

Лабораторная работа №8

Выполнил Урусов Максим

ИВТ-б-о-21-1

Цель: приобретение навыков по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.

1. Создал общедоступный репозиторий на GitHub с MIT

Create a new repository



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

Owner * Repository name *

 MaksimUrusov / Laba2.13 

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

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:

Skip this step if you're importing an existing repository.

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

Add .gitignore


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


.gitignore template: Python ▾

Choose a license

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

License: MIT License ▾

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

2. Выполнил клонирование созданного репозитория.

```
D:\Program Files\Git>git clone https://github.com/MaksimUrusov/Laba-2.13.git
Cloning into 'Laba-2.13'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

3. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.

```
# Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
# Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm

### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
```

4. Организовал репозиторий в соответствии с моделью ветвления git-flow.

```
D:\Program Files\Git\Laba-2.13>git flow init
Fatal: Working tree contains unstaged changes. Aborting.

D:\Program Files\Git\Laba-2.13>git reset --hard
HEAD is now at 85057cd Initial commit

D:\Program Files\Git\Laba-2.13>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [D:/Program Files/Git/Laba-2.13/.git/hooks]
```

5. Создал проект пайчарм

Новый том (D:) > REP6 > LB2.11 >

Имя	Дата изменения	Тип	Размер
.idea	18.10.2022 1:22	Папка с файлами	
doc	03.11.2022 4:19	Папка с файлами	
.gitignore	18.10.2022 1:22	Файл "GITIGNORE"	6 КБ
LICENSE	03.11.2022 4:01	Файл	2 КБ
README.md	03.11.2022 4:01	Файл "MD"	1 КБ

6. Проработал примеры лабораторной работы.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':

    from math import*
    print(cos(3.14))
    print(sin(76))

__name__ == '__main__'

prim5 x
C:\ProgramData\Anaconda3\python.exe "D:\Program Files\Git\Laba-2.13\prim5.py"
-0.9999987317275395
0.5661076368981803

Process finished with exit code 0
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import math as matematika

if __name__ == '__main__':

    print(matematika.cos(90))
```

me__ == '__main__'

prim3 x

C:\ProgramData\Anaconda3\python.exe "D:\Program Files\Git\Laba-2.13\prim3.py"
-0.4480736161291701

Process finished with exit code 0

```
a-2.13 D:\ 1  #!/usr/bin/env python3
doc        2  # -*- coding: utf-8 -*-
gitignore  3
LICENSE    4      import math
prim1.py   5
prim2.py   6  ►  if __name__ == '__main__':
prim3.py   7      I
prim4.py   8      print(math.factorial(5))
prim5.py
prim6.py
README.rst
```

ernal Librar if __name__ == '__main__'

prim1 x

C:\ProgramData\Anaconda3\python.exe "D:\Program Files\Git\Laba-2.13\prim1.py"
120

Process finished with exit code 0

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
```

I

```
if __name__ == '__main__':

    import fincalc.simper
    fv = fincalc.simper.fv(pv, i, n)
    import fincalc.simper as sp
    fv = sp.fv(pv, i, n)
    from fincalc import simper
    fv = simper.fv(pv, i, n)
```

```
__name__ == '__main__'
```

prim6 ×

C:\ProgramData\Anaconda3\python.exe "D:\Program Files\Git\Laba-2.13\prim6.py"

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
```

```
if __name__ == '__main__':
```

```
    from math import cos, sin, pi
    print(cos(3.14) + sin(60) + pi)
```

```
__name__ == '__main__'
```

prim4 ×

C:\ProgramData\Anaconda3\python.exe "D:\Program Files\Git\Laba-2.13\prim4.py"
1.836783300760037

Process finished with exit code 0

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import math, datetime

if __name__ == '__main__':

    print(math.cos(math.pi/4))
    print(datetime.date(2022, 11, 29))
```

ame__ == '__main__'

prim2 x

C:\ProgramData\Anaconda3\python.exe "D:\Program Files\Git\Laba-2.13\prim2.py"

0.7071067811865476

2022-11-29

Process finished with exit code 0

7. Зафиксируйте сделанные изменения в репозитории.(после создания веток не запустил, поэтому не работало)

