

УНИВЕРСИТЕТ ИТМО

Факультет «инфокоммуникационных технологий»
Направление подготовки «Программирование в инфокоммуникационных
системах»

Лабораторная работа №2
«Использование Git и Gulp для решения задач web-разработки»

Выполнила:
Егорова Валерия Игоревна
Группа №3323
Проверила:
Марченко Елена Вадимовна

Санкт-Петербург
2024

Цель работы

Познакомиться со средствами для автоматизации задач веб-разработки, изучить систему контроля версий Git и программу для организации и обработки задач Gulp.

Ход работы

Задание 1

В первом задании было необходимо установить систему контроля версий git. Установочный файл был скачан с официального сайта <https://git-scm.com/>. Далее по инструкции установщика был установлен git (см. рисунок 1).

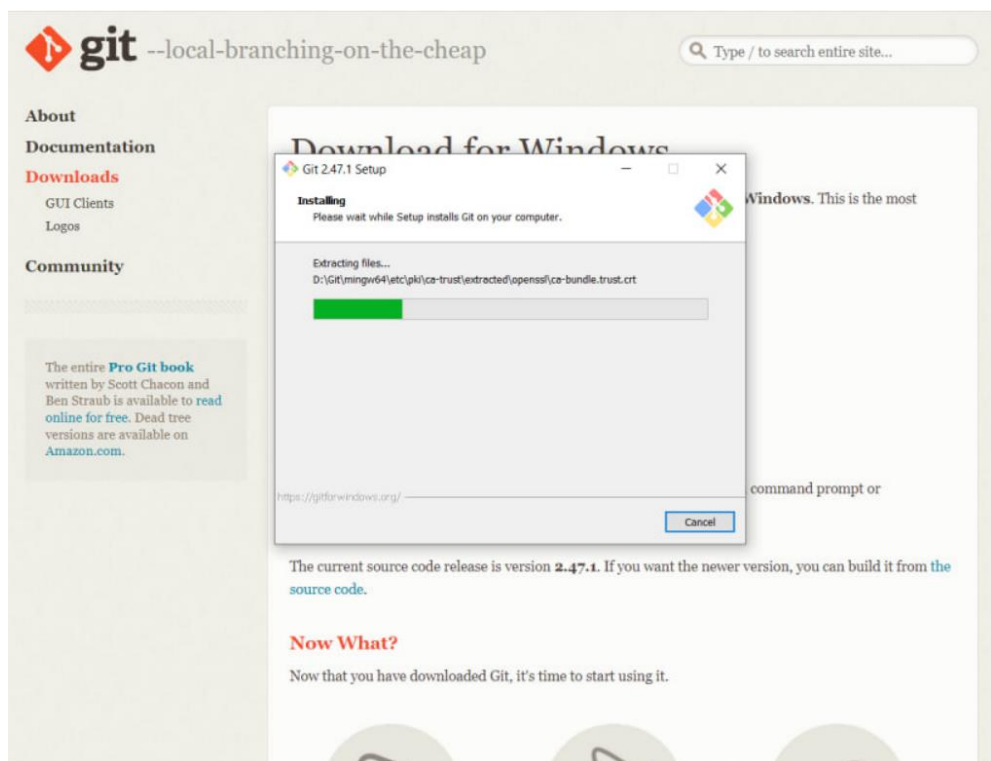


Рисунок 1 – Установщик git

При помощи команды `git -v` можно проверить наличие установленной версии git и ее версию. Удостоверимся, что установка прошла успешно (см. рисунок 2).

```
C:\Users\samsz>git -v
git version 2.47.1.windows.1
C:\Users\samsz>
```

Рисунок 2 – Команда git -v

Далее переходим в рабочую директорию и создаём в текущем каталоге новый подкаталог с именем .git, содержащий все необходимые файлы репозитория — структуру Git репозитория (см. рисунок 3).

```
C:\Users\samsz>cd C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1>git init
Initialized empty Git repository in C:/Users/samsz/OneDrive/Рабочий стол/hw/5/web-npora/practice1/lab1/.git/
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1>
```

Рисунок 3 – Инициализация Git репозитория

Добавим под версионный контроль файлы из первой лабораторной работы. Воспользуемся командой `git add *`, чтобы добавить все файлы, и проверим результат командой `git status` (см. рисунок 4).

