаборация).

1. На прошлой лекции

Системы контроля версий

Система управления **версиями** (от англ. *VersionControl System*, *VCS* или *Revision Control System*, *RCS*) – программное обеспечение для облегчения работы с изменяющейся информацией и разработки проекта совместно с коллегами.

История:



Ссылка:

<https://yourcmc.ru/wiki/images/generated/graph/5/54/54d518e63f6f25f3a2a3c90a1ab664d9/graph.source.svg>

Что такое Git

- Git – распределенная система контроля версий;
- поддерживается автономная работа (локальные фиксации изменений могут быть отправлены в репозиторий позже);
- каждое рабочее дерево в Git содержит хранилище с полной историей проекта;
- ни одно хранилище Git не является по своей природе более важным, чем любое другое;
- целостность Git: все объекты сохраняются в базу данных не по имени, а по хеш-сумме содержимого объекта;
- удобный и интуитивно понятный набор команд;
- гибкая система ветвления проектов и слияния веток между собой;
- почти все операции выполняются локально;
- высокая производительность: скорость работы в Git кажется мгновенной;
- универсальный сетевой доступ с использованием протоколов http, ftp, rsync, ssh и др.

Основные определения

Репозиторий Git:	Git хранит информацию в структуре данных, называемой <i>репозиторий</i> (repository). Репозиторий хранится в папке проекта – в папке <code>.git</code>
Репозиторий хранит:	<ul style="list-style-type: none">– набор коммитов (<i>commit objects</i>)– набор ссылок на коммиты (<i>heads</i>)
Commit objects содержат:	<ul style="list-style-type: none">– набор файлов, отображающий состояние проекта в текущий момент времени– ссылки на родительские <i>commit objects</i>– SHA1-имя – 40 символьная строка, которая <i>уникально идентифицирует commit object</i>

Основные команды

```
git init – создание репозитория  
git add <имена файлов> – добавляет файлы в индекс  
git commit – выполняет коммит проиндексированных файлов в репозиторий  
git status – показывает какие файлы изменились между текущей стадией и HEAD. Файлы разделяются на 3 категории: новые файлы, измененные файлы, добавленные новые файлы  
git checkout <SHA1 или метка> – получение указанной версии файла  
git push – отправка изменений в удаленный репозиторий  
git fetch – получение изменений из удаленного репозитория  
git clone <remote url> – клонирование удаленного репозитория себе
```

Что такое Github

Github – крупнейший веб-сервис онлайн-хостинга репозиториев, используется для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий **Git**.

Github

- бесплатный сервис для проектов с открытым исходным кодом;
- место сотрудничества миллионов разработчиков и проектов (кроме размещения кода участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых); GitHub – социальная сеть для разработчиков;
- предоставляет удобный интерфейс для совместной разработки проектов и может отображать вклад каждого участника в виде дерева;
- дополнительно для проектов есть личные страницы, вики-разметка и система отслеживания ошибок;
- предоставляет возможность прямо на сайте просмотреть файлы проектов с подсветкой синтаксиса для большинства языков программирования.

