

Лабораторная работа 1

Воинов Кирилл

Содержание I

1 Цель работы

2 Выполнение лабораторной работы

3 Выводы

Раздел 1

Цель работы

Цель работы

Приобрести практические навыки работы с системой управления версиями Git, языком программирования Julia.

Раздел 2

Выполнение лабораторной работы

Устанавливаю гит и настраиваю его(рис. 1).

```
[voinov_kirill@fedora ~]$ git config --global user.name "voinovkv"
[voinov_kirill@fedora ~]$ git config --global user.email "1132236017@rudn.ru"
[voinov_kirill@fedora ~]$ git config --global core.quotepath false
[voinov_kirill@fedora ~]$ git config --global init.defaultBranch master
[voinov_kirill@fedora ~]$ git config --global core.autocrlf input
[voinov_kirill@fedora ~]$ git config --global core.safecrlf warn
```

Рисунок 1: «Настройка гит»

Авторизуюсь в гит(рис. 3).

```
[voinov_kirill@fedora ~]$ gh auth login
? Where do you use GitHub? GitHub.com

? What is your preferred protocol for Git operations on this host?
SSH
? Generate a new SSH key to add to your GitHub account? Yes

? Enter a passphrase for your new SSH key (Optional):

? Title for your SSH key: GitHub CLI

? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 97C5-B5C5

Press Enter to open https://github.com/login/device in your browser
...
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
```



Создаю и добавляю ключ (рис. 3),(рис. 4).

```
[voinov_kirill@fedora ~]$ gh auth login
? Where do you use GitHub? GitHub.com

? What is your preferred protocol for Git operations on this host?
SSH
? Generate a new SSH key to add to your GitHub account? Yes

? Enter a passphrase for your new SSH key (Optional):

? Title for your SSH key: GitHub CLI

? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 97C5-B5C5

Press Enter to open https://github.com/login/device in your browser
...
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
```

Добавляю ключ в учетную запись гитхаба (рис. 5).

```
[voinov_kirill@fedora ~]$ ssh-add ~/.ssh/id_ed25519-git  
Identity added: /home/voinov_kirill/.ssh/id_ed25519-git (11322360  
17@pfur.ru)  
[voinov_kirill@fedora ~]$ gh ssh-key add ~/.ssh/id_ed25519-git.pu  
b --title 1132236017@pfur.ru  
✓ Public key added to your account
```

Рисунок 5: «Добавление ключа в гитхаб»

Генерирую ключ pgp (рис. 6).

```
[voinov_kirill@fedora ~]$ gpg --full-generate-key

gpg (GnuPG) 2.4.8; Copyright (C) 2025 g10 Code GmbH
This is free software: you are free to change and redistribute it
.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/voinov_kirill/.gnupg' created
Please select what kind of key you want:
(1) RSA and RSA
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(9) ECC (sign and encrypt) *default*
(10) ECC (sign only)
(14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
```



Добавляю PGP ключ на хостинг (рис. 7).

```
[voinov_kirill@fedora ~]$ gpg--list-secret-keys--keyid-format LONG
bash: gpg--list-secret-keys--keyid-format: command not found
[voinov_kirill@fedora ~]$ gpg --list-secret-keys--keyid-format LONG
gpg: invalid option "--list-secret-keys--keyid-format"
[voinov_kirill@fedora ~]$ gpg --list-secret-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:  1  signed:  0  trust: 0-, 0q, 0n, 0m, 0f
, 1u
[keyboxd]
```

Рисунок 7: «Добавление PGP ключа на хостинг»

Настраиваю автоматические подписи коммитов git (рис. 8).

```
[voinov_kirill@fedora ~]$ git config --global user.signingkey 1C7  
81EADC86551DBD2B6722CD69A47F2CE7AF87F  
[voinov_kirill@fedora ~]$ git config --global commit.gpgsign true  
[voinov_kirill@fedora ~]$ git config --global gpg.program $(which  
gpg)
```

Рисунок 8: «Настройка автоматических подписей коммитов git»

Устанавливаю средства разработки (рис. 10).

```
[voinov_kirill@fedora ~]$ sudo dnf -y group install development-tools
```

Рисунок 9: “ Установка средств разработки”

Устанавливаю средства разработки (рис. 10).

```
[voinov_kirill@fedora ~]$ sudo dnf -y group install development-tools
```

Рисунок 10: “ Установка средств разработки”

Клонирую репозиторий и инициализирую курс (рис. 11).

```
[voinov_kirill@fedora Моделирование]$ git clone  
ssh://git@gitverse.ru:2222/voinovkv/2026_mathmod.git
```

Рисунок 11: “Клонирование репозитория”

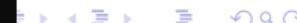
Отправляю файлы на сервер(рис. 12) и настраиваю git-flow.

```
[voinov_kirill@fedora mathmod]$ git push
Enumerating objects: 80, done.
Counting objects: 100% (80/80), done.
Compressing objects: 100% (62/62), done.
Writing objects: 100% (78/78), 703.06 KiB | 4.45 MiB/s, done.
Total 78 (delta 25), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (25/25), completed with 1 local ob
ject.
To github.com:voinovkv/study_2026_mod.git
  304a992..623eb36  master -> master
```