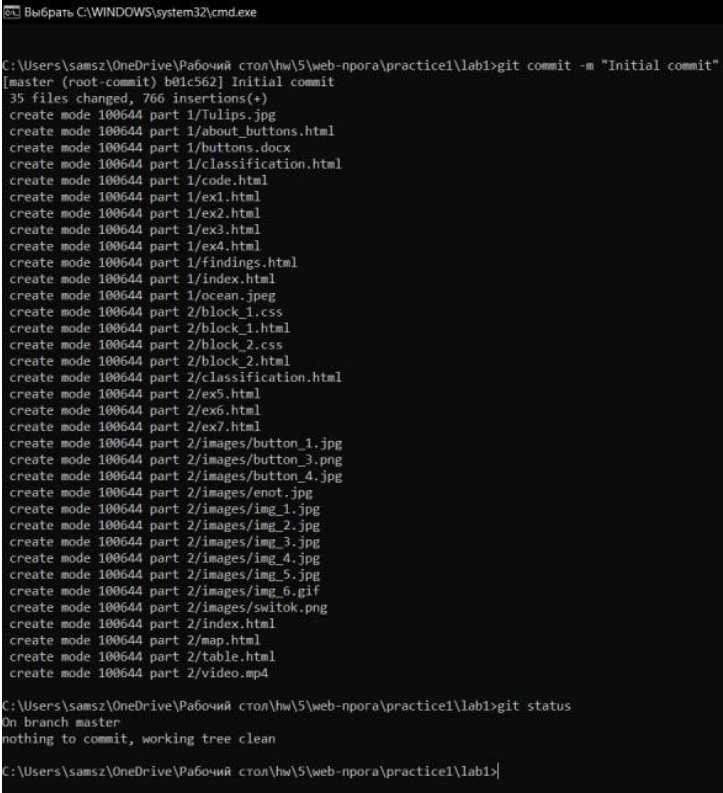
```
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1>git add *
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   part 1/Tulips.jpg
    new file:   part 1/about_buttons.html
    new file:   part 1/buttons.docx
    new file:   part 1/classification.html
    new file:   part 1/code.html
    new file:   part 1/ex1.html
    new file:   part 1/ex2.html
    new file:   part 1/ex3.html
    new file:   part 1/ex4.html
    new file:   part 1/findings.html
    new file:   part 1/index.html
    new file:   part 1/ocean.jpeg
    new file:   part 2/block_1.css
    new file:   part 2/block_1.html
    new file:   part 2/block_2.css
    new file:   part 2/block_2.html
    new file:   part 2/classification.html
    new file:   part 2/ex5.html
    new file:   part 2/ex6.html
    new file:   part 2/ex7.html
    new file:   part 2/images/button_1.jpg
    new file:   part 2/images/button_3.png
    new file:   part 2/images/button_4.jpg
    new file:   part 2/images/enot.jpg
    new file:   part 2/images/img_1.jpg
    new file:   part 2/images/img_2.jpg
    new file:   part 2/images/img_3.jpg
    new file:   part 2/images/img_4.jpg
    new file:   part 2/images/img_5.jpg
    new file:   part 2/images/img_6.gif
    new file:   part 2/images/switok.png
    new file:   part 2/index.html
    new file:   part 2/map.html
    new file:   part 2/table.html
    new file:   part 2/video.mp4
```