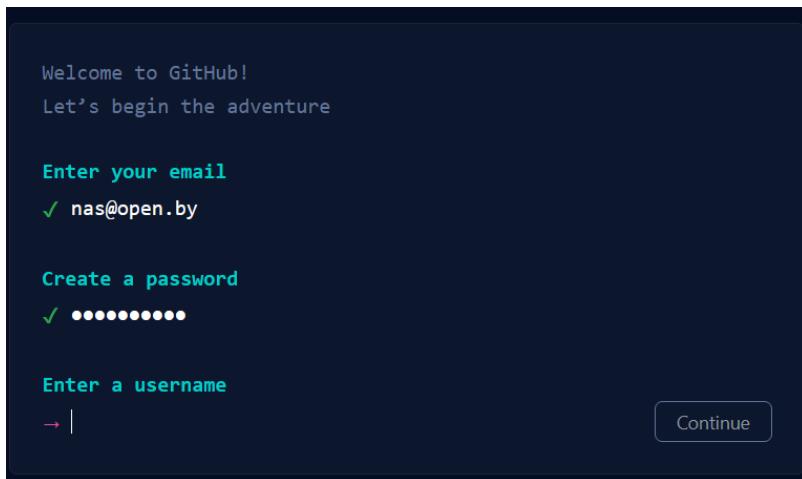
Чем отличается Git и GitHub

Git:	<ul style="list-style-type: none">– инструмент, позволяющий реализовать распределённую систему контроля версий.
GitHub:	<ul style="list-style-type: none">– сервис для проектов, использующих Git.

2. GitHub – Сопровождение проекта

Настройка и конфигурация учетной записи

- создать бесплатную учётную запись на сайте <https://github.com/>
- выбрать логин, который еще не занят, указать адрес вашей электронной почты и пароль.



- на следующем шаге (по желанию) можно настроить ваш профиль во вкладке «Settings»

A screenshot of the GitHub 'Public profile' settings page for a user named 'adeln'. The left sidebar shows 'Account settings' with options like Profile, Account, Appearance, etc. The main area shows 'Public profile' settings with fields for Name, Public email, Bio, and URL. A 'Profile picture' section shows a green pixelated image with an edit button. The top navigation bar includes 'Search or jump to...', 'Pull requests', 'Issues', 'Marketplace', 'Explore', and a user dropdown menu. The dropdown menu is open, showing 'Signed in as adeln', 'Set status', 'Your profile', 'Your repositories', etc., with 'Settings' highlighted in blue.

- Создадим репозиторий с именем mytest (режим доступа Public):

A screenshot of the GitHub repository creation menu. On the left, there is a user profile icon and a 'Repositories' section with a 'New' button. On the right, a dropdown menu is open with options: 'New repository' (which is highlighted in blue), 'Import repository', 'New gist', 'New organization', and 'New project'. Below the menu, the text 'либо из меню' (or from the menu) is visible.

е) Добавим в репозиторий README.md – readme-файл, используя web-интерфейс:

The screenshot shows the GitHub 'Edit new file' interface. At the top, there are buttons for 'Set up in Desktop' (disabled), 'HTTPS', and 'SSH', with the URL <https://github.com/adeln/trtrtr.git>. Below that, a message says 'Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#)'. The main area is titled 'Edit new file' and contains a preview section. The preview shows a list of items numbered 1 to 7:

```
1 # mytest
2 Фамилия
3 Имя
4 Отчество
5 Специальность
6 Дисциплина "Название"
7 курс К, группа Г, подгруппа 1 или 2
```

ф) Зафиксируем изменения:

The screenshot shows the GitHub 'Commit changes' dialog box. It features a green circular icon with a white cross and the text 'Commit changes'. Below it is a button labeled 'Update README.md'. There is a text input field with placeholder text 'Add an optional extended description...'. At the bottom, there are two radio button options: one selected ('Commit directly to the master branch.') and one unselected ('Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)'). At the very bottom are two buttons: 'Commit changes' (green) and 'Cancel' (gray).

g) Можно отследить изменения файла, выбрав этот файл и кликнув по ссылке:

The screenshot shows a GitHub commit history for a repository. At the top, there are two entries:

Collaboration.docx	Create issue 1	14 hours ago
README.md	Update README.md	16 hours ago

The second entry, "Update README.md", is highlighted with a red underline. Below this, a modal window titled "Update README.md" displays the details of the commit:

master

adeln committed 7 days ago Verified

Showing 1 changed file with 1 addition and 0 deletions.

diff --git a/README.md b/README.md

...	@@ -2,6 +2,7 @@
2	2 Фамилия
3	3 Имя
4	4 Отчество
5	+ Специальность
5	6 Дисциплина "Название"
6	7 курс К, группа Г, подгруппа 1 или 2
7	8

Клонирование существующего репозитория в Git

Для получения локальной копии существующего Git-репозитория нужно использовать команду `git clone`

Git получает копию практически всех данных, которые есть на сервере. При выполнении `git clone` с сервера выгружается версия каждого файла из истории проекта.