```
D:\REP6\LB2.11>git push --all
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (16/16), 92.88 KiB | 7.14 MiB/s, done.
Total 16 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/Arsen445/LB2.11.git
30b5e5e..c836547 develop -> develop

D:\REP6\LB2.11>git status
On branch develop
nothing to commit, working tree clean

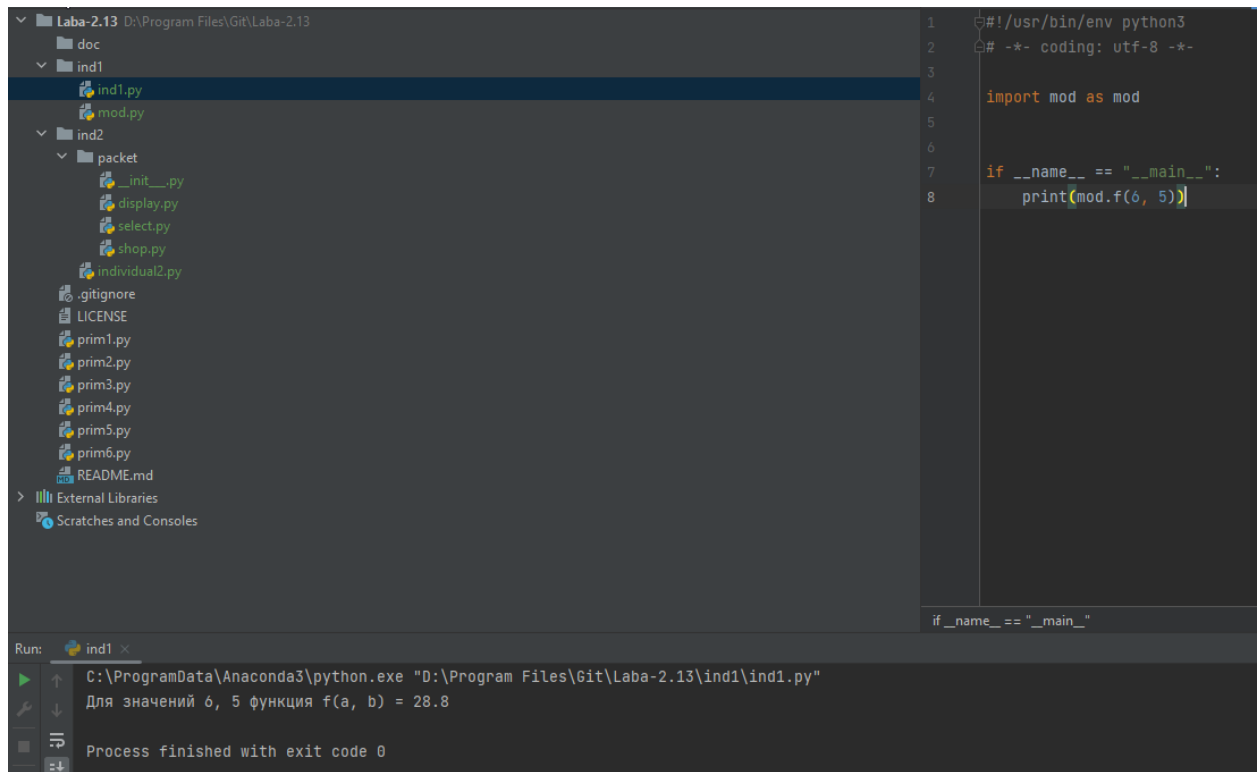
D:\REP6\LB2.11>_
```

8. Решите индивидуальное задание.

Оформил как ветвление последнее действие

Задание 1

Выполнить индивидуальное задание лабораторной работы 2.11, оформив все функции программы в виде отдельного