

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:
Светлакова Софья Андреевна
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Ассистент кафедры инфокоммуникаций
Богданов С. С.

(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.

Порядок выполнения работы

1. Создала репозиторий GitHub.


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 *

Repository name *

 seihposv

/


vsekarvem

 vsekarvem is available.

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


Description (optional)

☒

 **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: **Node**


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. Написала небольшую программу на языке Python.

```
number = int(input("Введите число от 0 до 21: "))
if number < 0 or number > 21:
    print("Число должно быть от 0 до 21")
elif number == 0:
    print("Вы выбрали аркан 'Дурак'")
elif number == 1:
    print("Вы выбрали аркан 'Маг'")
elif number == 2:
    print("Вы выбрали аркан 'Верховная Жрица'")
elif number == 3:
    print("Вы выбрали аркан 'Императрица'")
elif number == 4:
    print("Вы выбрали аркан 'Император'")
elif number == 5:
    print("Вы выбрали аркан 'Иерофант'")
elif number == 6:
    print("Вы выбрали аркан 'Влюбленные'")
elif number == 7:
    print("Вы выбрали аркан 'Колесница'")
elif number == 8:
    print("Вы выбрали аркан 'Сила'")
elif number == 9:
    print("Вы выбрали аркан 'Отшельник'")
elif number == 10:
    print("Вы выбрали аркан 'Колесо Фортуны'")
elif number == 11:
    print("Вы выбрали аркан 'Справедливость'")
elif number == 12:
    print("Вы выбрали аркан 'Повешенный'")
elif number == 13:
    print("Вы выбрали аркан 'Смерть'")
elif number == 14:
    print("Вы выбрали аркан 'Умеренность'")
elif number == 15:
    print("Вы выбрали аркан 'Дьявол'")
elif number == 16:
    print("Вы выбрали аркан 'Башня'")
elif number == 17:
    print("Вы выбрали аркан 'Звезда'")
elif number == 18:
    print("Вы выбрали аркан 'Луна'")
elif number == 19:
    print("Вы выбрали аркан 'Солнце'")
elif number == 20:
    print("Вы выбрали аркан 'Суд'")
elif number == 21:
    print("Вы выбрали аркан 'Мир'")
```

Рисунок 2 – Написанная программа

3. Просмотрела историю коммитов.

```
C:\Users\Admin_GGS\vscode>git log --graph --pretty=oneline --abbrev-commit
* 47131d7 (HEAD -> main, tag: harry4) дайте красный диплом
* 896a495 (tag: harry3) ура ура
* a4f8dd8 (tag: harry2, tag: harry) меня называют солнце, потому что на меня больно смотреть
* 43092a3 луна класс
* 1d8730b звезда(я)
* b7dc77e башня топ
* b9649f2 сося класс
\
* bad7463 (origin/main, origin/HEAD) Add files via upload
* f4dcfea Update README.md
* 6fa6a20 Update .gitignore
* | 36574e5 повешенный)
|/
* 92a1368 Initial commit
```

Рисунок 3 – История коммитов

4. Просмотрела содержимое коммитов командой git show.

```
C:\Users\Admin_GGS\vsekarvem>git show HEAD
commit 47131d70bc874e274900ca1deb31dd5663b4b42b (HEAD -> main, tag: harry4)
Author: seihposv <svetlakovasofi@mail.ru>
Date: Mon Nov 13 19:14:49 2023 +0300

    дайте красный диплом

diff --git a/arkantarot.py b/arkantarot.py
index 7e581e2..7a822e9 100644
--- a/arkantarot.py
+++ b/arkantarot.py
@@ -1,6 +1,6 @@
 number = int(input("Введите число от 0 до 21: "))
 if number < 0 or number > 21:
-    print("Число должно быть от 0 до 21")
+    print("Число должно быть от 0 до 21, sorry")
 elif number == 0:
     print("Вы выбрали аркан 'Дурак'")
 elif number == 1:
```