Т.о., если сервер выйдет из строя, то можно использовать любой из клонов на любом из клиентов, для того, чтобы вернуть сервер в то состояние, в котором он находился в момент клонирования – все данные, помещенные под версионный контроль, будут сохранены

a) Клонирование репозитория в Git осуществляется командой `git clone <url>`:

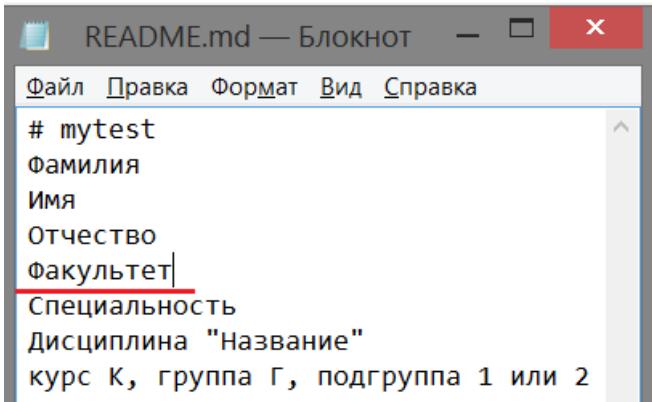
```
$ git clone https://github.com/adeln/mytest
```

```
...$ git clone https://github.com/adeln/mytest
Cloning into 'mytest'...
remote: Enumerating objects: 39, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 39 (delta 1), reused 0 (delta 0), pack-reused 33
Receiving objects: 100% (39/39), 10.79 KiB | 1.35 MiB/s, done.
Resolving deltas: 100% (6/6), done.
```

Просмотр конфигурационного файла:

```
$ git config -l
pack.packsize=2g
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files (x86)/Git/mingw32/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com/usehttppath=true
init.defaultbranch=master
user.name=adel
user.email=narkevich.adelina@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/adeln/mytest
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
```

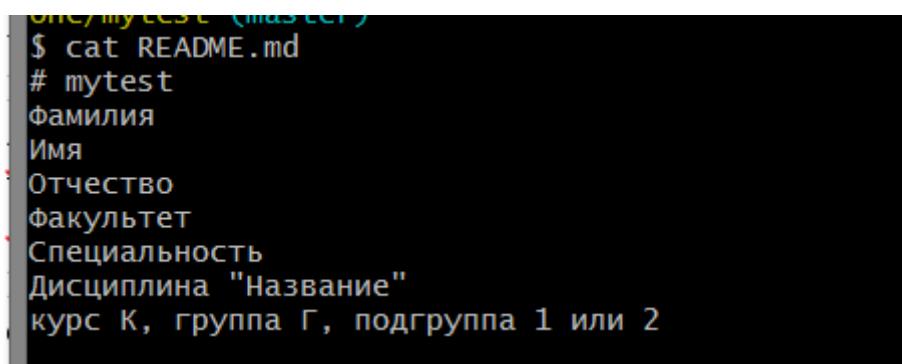
b) Внесем изменения в файл README.md



Screenshot of the Windows Notepad application showing the contents of the README.md file. The file contains the following text:

