

Программный продукт «Программный актуарный расчетный комплекс»

Инструкция по установке экземпляра ПО

Содержание

О программном продукте	3
Установка необходимых компонентов, развертывание экземпляра ПО	3
Установка Python	3
Установка экземпляра ПО ПАРК	4
Запуск ПО ПАРК	6
Контактные лица	8

О программном продукте

Программный актуарный расчетный комплекс (ПАРК) предназначен для проведения актуарных расчетов для построения денежных потоков с тестированием входных данных для широкого спектра задач, вариантов использования и условий НПО.

Установка необходимых компонентов, развертывание экземпляра ПО

Установка Python

Для работы необходим установленный Python (допустимы версии 3.9 – 3.11.9).

Для установки Python требуется загрузить установщик с сайта разработчиков. Для этого необходимо перейти по ссылке (для версии Python 3.11.8): <https://www.python.org/downloads/release/python-3118/>, скачать файл (Windows installer (64-bit)) и запустить установщик (см. Рис. 1 - 3)



Рис. 1 Установка Python

В ходе установки необходимо установить флаги:

- Use admin privileges when installing py.exe
- Add python.exe to PATH

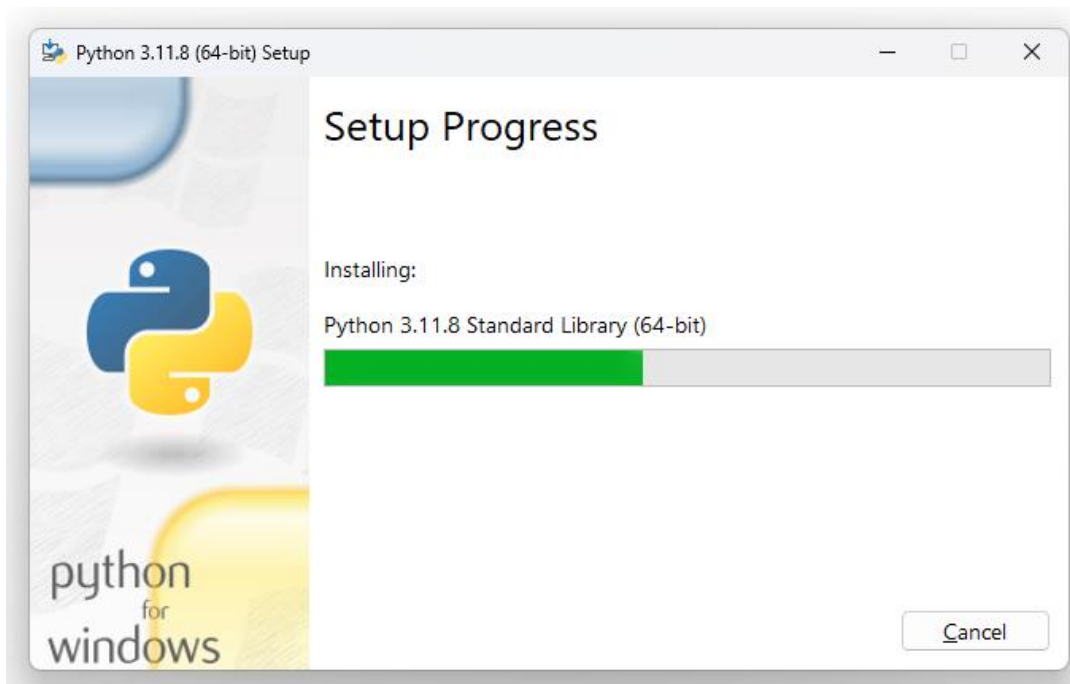


Рис. 2 Процесс установки Python

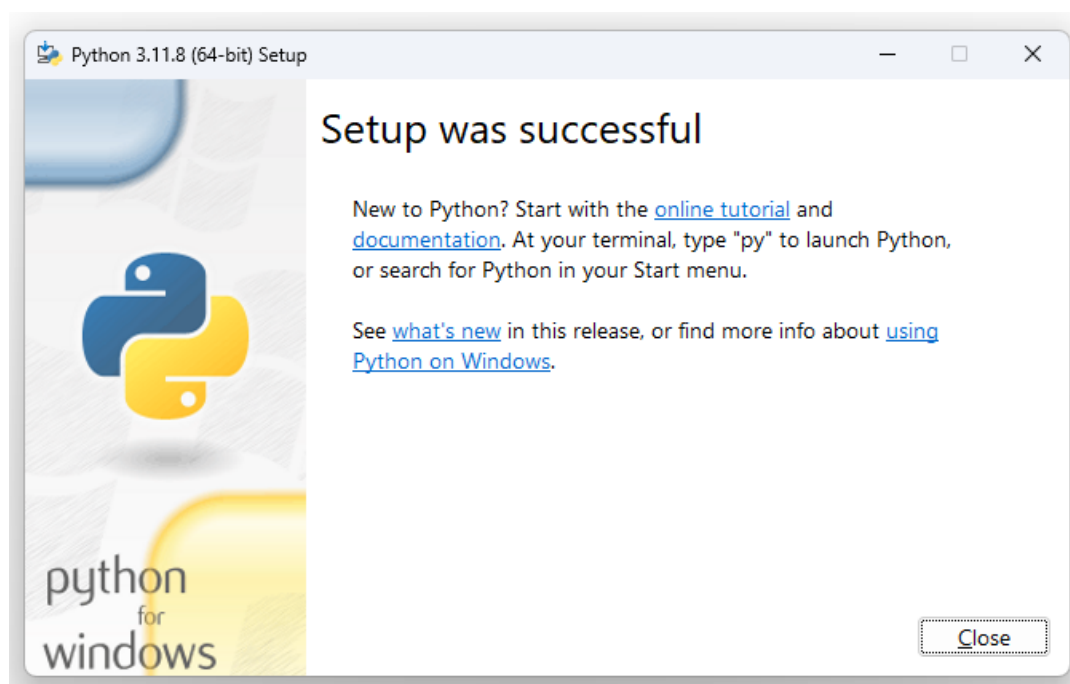


Рис. 3 Завершение установки Python

Если после окончания установки будет выведено предложение «Disable Path length limit», то его можно проигнорировать.

Установка экземпляра ПО ПАРК

1. Архив дистрибутива **Park_distr.zip** распаковать в папку (например, в **C:\PARK**)
2. Установить виртуальное окружение (см. Рис. 4).

Для этого требуется открыть Командную строку в папке, с распакованными файлами дистрибутива, например открыв в проводнике эту папку (см. Рис. 4) и введя в адресной строке команду `cmd` (см. Рис. 5). Также Командную строку можно открыть, нажав сочетание клавиш `Win + R` и введя команду `cmd` в окне «Выполнить», затем перейти к папке, выполнив в окне командной строки команду `cd C:\Park` (где `C:\Park` – папка с файлами ПАРК).

