

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
дисциплины «Основы программной инженерии»

Выполнил:
Матвеев Александр Иванович
1 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

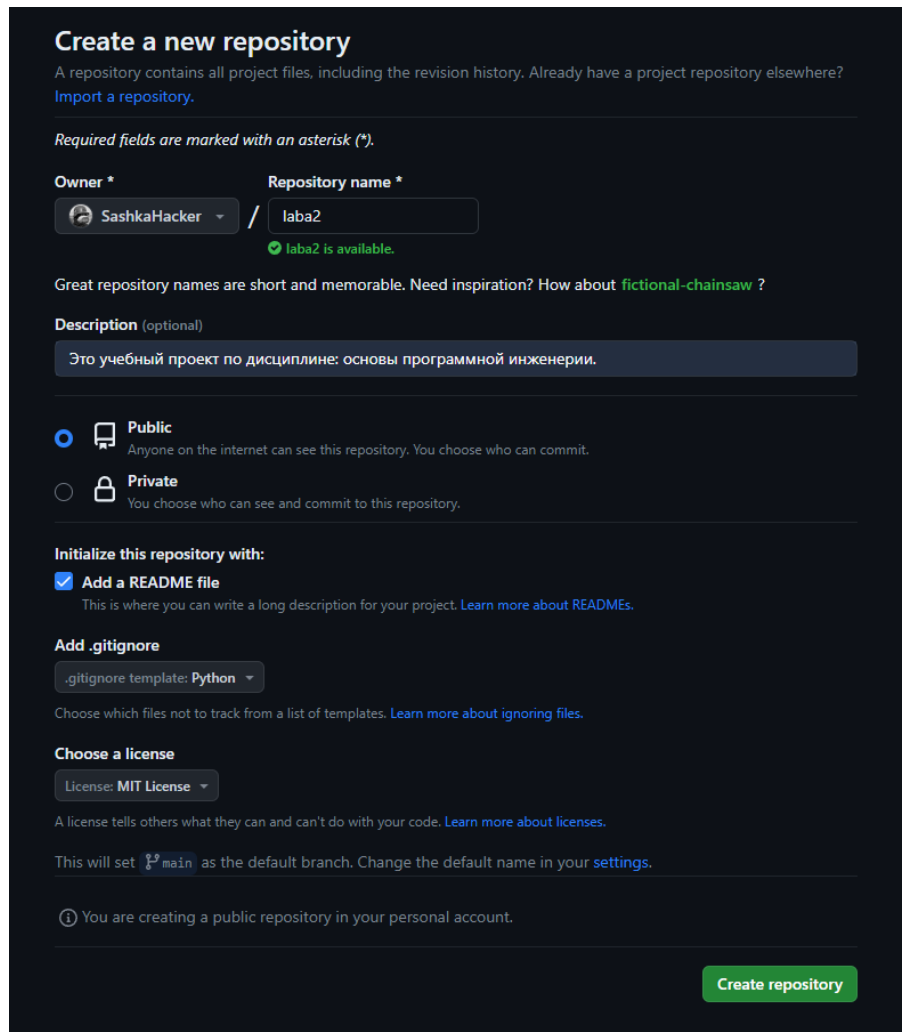
Ставрополь, 2023 г.

Тема: исследование возможностей Git для работы с локальными репозиториями.

Цель работы: исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Ход работы.

1. Создание нового репозитория с лицензией MIT.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * SashkaHacker / **Repository name *** laba2
✔ laba2 is available.

Great repository names are short and memorable. Need inspiration? How about [fictional-chainsaw](#) ?

Description (optional)
Это учебный проект по дисциплине: основы программной инженерии.

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: Python

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: MIT License

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a public repository in your personal account.

Create repository

Рисунок 1 – Создание репозитория

2. Отработка примеров из лабораторной работы.

```

Sashka@DESKTOP-U4RPSBI MINGW64 ~/simplegit-progit (master)
$ git log
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

    first commit

```

Рисунок 2 – Просмотр логов в проекте simplegit-progit

```

Sashka@DESKTOP-U4RPSBI MINGW64 ~/simplegit-progit (master)
$ git log --stat
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

Rakefile | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

lib/simplegit.rb | 5 -----
1 file changed, 5 deletions(-)

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

    first commit

README           | 6 ++++++
Rakefile         | 23 ++++++
lib/simplegit.rb | 25 ++++++
3 files changed, 54 insertions(+)