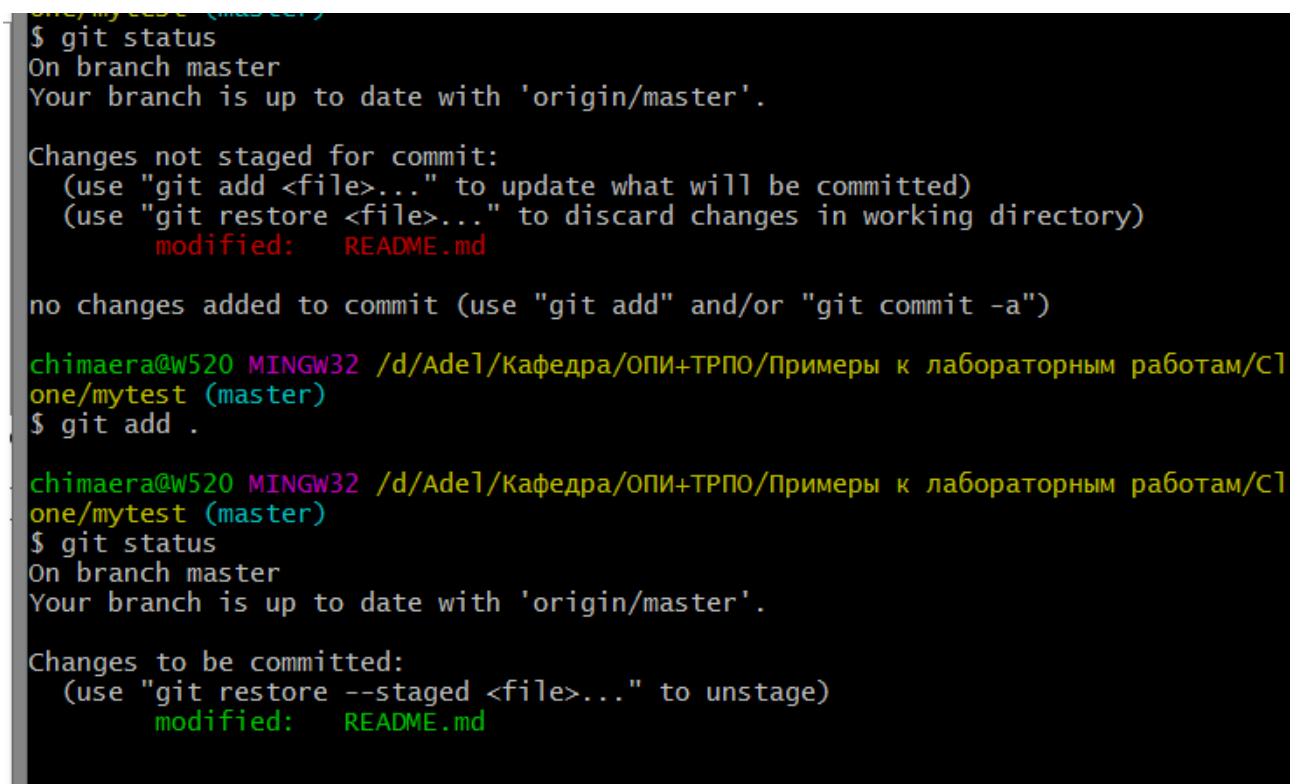
```
# mytest
Фамилия
Имя
Отчество
Факультет
Специальность
Дисциплина "Название"
курс К, группа Г, подгруппа 1 или 2
```

Просмотр содержимого файла README.md в локальном репозитории с помощью программы `cat`:



```
chimaera@W520 MINGW32 ~
$ cat README.md
# mytest
Фамилия
Имя
Отчество
факультет
Специальность
Дисциплина "Название"
курс К, группа Г, подгруппа 1 или 2
```

На скриншоте просмотр *состояния* репозитория, *индексирование* изменений файла README.md и просмотр *текущего состояния* репозитория:



```
chimaera@W520 MINGW32 ~
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

chimaera@W520 MINGW32 /d/Ade1/Кафедра/ОПИ+ТРПО/Примеры к лабораторным работам/C1
one/mytest (master)
$ git add .

chimaera@W520 MINGW32 /d/Ade1/Кафедра/ОПИ+ТРПО/Примеры к лабораторным работам/C1
one/mytest (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md
```

