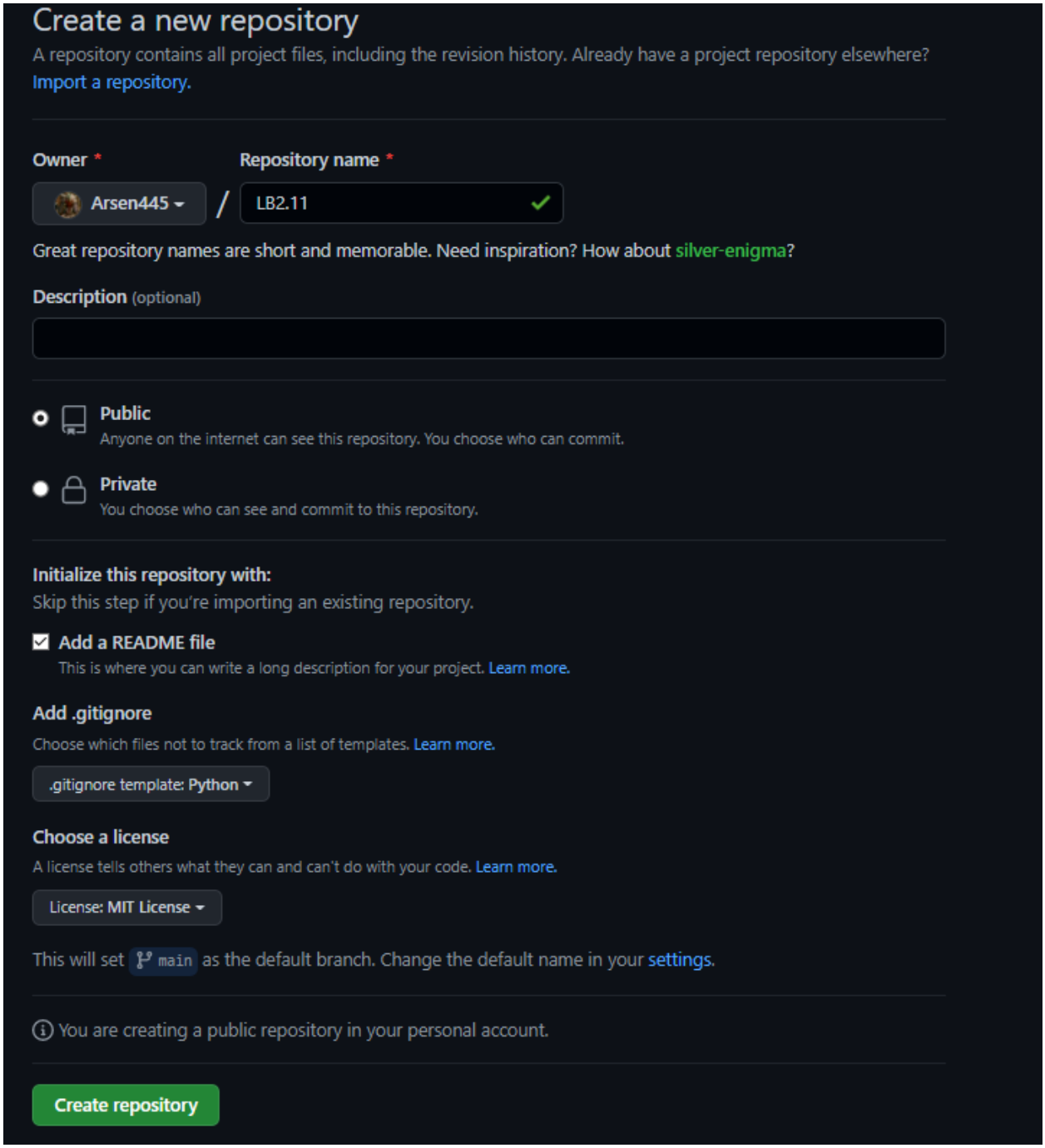
Лабораторная работа №6

Выполнил Эсеналиев Арсен

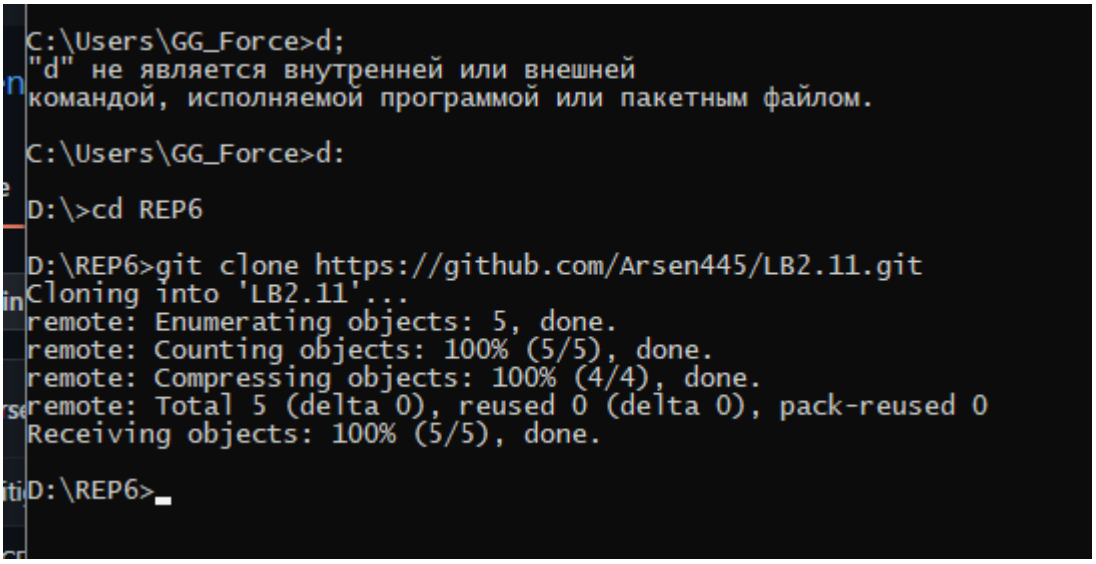
ИВТ-б-о-21-1

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