Рисунок 4 – Добавление файлов в первый коммит

Осуществим первый коммит с сообщением “Initial commit”, а затем при помощи команды `git status` удостоверимся, что все изменения попали в коммит (см. рисунок 5).



```
Выбрать C:\WINDOWS\system32\cmd.exe

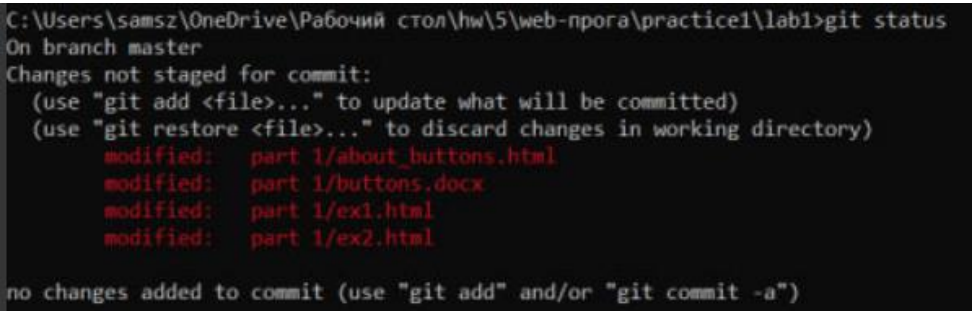
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1>git commit -m "Initial commit"
[master (root-commit) b01c562] Initial commit
35 files changed, 766 insertions(+)
create mode 100644 part 1/Tulips.jpg
create mode 100644 part 1/about_buttons.html
create mode 100644 part 1/buttons.docx
create mode 100644 part 1/classification.html
create mode 100644 part 1/code.html
create mode 100644 part 1/ex1.html
create mode 100644 part 1/ex2.html
create mode 100644 part 1/ex3.html
create mode 100644 part 1/ex4.html
create mode 100644 part 1/findings.html
create mode 100644 part 1/index.html
create mode 100644 part 1/ocean.jpeg
create mode 100644 part 2/block_1.css
create mode 100644 part 2/block_1.html
create mode 100644 part 2/block_2.css
create mode 100644 part 2/block_2.html
create mode 100644 part 2/classification.html
create mode 100644 part 2/ex5.html
create mode 100644 part 2/ex6.html
create mode 100644 part 2/ex7.html
create mode 100644 part 2/images/button_1.jpg
create mode 100644 part 2/images/button_3.png
create mode 100644 part 2/images/button_4.jpg
create mode 100644 part 2/images/enot.jpg
create mode 100644 part 2/images/img_1.jpg
create mode 100644 part 2/images/img_2.jpg
create mode 100644 part 2/images/img_3.jpg
create mode 100644 part 2/images/img_4.jpg
create mode 100644 part 2/images/img_5.jpg
create mode 100644 part 2/images/img_6.gif
create mode 100644 part 2/images/switok.png
create mode 100644 part 2/index.html
create mode 100644 part 2/map.html
create mode 100644 part 2/table.html
create mode 100644 part 2/video.mp4

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1>git status
On branch master
nothing to commit, working tree clean

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1>
```

Рисунок 5 – Первый коммит

Сделаем изменения в файлах и проверим `git status` (см. рисунок 6).



```
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1>git status
On branch 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:   part 1/about_buttons.html
        modified:   part 1/buttons.docx
        modified:   part 1/ex1.html
        modified:   part 1/ex2.html

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

Рисунок 6 – `git status` после изменения в файлах

Добавим в следующий коммит файлы `about_buttons.html` и `buttons.docx`. Проверим `git status`. Осуществим коммит и убедимся, что изменения в файлах `about_buttons.html` и `buttons.docx` добавлены в следующий коммит (см. рисунок 7).

```

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git add about_buttons.html

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git add buttons.docx

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   about_buttons.html
        modified:   buttons.docx

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:   ex1.html
        modified:   ex2.html

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git commit -m "Buttons info fix"
[master e11d8f4] Buttons info fix
2 files changed, 1 insertion(+), 1 deletion(-)

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git status
On branch 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:   ex1.html
        modified:   ex2.html

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

```

Рисунок 7 – Добавление about_buttons.html и buttons.docx в следующий КОММИТ

Аналогично сделаем два коммита для изменений в файлах ex1.html и ex2.html (см. рисунки 8–9).