Фиксация изменений в репозиторий:

```
$ git commit -m "Updated from local repo"  
[master 8396e75] Updated from local repo  
 1 file changed, 2 insertions(+), 1 deletion(-)
```

Просмотр удаленных репозиториев в Git

Просмотреть в Git список *настроенных* удаленных репозиториев можно командой

```
$ git remote.
```

Выполнение: выведены названия доступных удаленных репозиториев, где origin - имя по умолчанию, которое Git дает серверу, с которого производилось клонирование:

```
$ git remote  
origin
```

Просмотр log файла*:

```
$ git log  
commit 8396e75fbfb1530f42ab5a7fa8ebd3f908a5ed7 (HEAD -> master)  
Author: adel <narkevich.adelina@gmail.com>  
Date:   Wed Nov 10 17:51:02 2021 +0300  
  
        Updated from local repo  
  
commit 722fa55b187e36d72eb4dc9fd87b873d153524c5 (origin/master, origin/HEAD)  
Author: adeln <narkevich.adelina@gmail.com>  
Date:   Wed Nov 3 22:31:57 2021 +0300  
  
        Update README.md  
  
commit 61abfc0b0826daf0b6915a89a6ed218a96c8f1dd  
Author: adeln <narkevich.adelina@gmail.com>  
Date:   Wed Nov 3 14:44:51 2021 +0300  
  
        Update README.md  
  
commit 549b29e35675e42c9cddc13b8860d0e653180260  
Author: adeln <narkevich.adelina@gmail.com>  
Date:   Fri Sep 9 18:56:00 2016 +0300
```

Отправка изменений в удаленный репозиторий (Push)

Поделиться своими наработками и отправить их в удаленный репозиторий можно с помощью команды `git push`. Формат команды:

```
git push <имя_удаленного_репозитория> <имя_локальной_ветки>
```

Выполним отправку изменений в удаленный репозиторий:

```
$ git push origin master
```

Эта команда выполнится только в случае, если с сервера клонирован репозиторий, к которому у вас есть права на запись (в нашем случае удаленный репозиторий с правами доступа **Public**)

Результат успешного выполнения команды:

```
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 338 bytes | 338.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/adeln/mytest
    722fa55..8396e75  master -> master
```

Просмотр удаленного репозитория:

The screenshot shows a GitHub repository page for 'adeln / mytest'. The repository is marked as 'Public'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation, there are buttons for master, 3 branches, 0 tags, Go to file, Add file, and Code. The main area displays a commit history:

Author	Message	Date	Commits
adeln	Updated from local repo	8396e75 1 hour ago	5 commits
	README.md	Updated from local repo	1 hour ago
	oop.cpp	Create oop.cpp	5 years ago

Below the commit history, there is a preview of the README.md file content:

mytest

Фамилия Имя Отчество Факультет Специальность Дисциплина "Название" курс К, группа Г, подгруппа 1 или 2

Указано, что файл **README.md** изменен в локальном репозитории и эти изменения зафиксированы (коммит **8396e75fbbfb1530f42ab5a7fa8ebd3f908a5ed7**), см. лог-файл выше, помеченный *)

Получение изменений из удаленного репозитория — Fetch и Pull

Изменим файл `Readme.md` в удаленном репозитории через web-интерфейс, добавив текущую дату и время.

Зафиксируем изменения и «заберем» изменения из удаленного репозитория в локальный.