```

Рисунок 3 – Использование аргумента (--stat) для вывода логов

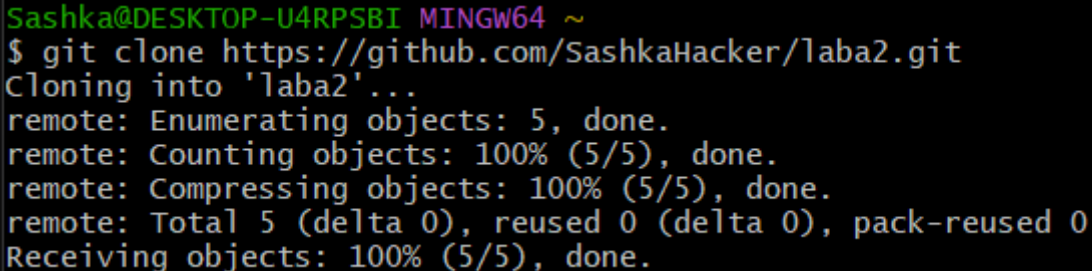
```

Sashka@DESKTOP-U4RPSBI MINGW64 ~/simplegit-progit (master)
$ git log --pretty=oneline
ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD) changed the verison number
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7 removed unnecessary test code
a11bef06a3f659402fe7563abf99ad00de2209e6 first commit

```

Рисунок 4 – Использование аргумента (--pretty) для вывода логов

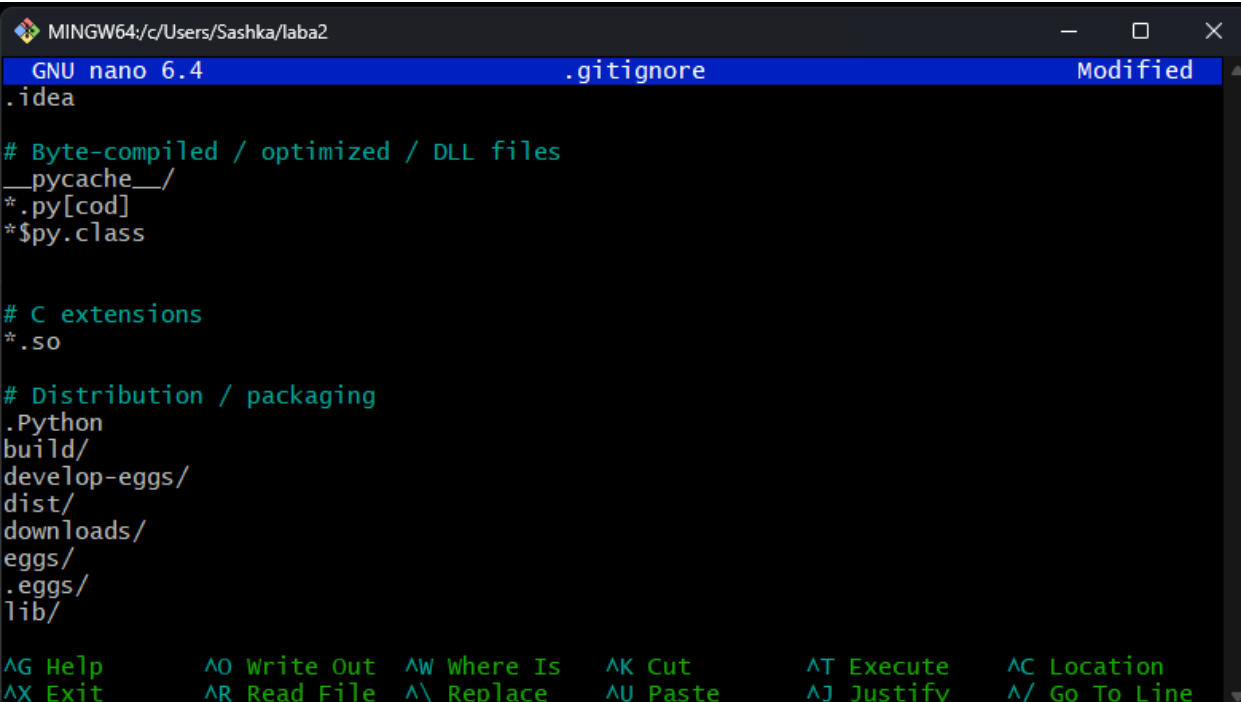
3. Клонировал репозиторий на рабочий компьютер.