```

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git status
On branch 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:   ex1.html
        modified:   ex2.html

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

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git add ex1.html

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git commit -m "Fix ex1"
[master 2ed80d5] Fix ex1
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git status
On branch 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:   ex2.html

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

```

Рисунок 8 – Коммит ex1.html


```

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git status
On branch 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:   ex2.html

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

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git add ex2.html

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git commit -m "Fix ex2"
[master 8a02e5b] Fix ex2
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git status
On branch master
nothing to commit, working tree clean

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>

```

Рисунок 9 – Коммит ex2.html

При помощи команды `git log` убедимся, что все коммиты выполнены успешно (см. рисунок 10).

```

C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git log
commit 8a02e5bcefa1308a2600772a3915271eed6ecaa5 (HEAD -> master)
Author: ValeriiaEgorova <samszvd02@gmail.com>
Date: Thu Dec 19 02:16:18 2024 +0300

    Fix ex2

commit 2ed80d5fb50f1b72926c4e66ee24913765d15a43
Author: ValeriiaEgorova <samszvd02@gmail.com>
Date: Thu Dec 19 02:15:36 2024 +0300

    Fix ex1

commit e11d8f40e130c17ed37dd3039ff6c744f4fa807b
Author: ValeriiaEgorova <samszvd02@gmail.com>
Date: Thu Dec 19 02:14:09 2024 +0300

    Buttons info fix

commit b01c5620556037a50ac8ed6e99a9e6fc42c45dbf
Author: ValeriiaEgorova <samszvd02@gmail.com>
Date: Thu Dec 19 01:55:41 2024 +0300

    Initial commit

```

Рисунок 10 – Команда git log

Создадим новый удаленный репозиторий GitHub. Он находится по ссылке: <https://github.com/ValeriiaEgorova/web-lab-2>. Свяжем наш локальный репозиторий с удаленным командой `git remote add`. Назовем его `origin` и укажем ссылку на удаленный репозиторий. Проверим подключение командой `git remote -v` (см. рисунок 11).

```
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git remote
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git remote add origin https://github.com/ValeriiaEgorova/web-lab-2.git
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git remote
origin
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git remote -v
origin https://github.com/ValeriiaEgorova/web-lab-2.git (fetch)
origin https://github.com/ValeriiaEgorova/web-lab-2.git (push)
```

Рисунок 11– Соединение с удаленным репозиторием


Отправим коммиты в ветку master на удаленный репозиторий origin командой `git push origin master` (см. рисунок 12).


```
C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\part 1>git push origin master
Enumerating objects: 52, done.
Counting objects: 100% (52/52), done.
Delta compression using up to 8 threads
Compressing objects: 100% (52/52), done.
Writing objects: 100% (52/52), 15.73 MiB | 2.32 MiB/s, done.
Total 52 (delta 10), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (10/10), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/ValeriiaEgorova/web-lab-2/pull/new/master
remote:
To https://github.com/ValeriiaEgorova/web-lab-2.git
 * [new branch]      master -> master
```


Рисунок 12 – Команда `git push origin master`


Проверим, что коммиты появились на GitHub (см. рисунок 13).


Commits

 master

 Commits on Dec 19, 2024

Fix ex2
 ValeriiaEgorova committed yesterday

Fix ex1
 ValeriiaEgorova committed yesterday

Buttons info fix
 ValeriiaEgorova committed yesterday


Initial commit
 ValeriiaEgorova committed yesterday

Рисунок 13 – Коммиты на GitHub

Задание 2

Для установки `gulp` потребуются `node.js`, `npm` и `prx`. Произведем их установку и проверим ее успешность командами `node -v`, `npm -v`, `prx -v` (см. рисунок 14).

```
PS C:\WINDOWS\system32> winget install Schniz.fnm
Найден существующий установленный пакет. Попытка обновления установленного пакета...
Доступные обновления не найдены.
Более новые версии пакетов недоступны в настроенных источниках.
PS C:\WINDOWS\system32> fnm env --use-on-cd | Out-String | Invoke-Expression
>>
PS C:\WINDOWS\system32> fnm use --install-if-missing 22
>>
Installing +[36mNode v22.12.0-[0m (x64)
00:00:14 33.26 MiB/33.26 MiB (2.28 MiB/s, 0s) Using Node v22.12.0
PS C:\WINDOWS\system32> node -v
v22.12.0
PS C:\WINDOWS\system32> npm -v
10.9.0
PS C:\WINDOWS\system32> npx -v
10.9.0
PS C:\WINDOWS\system32>
```

Рисунок 14 – Установка node.js, npm и npx

Установим утилиту командной строки gulp командой `npm install --global gulp-cli`. Далее создадим в папку my-project в рабочей директории и создадим файл package.json командой `npm init` (см. рисунки 15-16).