формат команды `fetch`: `git fetch <URL_удаленного_репозитория>`

Выполним команду:

```
$ git fetch https://github.com/adeln/mytest
```

```
$ git fetch https://github.com/adeln/mytest
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 811 bytes | 202.00 KiB/s, done.
From https://github.com/adeln/mytest
 * branch HEAD      -> FETCH_HEAD
```

По команде выполняется обращение к указанному удаленному репозиторию и забираются все те данные проекта, которых у вас ещё нет. При этом их слияния с вашими наработками не происходит и то, над чем вы работаете в данный момент, не модифицируется.

```
$ git show
commit 8396e75fbbfb1530f42ab5a7fa8ebd3f908a5ed7 (HEAD -> master, origin/master,
origin/HEAD)
Author: adel <narkevich.adelina@gmail.com>
Date:   Wed Nov 10 17:51:02 2021 +0300

    Updated from local repo

diff --git a/README.md b/README.md
index 2092e47..b61892c 100644
--- a/README.md
+++ b/README.md
@@ -1,7 +1,8 @@
-# mytest
+# mytest
    Фамилия
    Имя
    Отчество
+Факультет
    Специальность
    Дисциплина "Название"
    курс К, группа Г, подгруппа 1 или 2
```

Можно использовать команду `git pull` чтобы автоматически получить изменения из удалённой ветки и слить их со своей текущей:

```
$ git pull https://github.com/adeln/mytest
From https://github.com/adeln/mytest
 * branch HEAD      -> FETCH_HEAD
Updating 8396e75..b0ec98f
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```

Изменения в папке локального репозитория:

Имя	Дата изменения	Тип	Размер
.git	10.11.2021 19:23	Папка с файлами	
oop.cpp	10.11.2021 17:30	Файл "CPP"	1 КБ
README.md	10.11.2021 19:23	Файл "MD"	1 КБ

Просмотр лог-файла:

```
$ git log
commit b0ec98f9866f630569e93558425ef0d668d1a725 (HEAD -> master)
Author: adeln <narkevich.adelina@gmail.com>
Date:   Wed Nov 10 18:55:16 2021 +0300

    Update README.md

commit 8396e75fbfb1530f42ab5a7fa8ebd3f908a5ed7 (origin/master, origin/HEAD)
Author: adeln <narkevich.adelina@gmail.com>
Date:   Wed Nov 10 17:51:02 2021 +0300