Рисунок 4.1 – Содержимое последнего коммита

```
C:\Users\Admin_GGS\vsekarvem>git show HEAD~1
commit 896a495e02f8dd0c9f7a39eec810113b061d9c8a (tag: harry3)
Author: seihposv <svetlakovasofi@mail.ru>
Date: Mon Nov 13 19:12:37 2023 +0300

    ура ура

diff --git a/arkantarot.py b/arkantarot.py
index a28b0d7..7e581e2 100644
--- a/arkantarot.py
+++ b/arkantarot.py
@@ -41,3 +41,7 @@ elif number == 18:
     print("Вы выбрали аркан 'Луна'")
 elif number == 19:
     print("Вы выбрали аркан 'Солнце'")
+elif number == 20:
+    print("Вы выбрали аркан 'Суд'")
+elif number == 21:
+    print("Вы выбрали аркан 'Мир'")
\ No newline at end of file
```

Рисунок 4.2 – Содержимое предпоследнего коммита

```
C:\Users\Admin_GGS\vsekarvem>git checkout -- arkantarot.py

C:\Users\Admin_GGS\vsekarvem>git commit -m "люблю скфу"
On branch main
Your branch is ahead of 'origin/main' by 8 commits.
  (use "git push" to publish your local commits)

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:   arkantarot.py

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

C:\Users\Admin_GGS\vsekarvem>git reset --hard HEAD~1
HEAD is now at 896a495 ура ура
```

Рисунок 4.3 – Содержимое коммита с указанным хэшем

5. Освоила возможность отката к заданной версии.

```
C:\Users\Admin_GGS\vscode>git show a4f8dd8
commit a4f8dd8d03f6f657e83532e5646e727612f95429 (tag: harry2, tag: harry)
Author: seihposv <svetlakovasofi@mail.ru>
Date: Mon Nov 13 18:50:31 2023 +0300

    меня называют солнце, потому что на меня больно смотреть

diff --git a/arkantarot.py b/arkantarot.py
index 87e1f53..a28b0d7 100644
--- a/arkantarot.py
+++ b/arkantarot.py
@@ -39,3 +39,5 @@ elif number == 17:
     print("Вы выбрали аркан 'Звезда'")
 elif number == 18:
     print("Вы выбрали аркан 'Луна'")
+elif number == 19:
+    print("Вы выбрали аркан 'Солнце'")
```

Рисунок 5 – Откат к предпоследнему коммиту

Ответы на контрольные вопросы

1. С помощью команды `git log`. Существуют множество дополнительных опций вот некоторые из них:
 - `-p` или `-patch` – показывает разницу, внесенную в каждый коммит;
 - `--stat` – позволяет увидеть сокращенную статистику;
 - `--pretty` – эта опция меняет формат вывода;
 - `--since` и `--until` – опции для ограничения вывода по времени
2. С помощью команды `git log` и его аргументов, например `--since`.
3. С помощью команды `git commit --amend`.
4. С помощью команды `git reset HEAD`.
5. С помощью команды `git checkout -- <file>`.
6. Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети.
7. С помощью команды `git remote -v`.
8. С помощью команды `git remote add <shortname> <url>`.
9. Получение изменение – `git fetch [remote-name]`, отправка изменений `git push <remote-name> <branch-name>`.
10. С помощью команды `git remote show <remote>`.
11. Тэги Git - это ссылки на определенные коммиты в истории разработки. Они используются для пометки определенных версий или моментов в вашем проекте. Тэги обычно используются для обозначения релизов или важных этапов в разработке.
12. Для просмотра тэгов – `git tag`, для создания аннотированного тэга – `git tag -a v1.4 -m "сообщение"`, для отправки тэга на удаленный сервер – `git push origin <tagname>`, для удаления тэгов – `git tag -d <tagname>`.
13. Флаг `--prune` в командах `git fetch` и `git push` используется для удаления удаленных веток или тэгов, которые больше не существуют на удаленном репозитории. Если выполнить команду `git fetch --prune`, Git удалит локальные ссылки на удаленные ветки и тэги, которые были удалены на удаленном репозитории. Если выполнить команду `git push --prune`, Git удалит

удаленные ветки и тэги на удаленном репозитории, которых больше нет в вашем локальном репозитории.