```
PS C:\WINDOWS\system32> cd 'C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\'
PS C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1> npx mkdirp my-project
PS C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1> cd my-project
```

Рисунок 15 – Создание папки my-project

```
PS C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\my-project> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (my-project)
version: (1.0.0)
license: (ISC)
About to write to C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\my-project\package.json:
{
  "name": "my-project",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "description": ""
}
```

Рисунок 16 – Создание файла package.json

Установим пакет gulp в devDependencies (см. рисунок 17).

```
PS C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-npora\practice1\lab1\my-project> npm install --save-dev gulp
up to date, audited 144 packages in 23s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Рисунок 17 – Установка gulp в devDependencies

Проверим, что gulp установлен успешно (см. рисунок 18).


```
PS C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-прора\practice1\lab1\my-project> gulp -v
CLI version: 3.0.0
Local version: 5.0.0
```

Рисунок 18 – Команда gulp -v

Создадим файл gulpfile.js в директории my-project. И напишем в нем простую задачу для копирования файлов из папки src в папку dist (см. рисунок 19).

```
gulpfile.js > ...
1  const gulp = require('gulp');
2
3  function copyFiles() {
4    return gulp.src('src/*')
5      .pipe(gulp.dest('dist'));
6  }
7
8  exports.default = copyFiles;
```

Рисунок 19 – Код gulpfile.js

Так теперь выглядит структура проекта (см. рисунок 20).

dist	19.12.2024 3:04	Папка с файлами	
node_modules	19.12.2024 2:57	Папка с файлами	
src	19.12.2024 3:03	Папка с файлами	
gulpfile.js	19.12.2024 3:03	Исходный файл J...	1 КБ
package.json	19.12.2024 2:57	JSON File	1 КБ
package-lock.json	19.12.2024 2:57	JSON File	61 КБ

Рисунок 20 – Структура проекта

Запускаем задачу командой gulp и получаем ожидаемый результат. Теперь в папке dist находятся копии файлов из папки src (см. рисунок 21).

```
PS C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-прора\practice1\lab1\my-project> gulp
[ ] Using gulpfile
[ ] Starting 'default'...
[ ] Finished 'default' after
PS C:\Users\samsz\OneDrive\Рабочий стол\hw\5\web-прора\practice1\lab1\my-project> gulp --version
CLI version: 3.0.0
Local version: 5.0.0
```

Рисунок 21 – Запуск задачи

Задание 3

Программа будет написана на языке Python, потому что для него существует множество готовых библиотек для самых разных задач. Для того,

чтобы написать программу-клиент потребуются библиотеки tkinter, webbrowser, threading и time (см. рисунок 22).

```
1 import tkinter as tk
2 from tkinter import messagebox
3 import webbrowser
4 import threading
5 import time
```

Рисунок 22 – Импортируемые библиотеки

Реализуем класс WebPageViewer и его инициализацию. Добавим графический интерфейс, используя библиотеку tkinter (см. рисунок 23).

```
class WebPageViewer:
    def __init__(self, root):
        self.root = root
        self.root.title("Web Page Viewer")

        self.label = tk.Label(root, text="Enter URLs (one per line):")
        self.label.pack()

        self.text_area = tk.Text(root, height=10, width=50)
        self.text_area.pack()

        self.interval_label = tk.Label(root, text="Enter interval (seconds):")
        self.interval_label.pack()

        self.interval_entry = tk.Entry(root)
        self.interval_entry.pack()

        self.start_button = tk.Button(root, text="Start", command=self.start_viewing)
        self.start_button.pack()

        self.stop_button = tk.Button(root, text="Stop", command=self.stop_viewing, state=tk.DISABLED)
        self.stop_button.pack()

        self.running = False
```