    Updated from local repo
```

3. Совместная работа над проектом

Совместная работа с репозиторием требуется, когда необходимо учитывать текущие задачи, выполнять требования к ним и исправлять баги.

a. На главной странице аккаунта отображается содержимое и все текущие активности:

The screenshot shows a GitHub profile for the user 'adeln'. On the left sidebar, there are sections for 'Repositories' (listing 'adeln / mytest', 'adeln / trtrtr', 'adeln / test', 'adeln / gitignore', 'smw60 / EGHRepo', and 'smw60 / EGH') and 'Recent activity' (listing '#103', 'Новый проект', and 'My new issue'). The main content area is titled 'All activity' and shows a recent event: 'ivanovmixa forked ivanovmixa/mytest from adeln/mytest 10 hours ago'. It also includes a 'ProTip!' message, a 'Subscribe to your news feed' link, and the GitHub footer with links to Blog, API, Terms, About, Training, Privacy, Shop, Status, Contact GitHub, Security, and Pricing.

b. Копирование репозитория в Github.

Последовательность действий:

- *создайте форк репозитория коллеги:*

Нажать кнопку в верхнем правом заголовке проекта коллеги.

Репозиторий коллеги клонирован в отдельную ветку:

The screenshot shows a GitHub repository for 'adeln / Python_homework'. The repository is public and was forked from 'IrinaSuh/Python_homework'. The 'Code' tab is selected, showing a single branch named 'main' (1 branch, 0 tags). A note says 'This branch is up to date with IrinaSuh:main.' Below the branches, a commit by 'IrinaSuh' is listed: 'Add files via upload' (d593332 on 21 Apr, 8 commits). Two files are shown: 'Class_Phone.py' and 'Lab-1-2.py', both added via upload 7 months ago.

- *внесите изменения в своей ветке форка (например, добавив какой-нибудь файл).*
- *создайте запрос (pull-реквест) в репозиторий коллеги, предложив свои изменения:*

Pull requests

- *после слияния ваших изменений в исходный репозиторий его владельцем, заберите в свой форк последние изменения.*

c. Добавление членов команды: организация и соавторы

Существует два способа настройки Github для совместной работы:

- *Организации.* Владелец организации может создавать множество команд с разными уровнями доступа для различных репозиториев.
- *Сотрудники.* Владелец репозитория может добавлять колабораторов с доступом Read + Write для одного репозитория.

The screenshot shows the GitHub repository settings page. At the top, there are navigation links: Issues (1), Pull requests, Discussions, Actions, Projects, Wiki, Security, Insights, and Settings (highlighted with a red box). The main area is titled "Who has access". It shows that the repository is a "PUBLIC REPOSITORY" and that "DIRECT ACCESS" is available to 2 people. A red box highlights the "Manage" button under "DIRECT ACCESS". Below this, a "Manage access" section lists two pending invitations: "IrinaSuh" and "ivanovmixa", both with "Pending Invite" status and a trash icon. On the left sidebar, a "Manage access" link is also highlighted with a red box. Other sidebar options include Options, Security & analysis, Branches, Webhooks, Notifications, Integrations, Deploy keys, Actions, Environments, Secrets, and Pages.

d. Создание проблемы (issue)

Для этого нужно включить вкладку issues.

