

Гайд с прошлой работы все также актуален:

1. Необходим Python с версией ≥ 3.8

Проверить версию командой:

```
python3 -V
```

2. Установить менеджер пакетов командой:

```
sudo apt install python3-pip
```

3. (Необязательно) Установить модуль `venv`, для создания виртуального окружения командой:

```
sudo apt install -y python3-venv
```

Далее нужно находится в корневой папке проекта

4. (Необязательно) Создать и активировать виртуальное окружение командами:

```
python3 -m venv env
```

```
source env/bin/activate
```

Для выхода из виртуального окружения команда:

```
deactivate
```

5. Установить зависимости командой:

```
pip3 install -r requirements.txt
```

6. Запустить тесты командой:

```
pytest test.py
```

7. Документация в папке `documentation\bitarray`. Открыть файл `index.html`

Документация формировалась в `numpydoc` стиле с помощью библиотеки `pdoc3` и команды:

```
pdoc --html bitarray --output-dir documentation --force
```

Касаемо третьей работы – использовалась библиотека **pytest-cov**. Опять же, либо в виртуальном окружении, либо в папке с проектом установить её можно командой:

```
pip3 install pytest-cov
```

Либо повторно установить все зависимости из файла `requirements.txt` (см. 5 пункт).

Однако устанавливать её вовсе не обязательно. Во время выполнения лабораторной работы процент покрытия инструкций контролировался с помощью автоматически формируемых отчетов командой:

```
pytest -v test.py --cov=bitarray.core --cov=bitarray.services --cov-report=html
```

Формируемые командой `html`-файлы находятся в директории `htmlcov`, файл `index.html`. Можно его открыть и убедиться, что удалось достичь 100% покрытия инструкций тестами. В качестве большого метода был написан метод, который принимает на вход строку с двумя операндами и самой операцией. Документация была также дополнена

Coverage report: 100%

Module	statements	missing	excluded	coverage
bitarray\core.py	97	0	0	100%
bitarray\services.py	11	0	0	100%
Total	108	0	0	100%

coverage.py v6.1.1, created at 2021-11-10 14:09 +0700

Coverage for **bitarray\core.py**: 100%

97 statements

97 run

0 missing

0 excluded

```
1 # a b c d e f g h i j k l m n o p q r s t u v w x y z
2 """
3 Class implementation
4 """
5 from __future__ import annotations
6
7 import operator
8 import re
9 from typing import List
10 from typing import Optional
11 from typing import Union
12
13 from bitarray.services import is_bool
14 from bitarray.services import is_char_0_or_1
15 from bitarray.services import is_int_0_or_1
16
17
```