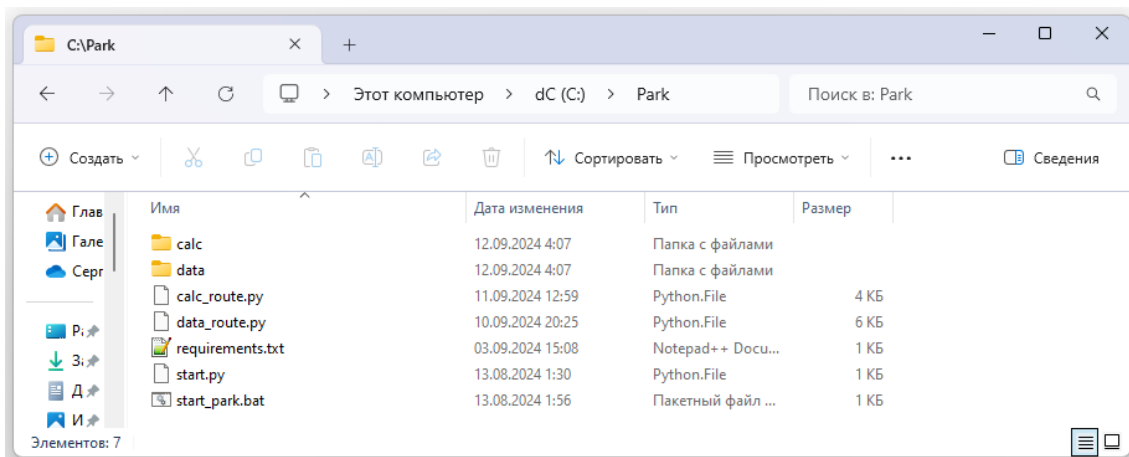


Рис. 4 Папка с файлами ПАРК

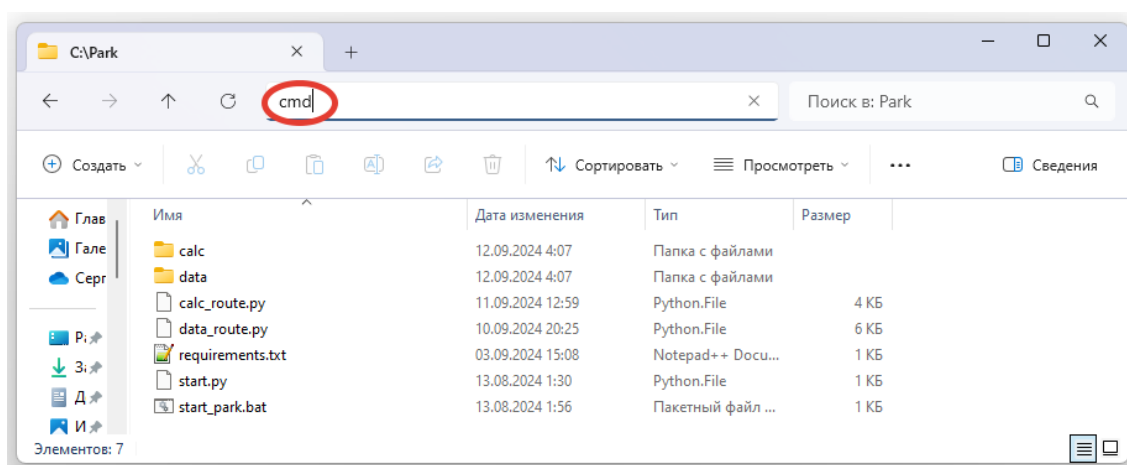


Рис. 5 Запуск командной строки (`cmd`) из адресной строки проводника

Для установки виртуального окружения в командной строке (см. Рис. 6) выполнить команду

```
python -m venv .env
```

3. Перейти в виртуальное окружение командой (см. Рис. 6)

```
.env\Scripts\activate
```

Установить дополнительные пакеты Python командой

```
pip install -r requirements.txt
```

```
C:\Windows\System32\cmd.e x + v - □ x
Microsoft Windows [Version 10.0.22631.3880]
(с) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\PARK>python -m venv .env

C:\PARK>.env\Scripts\activate

(.env) C:\PARK>pip install -r requirements.txt|
```

Рис. 6 Установка виртуального окружения и дополнительных пакетов Python

```
C:\Windows\System32\cmd.e x + v - □ x
Microsoft Windows [Version 10.0.22631.4169]
(с) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\PARK>python -m venv .env

C:\PARK>.env\Scripts\activate

(.env) C:\PARK>pip install -r requirements.txt
Collecting greenlet==3.0.3 (from -r requirements.txt (line 1))
  Using cached greenlet-3.0.3-cp311-cp311-win_amd64.whl.metadata (3.9 kB)
Collecting numpy==1.26.4 (from -r requirements.txt (line 2))
  Using cached numpy-1.26.4-cp311-cp311-win_amd64.whl.metadata (61 kB)
Collecting pandas==2.2.2 (from -r requirements.txt (line 3))
  Using cached pandas-2.2.2-cp311-cp311-win_amd64.whl.metadata (19 kB)
Collecting python-dateutil==2.9.0.post0 (from -r requirements.txt (line 4))
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz==2024.1 (from -r requirements.txt (line 5))