2. Выполнил клонирование созданного репозитория.



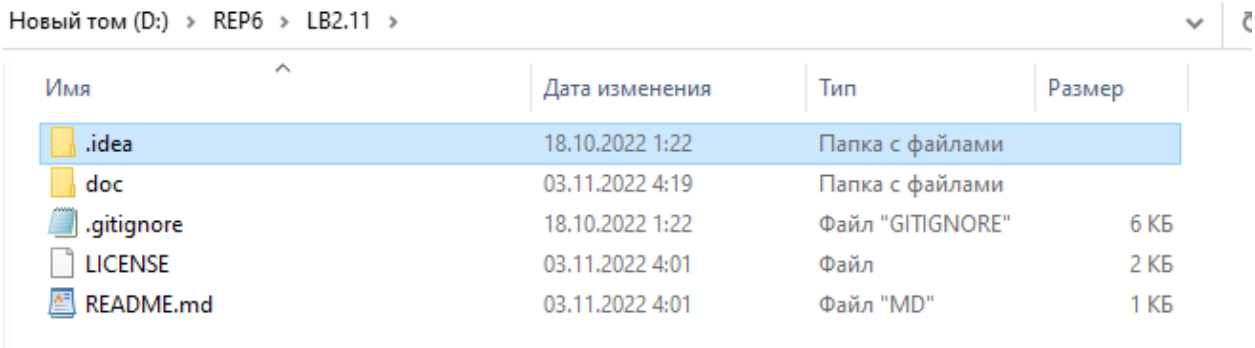
3. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.

```
# 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/
```