Рисунок 23 – Класс WebPageViewer и его метод init

Реализуем метод start_viewing в классе WebPageViewer. Он проверяет ввод пользователя на наличие ошибок, сохраняет список URL и интервал, запускает отдельный поток для последовательного открытия страниц (см. рисунок 24).

```
def start_viewing(self):
    try:
        urls = self.text_area.get(index="1.0", tk.END).strip().split("\n")
        self.urls = [url.strip() for url in urls if url.strip()]
        if not self.urls:
            raise ValueError("No URLs provided.")

        self.interval = float(self.interval_entry.get())
        if self.interval <= 0:
            raise ValueError("Interval must be positive.")

        self.start_button.config(state=tk.DISABLED)
        self.stop_button.config(state=tk.NORMAL)
        self.running = True

        threading.Thread(target=self.view_pages).start()
    except ValueError as e:
        messagebox.showerror(title="Input Error", str(e))
```

Рисунок 24 – Метод start_viewing

Метод `stop_viewing` останавливает процесс открытия страниц и разблокирует элементы интерфейса. А метод `view_pages` последовательно открывает страницы из списка с использованием браузера Google Chrome. Интервал между открытиями определяется пользователем (см. рисунок 25).

```
def stop_viewing(self):
    self.running = False
    self.start_button.config(state=tk.NORMAL)
    self.stop_button.config(state=tk.DISABLED)

1 usage
def view_pages(self):
    chrome_path = "C:/Program Files/Google/Chrome/Application/chrome.exe %s"
    browser = webbrowser.get(chrome_path)

    for url in self.urls:
        if not self.running:
            break
        browser.open(url)
        time.sleep(self.interval)

    self.stop_viewing()
```

Рисунок 25 – Методы `stop_viewing` и `view_pages`

Программа обладает графическим интерфейсом. Текстовое поле позволяет пользователю вводить адреса веб-страниц (по одному на строку). Есть поле ввода для интервала (в секундах) между открытиями страниц. Кнопка Start запускает процесс последовательного открытия URL. Кнопка Stop позволяет остановить выполнение (см. рисунки 26–27).

```
if __name__ == "__main__":
    root = tk.Tk()
    app = WebPageViewer(root)
    root.mainloop()
```

Рисунок 26 – Запуск приложения

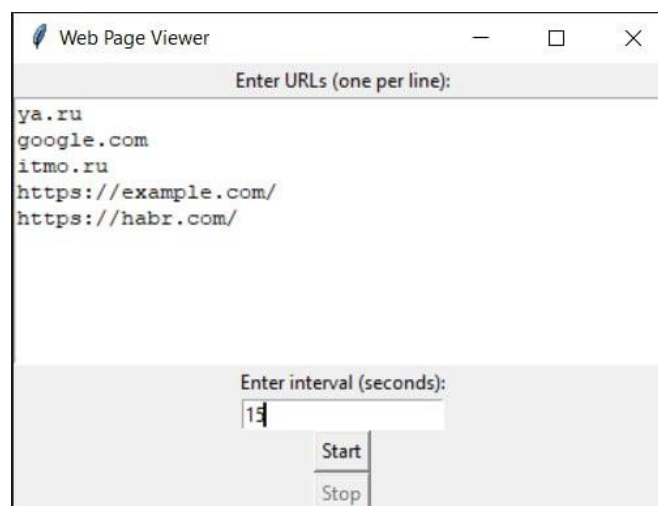


Рисунок 27 – Графический интерфейс

Программа работает следующим образом: пользователь вводит список URL, интервал, нажимает кнопку Start. Программа начинает открывать страницы в Chrome с указанным интервалом, а нажатие кнопки Stop или завершение списка URL останавливает процесс.

Вывод

В ходе выполнения лабораторной работы были изучены основы работы с системой контроля версий Git и программой для организации и обработки задач Gulp. Поставленная в лабораторной работе цель была выполнена успешно.