Рисунок 12: «Отправка файлов на сервер»

Открываю терминал, запускаю Julia и устанавливаю необходимые пакеты(рис. 13)

```
julia> Pkg.add("DrWatson")
Installing known registries into 'C:\Users\voino\.julia'
    Added 'General' registry to C:\Users\voino\.julia\registries
    Updating registry at 'C:\Users\voino\.julia\registries\General.tom
l'
Resolving package versions...
Installed Scratch ━━━━━━ v1.3.0
Installed Zstd_jll ━━━━━━ v1.5.7+1
Installed JLLWrappers ━━━━ v1.7.1
Installed ChunkCodecLibZlib ━━ v1.0.0
Installed ChunkCodecLibZstd ━━ v1.0.0
Installed OrderedCollections ━━ v1.8.1
Installed Preferences ━━━━ v1.5.1
Installed ScopedValues ━━━━ v1.5.0
Installed Requires ━━━━ v1.3.1
Installed ChunkCodecCore ━━ v1.0.1
Installed PrecompileTools ━━ v1.3.3
Installed UnPack ━━━━ v1.0.2
Installed FileIO ━━━━ v1.18.0
Installed MacroTools ━━━━ v0.5.16
Installed DrWatson ━━━━ v2.19.1
Installed JLD2 ━━━━ v0.6.3
Installed HashArrayMappedTries - v0.2.0
```



Создаю тестовый скрипт scripts/test_setup.jl(рис. 14)

```
##!/usr/bin/env julia
## test_setup.jl
using DrWatson
@quickactivate "project"
println(" ✅ Проект активирован: ", projectdir())
## Проверка пакетов
packages = [
    "DrWatson",
    "DifferentialEquations",
    "Plots",
    "DataFrames",
    "CSV",
    "JLD2",
    "Literate",
    "IJulia",
    "BenchmarkTools",
    "Quarto"
]
println("\nПроверка пакетов:")
for pkg in packages
    try
        eval(Meta.parse("using $pkg"))
        println(" ✓ $pkg")
    catch e
        println(" X $pkg: ошибка загрузки")
    end
end
## Проверка путей
println("\nСтруктура проекта:")
```

Создаю следующий скрипт (scripts/01_exponential_growth.jl) (рис. 16)

```
using DrWatson
@quickactivate "project"

using DifferentialEquations
using Plots
using DataFrames

function exponential_growth!(du, u, p, t)
    a = p
    du = a * u
end

u0 = [1.0]                                # начальная популяция
a = 0.3                                     # скорость роста
tspan = (0.0, 10.0)                         # временной интервал

prob = ODEProblem(exponential_growth!, u0, tspan, a)
sol = solve(prob, Tsit5(), saveat=0.1)

plot(sol, label="u(t)", xlabel="Время t", ylabel="Популяция u",
      title="Экспоненциальный рост (a = $a)", lw=2, legend=:topleft)

savefig(plotsdir("exponential_growth_alpha=$a.png"))

df = DataFrame(t=sol.t, u=first.(sol.u))
println("Первые 5 строк результатов:")
println(first(df, 5))
```

Создаю генератор отчетов (рис. 18)

```
#!/usr/bin/env julia
# tangle.jl - Генератор отчетов из Literate-скриптов
# Использование: julia tangle.jl <путь_к_скрипту>

using DrWatson
@quickactivate # Активирует текущий проект DrWatson
using Literate
function main()
    if length(ARGS) == 0
        println("""
        Использование: julia tangle.jl <путь_к_скрипту>

        Примеры:
        julia tangle.jl scripts/lab1.jl
        """)
        return
    end

    script_path = ARGS[1]
    if !isfile(script_path)
        error("Файл не найден: $script_path")
    end
    # Пути и имена
    script_dir = dirname(script_path)
    script_name = splitext(basename(script_path))[1]
    println("Генерация из: $script_path")
    # Чистый скрипт (без комментариев)
    scripts_dir = scriptsdir(script_name)
    Literate.script(script_path, scripts_dir; credit=false)
    println(" ✓ Чистый скрипт: $(scripts_dir)/$(script_name).jl")
    # Quarto-документ
    quarto_dir = projectdir("markdown", script_name);
    Literate.markdown(script_path, quarto_dir;
                      flavor=Literate.QuartoFlavor(),
                      name=script_name, credit=false)
    println(" ✓ Quarto: $(quarto_dir)/$(script_name).qmd")
    # Jupyter notebook]
    notebooks_dir = projectdir("notebooks", script_name)
    Literate.notebook(script_path, notebooks_dir, name=script_name; execute=false, credit=false)
```

В каталоге отчёта в файл _quarto.yml включаю поддержку кода julia.
(рис. 20)

```
## Julia support
engine: julia
julia:
  exeflags: ["--project=../project"]
```

Рисунок 20: «Поддержка кода julia»

Раздел 3

Выводы

Выводы

В итоге этой лабораторной работы я приобрел знания о языке программирования Julia и настроил рабочее пространство лабораторных работ.