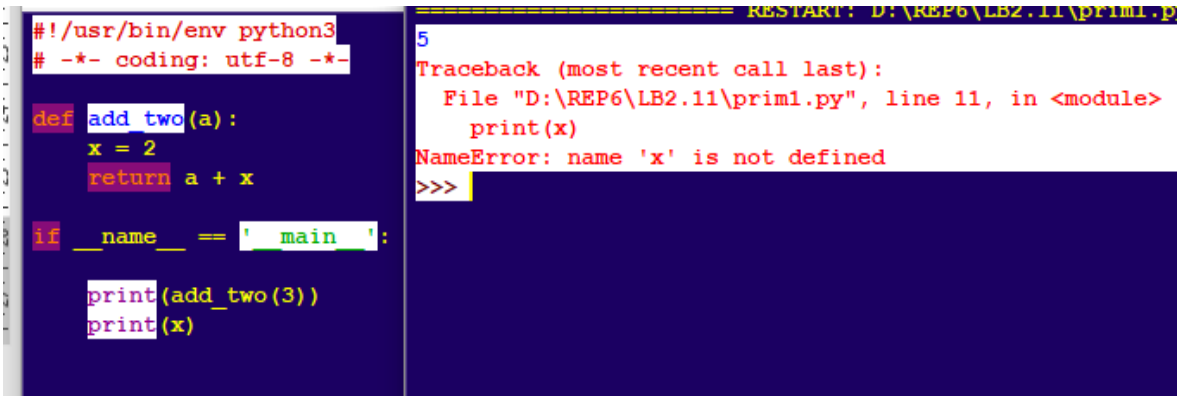
4. Организовал репазиторий в соответствии с моделью ветвления git-flow.