  Using cached pytz-2024.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting six==1.16.0 (from -r requirements.txt (line 6))
  Using cached six-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)
Collecting SQLAlchemy==2.0.32 (from -r requirements.txt (line 7))
  Using cached SQLAlchemy-2.0.32-cp311-cp311-win_amd64.whl.metadata (9.8 kB)
Collecting typing_extensions==4.12.2 (from -r requirements.txt (line 8))
  Using cached typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Collecting tzdata==2024.1 (from -r requirements.txt (line 9))
  Using cached tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached greenlet-3.0.3-cp311-cp311-win_amd64.whl (292 kB)
Using cached numpy-1.26.4-cp311-cp311-win_amd64.whl (15.8 MB)
Using cached pandas-2.2.2-cp311-cp311-win_amd64.whl (11.6 MB)
Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Using cached pytz-2024.1-py2.py3-none-any.whl (505 kB)
Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Using cached SQLAlchemy-2.0.32-cp311-cp311-win_amd64.whl (2.1 MB)
Using cached typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Installing collected packages: pytz, tzdata, typing_extensions, six, numpy, greenlet, SQLAlchemy, python-dateutil, pandas
Successfully installed SQLAlchemy-2.0.32 greenlet-3.0.3 numpy-1.26.4 pandas-2.2.2 python-dateutil-2.9.0.post0 pytz-2024.1 six-1.16.0 typing_extensions-4.12.2 tzdata-2024.1

[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(.env) C:\PARK>|
```

Рис. 7 Окончание установки

После окончания установки (см. Рис.7) окно командной строки можно закрыть.

Запуск ПО ПАРК

1. Перед запуском необходимо разместить исходные данные и настройки исходных данных и расчетов в папках с установленной структурой.