A terminal window with a black background and green text. The prompt is 'Sashka@DESKTOP-U4RPSBI MINGW64 ~'. The command '\$ git clone https://github.com/SashkaHacker/laba2.git' is entered. The output shows the cloning process: 'Cloning into 'laba2'...', 'remote: Enumerating objects: 5, done.', 'remote: Counting objects: 100% (5/5), done.', 'remote: Compressing objects: 100% (5/5), done.', 'remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0', and 'Receiving objects: 100% (5/5), done.'

```
Sashka@DESKTOP-U4RPSBI MINGW64 ~  
$ git clone https://github.com/SashkaHacker/laba2.git  
Cloning into 'laba2'...  
remote: Enumerating objects: 5, done.  
remote: Counting objects: 100% (5/5), done.  
remote: Compressing objects: 100% (5/5), done.  
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (5/5), done.
```

Рисунок 5 – Клонирование репозитория

4. Дополнил файл .gitignore нужными правилами для языка Python.

A screenshot of a MINGW64 terminal window with the GNU nano 6.4 editor open to the .gitignore file. The file contains rules for ignoring Python-related files and directories. The status bar at the bottom shows various keyboard shortcuts.

```
MINGW64:/c/Users/Sashka/laba2  
GNU nano 6.4 .gitignore Modified  
.idea  
  
# Byte-compiled / optimized / DLL files  
__pycache__/  
*.py[cod]  
*$py.class  
  
# C extensions  
*.so  
  
# Distribution / packaging  
.Python  
build/  
develop-eggs/  
dist/  
downloads/  
eggs/  
.eggs/  
lib/
```

Рисунок 6 – Изменение файла .gitignore

5. Добавил в файл README.md необходимую информацию.

```
MINGW64:/c/Users/Sashka/laba2
GNU nano 6.4                                     README.md
# laba2
Это учебный проект по дисциплине: основы программной инженерии.
Выполнил студент группы ПИЖ-6-о-22-1, Матвеев Александр Иванович.
```

Рисунок 7 – Изменение README.md

6. Сделал 7 коммитов с 3-мя тегами.

```
Sashka@DESKTOP-U4RPSBI MINGW64 ~/laba2 (main)
$ git log --graph --pretty=oneline --abbrev-commit
* 5d04866 (HEAD -> main, tag: v1.2) added final changes
* 7379b19 (tag: v1.1) Add function update()
* 1c7938d Add logica of calculator
* 4040d67 Add function build()
* e250115 Add class Main
* 8502229 (tag: v1.0) add main.py
* a5bcdf9 (origin/main, origin/HEAD) Initial commit
```

Рисунок 8 – Список коммитов

7. Выполнение ряда команд: git show head/head~1/850222.

```
Sashka@DESKTOP-U4RPSBI MINGW64 ~/laba2 (main)
$ git show head
commit 5d048660918ef5fdcf18134229d26237180bc5be (HEAD -> main, tag: v1.2)
Author: Матвеев <am942081@gmail.com>
Date:   Wed Sep 20 08:17:02 2023 +0300

    added final changes

diff --git a/main.py b/main.py
index 31e9e10..becad60 100644
--- a/main.py
+++ b/main.py
@@ -51,4 +51,15 @@ class Main(Frame):
     def update(self):
         if self.formula == "":
             self.formula = "0"
-        self.lbl.configure(text=self.formula)
+        self.lbl.configure(text=self.formula)
\ No newline at end of file
+
+if __name__ == '__main__':
+    root = Tk()
+    root["bg"] = "#000"
+    root.geometry("485x550+200+200")
+    root.title("Калькулятор")
+    root.resizable(False, False)
+    app = Main(root)
+    app.pack()
+    root.mainloop()

Sashka@DESKTOP-U4RPSBI MINGW64 ~/laba2 (main)
$
```