Сделать это можно так: настройка проекта, отметить галочку issues. Появляется вкладка issues, с помощью которой можно ставить задачи и обсуждать их.

Перейти на страницу репозитория.

Под заголовком репозитория выбрать меню Issues:



Нажать кнопку

New issue

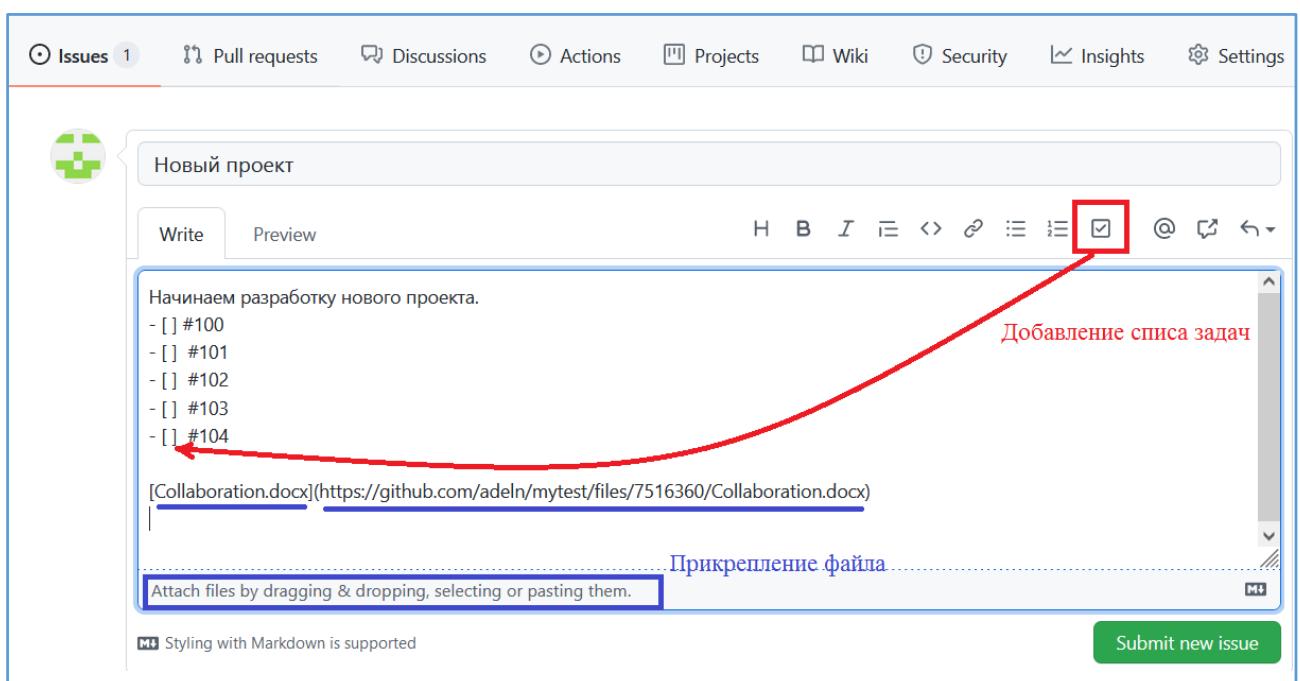
e. Заполнение информации

В поле заголовка **даем** проблеме **название** (название должно отражать суть проблемы).

В закладке Write рабочей области ввода **добавляем** текстовое **описание**, объясняющее цель проблемы, включая любые подробности, которые могут помочь решить проблему. В нашем случае инициируем начало разработки нового проекта. Определяем цель и ожидаемый результат.

Добавляем список задач – этапы разработки проекта (перед каждым элементом списка надо поставить символ []). Элементы списка могут быть обычным текстом или ссылками на существующие проблемы по их номеру либо по URL. Текст можно отформатировать.

Так же можно прикрепить файл с дополнительной информацией.



Issues 1 Pull requests Discussions Actions Projects Wiki Security Insights Settings

Новый проект

Write Preview

Начинаем разработку нового проекта.

- [] #100
- [] #101
- [] #102
- [] #103
- [] #104

[Collaboration.docx](https://github.com/adeln/mytest/files/7516360/Collaboration.docx)

Добавление списка задач

Прикрепление файла

Attach files by dragging & dropping, selecting or pasting them.

Styling with Markdown is supported

Submit new issue

Прикрепить файл можно несколькими способами:

- перетащить его из папки;
- выбрать из пункта «выбрать файл»

Имя	Дата изменения	Тип	Размер
.git	10.11.2021 20:41	Папка с файлами	
Collaboration.docx	10.11.2021 20:31	Документ Microsoft Word	13 КБ
README.md	10.11.2021 19:23	Файл "MD"	1 КБ

Issues 1

Open adeln opened this issue 2 minutes ago · 0 comments

Owner **...
adeln** commented 2 minutes ago // Заголовок
issue 1
Давайте назовем проект "Новый проект". Как вам? Есть другие предложения?
Описание в файле

Write **Preview** **H** **B** **I** **≡** **< >** **♂** **≡** **≡** **☒** **@** **✉** **↶**

Этап 1 // Прикрепленный файл
[Collaboration.docx](https://github.com/adeln/mytest/files/7515148/Collaboration.docx)

Attach files by dragging & dropping, selecting or pasting them.

Close with comment **Comment**

f. Назначение проблем и задач другим пользователям

Можно назначить до 10 человек для решения каждой проблемы, включая вас самих, и всех, у кого есть разрешения на запись в репозиторий.

Создаем новый проект, добавляем его описание и список задач, каждой из которых можно назначить исполнителя.

The screenshot shows a GitHub repository named 'adeln / mytest' with a public status. The 'Issues' tab is selected, showing 2 issues. A comment from 'adeln' is visible, stating: 'Начинаем разработку нового проекта.' Below the comment is a list of tasks: #100, #101, #102, #103, and #104. A file named 'Collaboration.docx' is attached to the comment.

Отображение назначенной проблемы у соавтора:

The screenshot shows a GitHub issue titled 'Issue 1 #1'. A comment from 'ivanovmixa' is displayed, containing the text: 'Называем проект! Может "C Collaboration.docx овместная работа?"'. This comment has been assigned the 'question' label. On the right side of the interface, there are sections for 'Assignees' (No one—assign yourself), 'Labels' (question), 'Projects' (None yet), and 'Milestone' (No milestone).