модуля. Разработанный модуль должен быть подключен в основную программу с помощью одного из вариантов команды `import`.



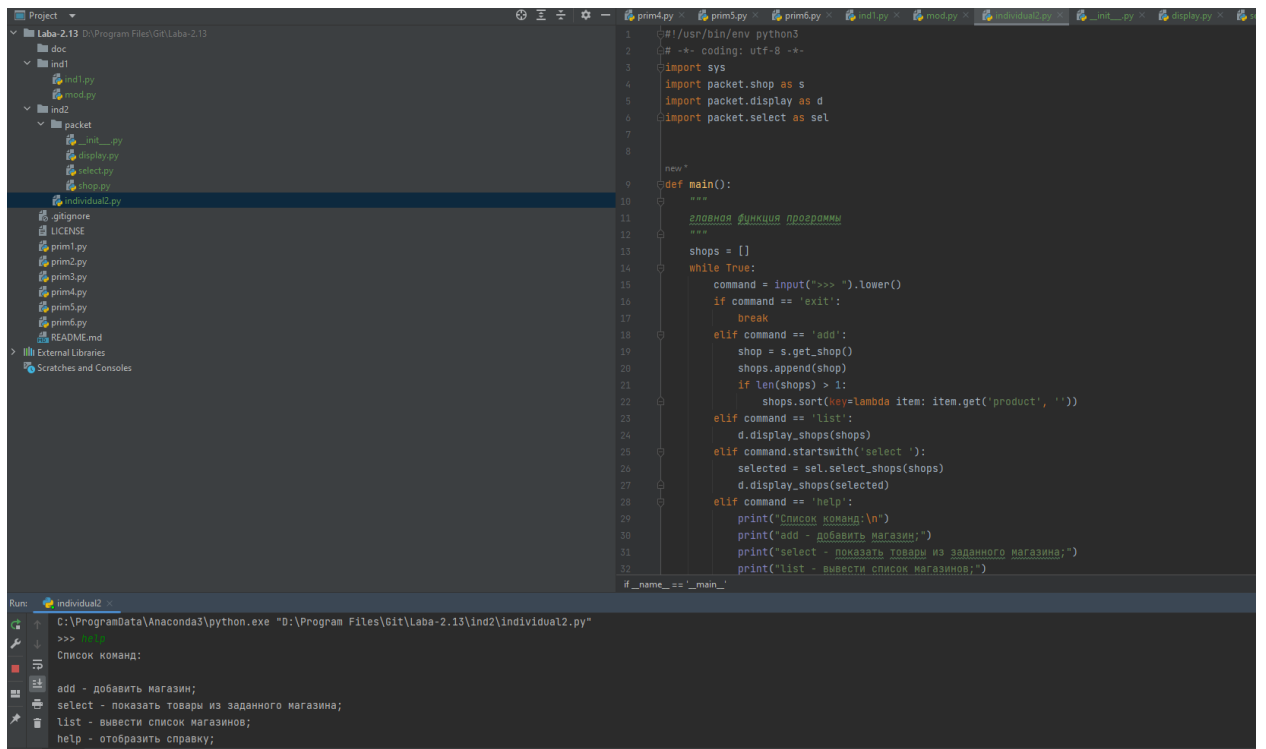
The screenshot shows an IDE with a project named 'Laba-2.13'. The file explorer on the left shows a directory structure: 'doc', 'ind1' (containing 'ind1.py' and 'mod.py'), 'ind2', and 'packet' (containing '.__init__.py', 'display.py', 'select.py', 'shop.py', and 'individual2.py'). Other files include '.gitignore', 'LICENSE', 'prim1.py' through 'prim6.py', and 'README.md'. The code editor on the right shows the content of 'ind1.py':