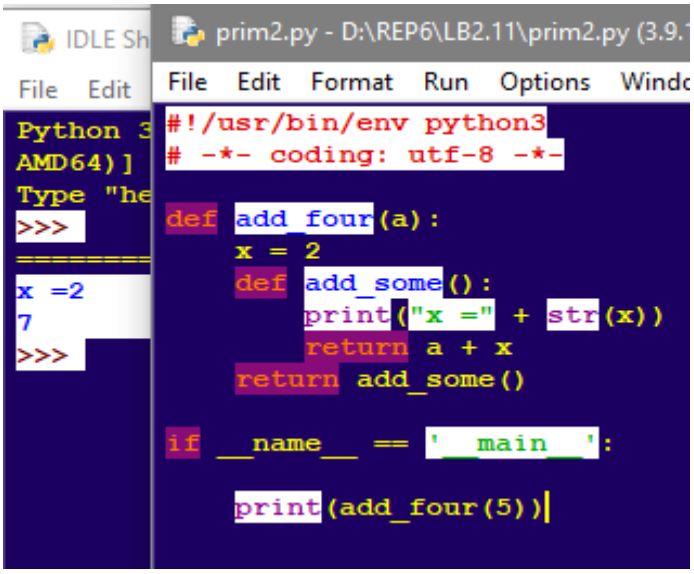
```
D:\REP6\LB2.11>git flow init

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:/REP6/LB2.11/.git/hooks]

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

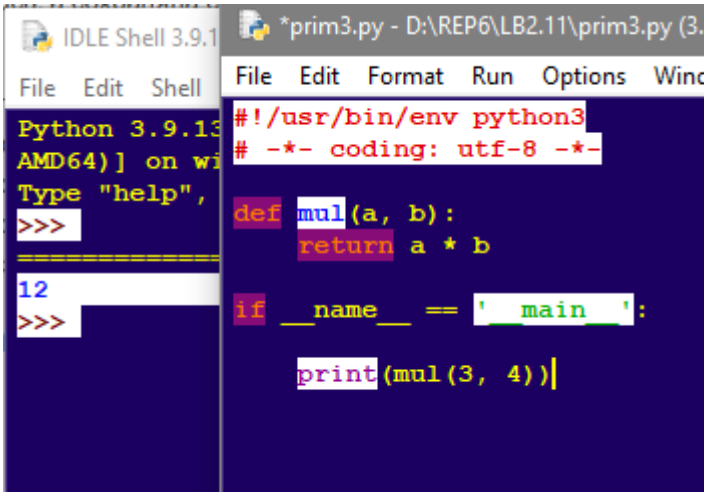
5. Создал проэкт пайчарм

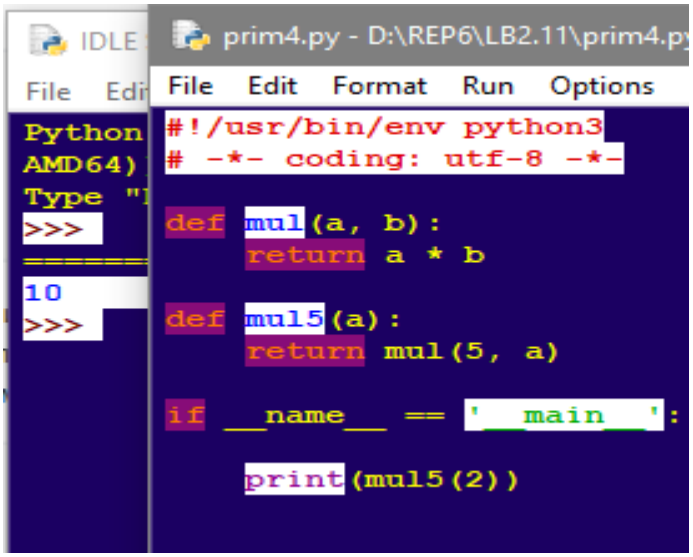


6. Проработал примеры лабораторной работы.







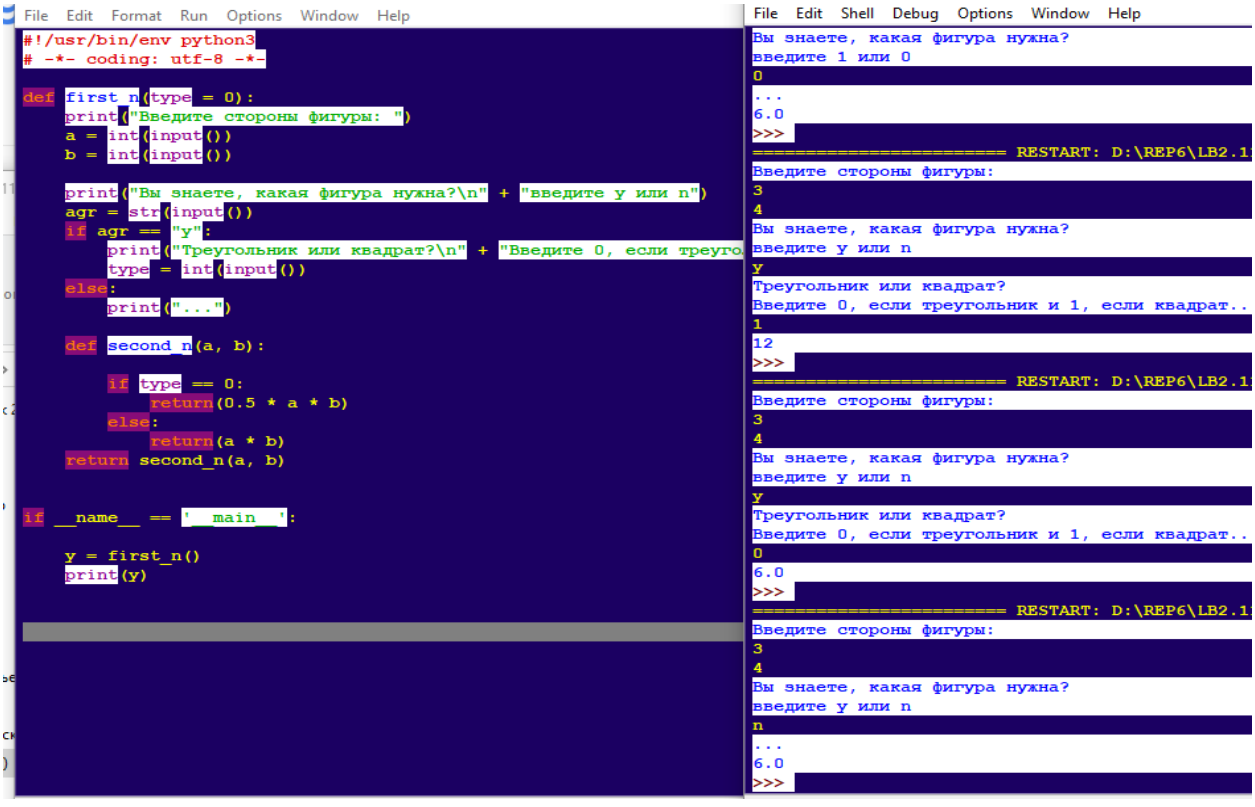


7. Зафиксируйте сделанные изменения в репозитории.(после создания веток не запушил, поэтому не работало)

```
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 ивидуальное задание.