В комплекте с ПАРК поставляется архив «Типовые настройки.zip» с типовыми настройками и примером исходных данных. Рекомендуется распаковать архив в

папку, которую предполагается использовать для работы, например C:\Data (см. Рис. 8 - 9):

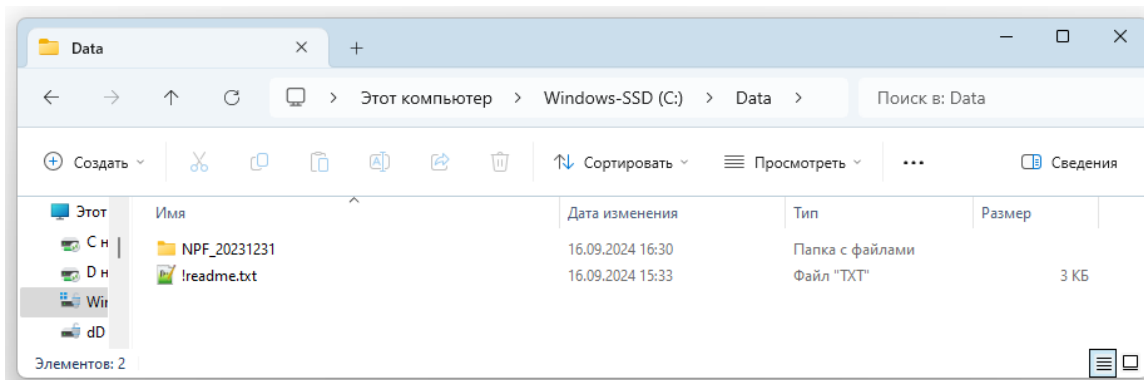


Рис. 8 Содержимое архива с типовыми настройками

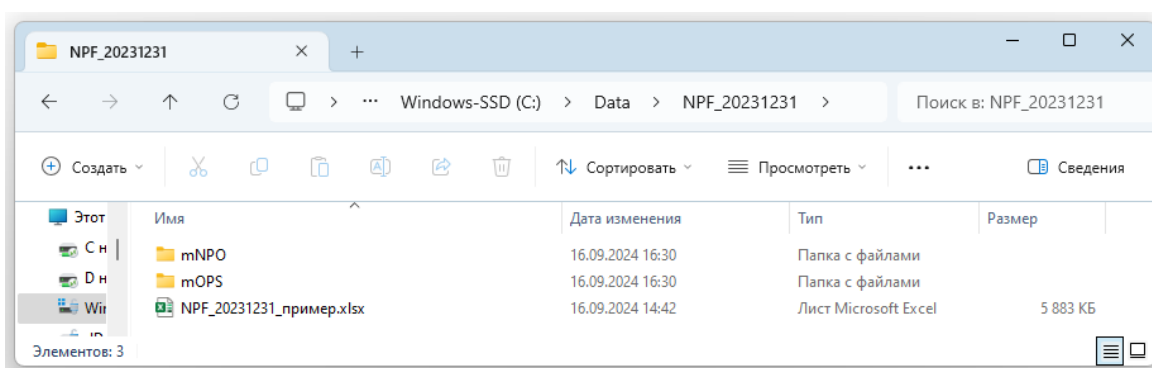


Рис. 9 Папки с типовыми настройками и примером исходных данных

2. Необходимо указать расположение папок (например папок с типовыми настройками) и тип задачи в файле **park_config.ini** (находится в папке с установленным экземпляром ПАРК):

main_folder – рабочая папка

data_folder – папка с данными и настройками (находится в **main_folder**)

task – задача, может принимать значения: **test** (тестирование данных) или **calc** (проведение расчета).

Например, если папки с типовыми настройками для негосударственного пенсионного обеспечения и обязательного пенсионного страхования (*mNPO* и *mOPS*) находятся в C:\Data\NPF_20231231 (Рис. 9), то для проведения расчетов по обязательному пенсионному страхованию **park_config.ini** должен содержать строки (Рис. 10):

```
park_config.ini
1 [config]
2 main_folder = C:\data\NPF_20231231
3 data_folder = mOPS
4 task = calc
5
```

Рис. 10 Пример park_config.ini

Для тестирования данных `task = test`

3. Запуск осуществляется файлом `start_park.bat`

Контактные лица

Контакты технических специалистов, которые могут проконсультировать по процессу развёртывания и настройки экземпляра ПО и его функционирования:

ФИО	Должность	Контакт
Парамонов Сергей Михайлович	Руководитель практики анализа данных и автоматизации	sergey.paramonov@embc.ru +7(915)268-5839