Рисунок 9 – Вывод команды git show head

```

Sashka@DESKTOP-U4RPSBI MINGW64 ~/laba2 (main)
$ git show head~1
commit 7379b19b029d2729ea44ce7369a9a8dbac257ecf (tag: v1.1)
Author: Матвеев <am942081@gmail.com>
Date:   Wed Sep 20 08:13:44 2023 +0300

    Add function update()

diff --git a/main.py b/main.py
index afa0d7d..31e9e10 100644
--- a/main.py
+++ b/main.py
@@ -46,4 +46,9 @@ class Main(Frame):
     if self.formula == "0":
         self.formula = ""
     self.formula += operation
-    self.update()
\ No newline at end of file
+    self.update()
+
+    def update(self):
+        if self.formula == "":
+            self.formula = "0"
+        self.lbl.configure(text=self.formula)
\ No newline at end of file

Sashka@DESKTOP-U4RPSBI MINGW64 ~/laba2 (main)
$

```

Рисунок 10 – Вывод команды git show head~1

```

Sashka@DESKTOP-U4RPSBI MINGW64 ~/laba2 (main)
$ git show 85022
commit 85022295a7d340a7e415ecc712a47e44c452bb9d (tag: v1.0)
Author: Матвеев <am942081@gmail.com>
Date:   Wed Sep 20 07:42:46 2023 +0300

    add main.py

diff --git a/.gitignore b/.gitignore
index 68bc17f..36206d8 100644
--- a/.gitignore
+++ b/.gitignore
@@ -1,8 +1,11 @@
+.idea
+
+# Byte-compiled / optimized / DLL files
+__pycache__/
+*.py[cod]
+*.py.class
+
+# C extensions
+*.so

diff --git a/README.md b/README.md
index bc7f717..e1bbfa0 100644
--- a/README.md
+++ b/README.md
@@ -1,2 +1,3 @@
# laba2
+Это учебный проект по дисциплине: основы программной инженерии.
+Выполнил студент группы ПИЖ-6-о-22-1, Матвеев Александр Иванович.
diff --git a/main.py b/main.py
new file mode 100644
index 0000000..e69de29

Sashka@DESKTOP-U4RPSBI MINGW64 ~/laba2 (main)
$ |

```

Рисунок 11 – Вывод команды git show 85022

Контрольные вопросы:

1. Для выполнения истории коммитов необходимо написать команду: (`git log`), так же в ней есть дополнительные аргументы, которые помогут улучшить опыт просмотра коммитов. Например, именной аргумент (`--pretty=oneline`), чтобы вывести все коммиты в одну строку, (`--pretty=format: "%h - %an, %ar : %s"`) помогает настроить вывод под нужный формат. Или (`-p/--patch`), показывает разницу между коммитами. Чтобы увидеть сокращенную статистику, можно использовать (`--stat`). (`--graph`) нужен для вывода истории коммитов в виде графа.
2. Чтобы ограничить вывод необходимо использовать аргумент (-“число”).
3. Для изменения уже существующего коммита необходимо использовать команду (`git commit –amend`).
4. Для отмены индексации файла нужно использовать (`git reset head “имя файла”`).
5. Чтобы откатить несохраненные изменения в файле, достаточно использовать (`git checkout - - “Имя файла”`).
6. Удаленный репозиторий Git это репозиторий, находящийся на каком-либо сервере. Для работы с другими пользователями необходимо использовать удаленный репозиторий.
7. Для просмотра удаленный репозиторииев нужно использовать команду (`git remote`) и использовать ключ `-v`, если мы хотим посмотреть адреса для чтения и записи, привязанные к репозиторию.
8. Чтобы добавить удаленный репозиторий для выбранного локального нужно: (`git remote add “name” <url>`).
9. Чтобы получить изменения из удаленного репозитория необходимо использовать команды `fetch/pull`, первая в свою очередь забирает данные с

сервера и не сливает с локальной веткой, а pull забирает и автоматически сливает. Для того, чтобы отправить изменения в удаленный репозиторий, необходимо использовать команду push.

10. Чтобы получить больше информации о удаленном репозитории, нужно использовать команду (git remote show origin).

11. Теги необходимы для пометки каких-либо коммитов, как важные.

12. Чтобы добавить легковесный тег, необходимо выполнить команду (git tag <name tag>). Чтобы добавить аннотирующий тег, нужно выполнить команду (git tag -a <name tag>). Для того, чтобы удалить существующий тег: (git tag -d <name tag>). Просмотр тегов в репозитории: (git tag). Чтобы запустить тег в удаленный репозиторий, необходимо: (git push origin <name tag>).

13. Флаг --prune для команд (git pull/fetch) необходим для поддержания локального репозитория в актуальном состоянии, он позволяет удалять ветки, которые больше не существуют в удаленном репозитории.