   Используя замыкания функций, объявите внутреннюю функцию, которая на основе двух параметров вычисляет площадь фигуры. Какой именно фигуры: треугольника или прямоугольника, определяется параметром type внешней функции. Если type принимает значение 0, то вычисляется площадь треугольника, а иначе – прямоугольника. По умолчанию параметр type должен быть равен 0. Вычисленное значение должно возвращаться внутренней функцией. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.



9. Зафиксируйте сделанные изменения в репозитории.

```
nothing added to commit but untracked files present (use "git add" to track)

D:\REP6\LB2.11>git add --all

D:\REP6\LB2.11>git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   ind.py


D:\REP6\LB2.11>git commit -m "e1"
[develop f888dbd] e1
 1 file changed, 31 insertions(+)
 create mode 100644 ind.py

D:\REP6\LB2.11>git push --all
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 632 bytes | 632.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Arsen445/LB2.11.git
   c836547..f888dbd  develop -> develop

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

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

```
D:\REP6\LB2.11>git checkout main
Unlink of file 'doc/Лб6.docx' failed. Should I try again? (y/n) y
Unlink of file 'doc/Лб6.docx' failed. Should I try again? (y/n) n
warning: unable to unlink 'doc/Лб6.docx': Invalid argument
Updating files: 100% (13/13), done.
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

D:\REP6\LB2.11>git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        doc/

nothing added to commit but untracked files present (use "git add" to track)

D:\REP6\LB2.11>git add --all

D:\REP6\LB2.11>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   "doc/\320\233\320\2216.docx"


D:\REP6\LB2.11>git merge develop
Updating 71773ee..f888dbd
Fast-forward
 .idea/.name                                        |   1 +
 .idea/LB2.8.iml                                    |   8 +++++++
 .idea/inspectionProfiles/profiles_settings.xml     |   6 +++++
 .idea/modules.xml                                  |   8 +++++++
 .idea/vcs.xml                                      |   6 +++++
 "doc/~$\320\233\320\2216.docx"                     | Bin 0 -> 162 bytes
 "doc/\320\233\320\2216.docx"                       | Bin 0 -> 566932 bytes
 ind.py                                             |  31 +++++++++++++++++++++++++++++
 prim1.py                                           |  15 +++++++++++++++
 prim2.py                                           |  13 +++++++++++++
 prim3.py                                           |   9 +++++++++
```

Контрольные вопросы:

1. **Что такое замыкание?**

   “замыкание (closure) в программировании — это функция, в теле которой присутствуют ссылки на переменные, объявленные вне тела этой функции в окружающем коде и не являющиеся ее параметрами.”

2. **Как реализованы замыкания в языке программирования Python?**

3. **Что подразумевает под собой область видимости Local?**

   Эту область видимости имеют переменные, которые создаются и используются внутри функций.

4. **Что подразумевает под собой область видимости Enclosing?**

   Суть данной области видимости в том, что внутри функции могут быть вложенные функции и локальные переменные, так вот локальная переменная функции для ее вложенной функции находится в enclosing области видимости.

5. **Что подразумевает под собой область видимости Global?**

   Переменные области видимости global – это глобальные переменные уровня модуля (модуль – это файл с расширением .py)

6. **Что подразумевает под собой область видимости Build-in?**

   Уровень Python интерпретатора. В рамках этой области видимости находятся функции open, len и т. п., также туда входят исключения. Эти сущности доступны в любом модуле Python и не требуют предварительного импорта. Built-in – это максимально широкая область видимости

7. **Как использовать замыкания в языке программирования Python?**

```
>>> def fun1(a):
        x = a * 3
        def fun2(b):
            nonlocal x
            return b + x
        return fun2

>>> test_fun = fun1(4)

>>> test_fun(7)
19
```

8. **Как замыкания могут быть использованы для построения иерархических данных?**

```
>>> tpl = lambda a, b: (a, b)
```

Если мы передадим в качестве аргументов числа, то, получим простой кортеж.

```
>>> a = tpl(1, 2)
>>> a
(1, 2)
```

Эту операцию можно производить не только над числами, но и над сущностями, ей же и порожденными.

```
>>> b = tpl(3, a)
>>> b
(3, (1, 2))

>>> c = tpl(a, b)
>>> c
((1, 2), (3, (1, 2)))
```