```
1 #!/usr/bin/env python3
2 #-*- coding: utf-8 -*-
3
4 import mod as mod
5
6
7 if __name__ == "__main__":
8     print(mod.f(6, 5))
```

At the bottom, the 'Run' console shows the execution of 'ind1.py' using the Anaconda3 Python interpreter. The output is: 'Для значений 6, 5 функция f(a, b) = 28.8'. The process finished with exit code 0.

Задание 2

Выполнить индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Разработанный пакет должен быть подключен в основную программу с помощью одного из вариантов команды `import`. Настроить соответствующим образом переменную `__all__` в файле `__init__.py` пакета. Номер варианта уточнить у преподавателя.



9. Зафиксируйте сделанные изменения в репозитории.

```
D:\REP8\LB2.13>git add .
D:\REP8\LB2.13>git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   "doc/~$\320\233\320\2218.docx"
        new file:   "doc/\320\233\320\2218.docx"
        new file:   ind1/indv2.py
        new file:   ind1/mod.py
        new file:   ind2/ind.py
        new file:   ind2/pacet/__init__.py
        new file:   ind2/pacet/display_trains.py
        new file:   ind2/pacet/get_train.py
        new file:   ind2/pacet/help_f.py
        new file:   ind2/pacet/select_trains.py
        new file:   prim1.py
        new file:   prim2.py
        new file:   prim3.py
        new file:   prim4.py
        new file:   prim5.py
        new file:   prim6.py

D:\REP8\LB2.13>git commit -m "1.2"
[develop 4003e1c] 1.2
16 files changed, 227 insertions(+)
create mode 100644 "doc/~$\320\233\320\2218.docx"
create mode 100644 "doc/\320\233\320\2218.docx"
create mode 100644 ind1/indv2.py
create mode 100644 ind1/mod.py
create mode 100644 ind2/ind.py
create mode 100644 ind2/pacet/__init__.py
create mode 100644 ind2/pacet/display_trains.py
create mode 100644 ind2/pacet/get_train.py
create mode 100644 ind2/pacet/help_f.py
create mode 100644 ind2/pacet/select_trains.py
```

10. Выполните слияние ветки для разработки с веткой main/master.

```

D:\REP8\LB2.13>git merge develop
Updating 7c0a384..4003e1c
Fast-forward
 "doc/~$\320\233\320\2218.docx" | Bin 0 -> 162 bytes
 "doc/\320\233\320\2218.docx" | Bin 0 -> 404686 bytes
 ind1/indv2.py | 12 ++++++
 ind1/mod.py | 29 ++++++
 ind2/ind.py | 49 ++++++
 ind2/pacet/__init__.py | 4 +++++
 ind2/pacet/display_trains.py | 35 ++++++
 ind2/pacet/get_train.py | 15 ++++++
 ind2/pacet/help_f.py | 11 ++++++
 ind2/pacet/select_trains.py | 18 ++++++
 prim1.py | 8 ++++++
 prim2.py | 9 ++++++
 prim3.py | 8 ++++++
 prim4.py | 8 ++++++
 prim5.py | 9 ++++++
 prim6.py | 12 ++++++
16 files changed, 227 insertions(+)
create mode 100644 "doc/~$\320\233\320\2218.docx"
create mode 100644 "doc/\320\233\320\2218.docx"
create mode 100644 ind1/indv2.py
create mode 100644 ind1/mod.py
create mode 100644 ind2/ind.py
create mode 100644 ind2/pacet/__init__.py
create mode 100644 ind2/pacet/display_trains.py
create mode 100644 ind2/pacet/get_train.py
create mode 100644 ind2/pacet/help_f.py
create mode 100644 ind2/pacet/select_trains.py
create mode 100644 prim1.py
create mode 100644 prim2.py
create mode 100644 prim3.py
create mode 100644 prim4.py
create mode 100644 prim5.py
create mode 100644 prim6.py

```

Контрольные вопросы:

1. Что является модулем языка Python? - файл с расширением .py.

Модули предназначены для того, чтобы в них хранить часто используемые функции, классы, константы и т. п.

2. Какие существуют способы подключения модулей в языке Python?

```
import имя_модуля
```

```
import имя_модуля as новое_имя
```

```
from имя_модуля import имя_объекта
```

```
from имя_модуля import *
```

```
from имя_модуля import имя_объекта as псевдоним_объекта
```

3. Что является пакетом языка Python? – это каталог, включающий в себя другие каталоги и модули, но при этом дополнительно содержащий файл `__init__.py` .

4. Каково назначение файла `__init__.py` ? - для формирования пространства имен, что позволяет работать с модулями через указание уровня вложенности (через точку).

5. Каково назначение переменной `__all__` файла `__init__.py` ?