Национальный исследовательский ядерный университет «МИФИ»

(Московский Инженерно-Физический Институт)

Кафедра №42 «Криптология и кибербезопасность»

**Лабораторная работа №2-3**

**«**Базовые задачи администрирования СУБД**»**

Тимин Александр Б21-515 (2024г.)

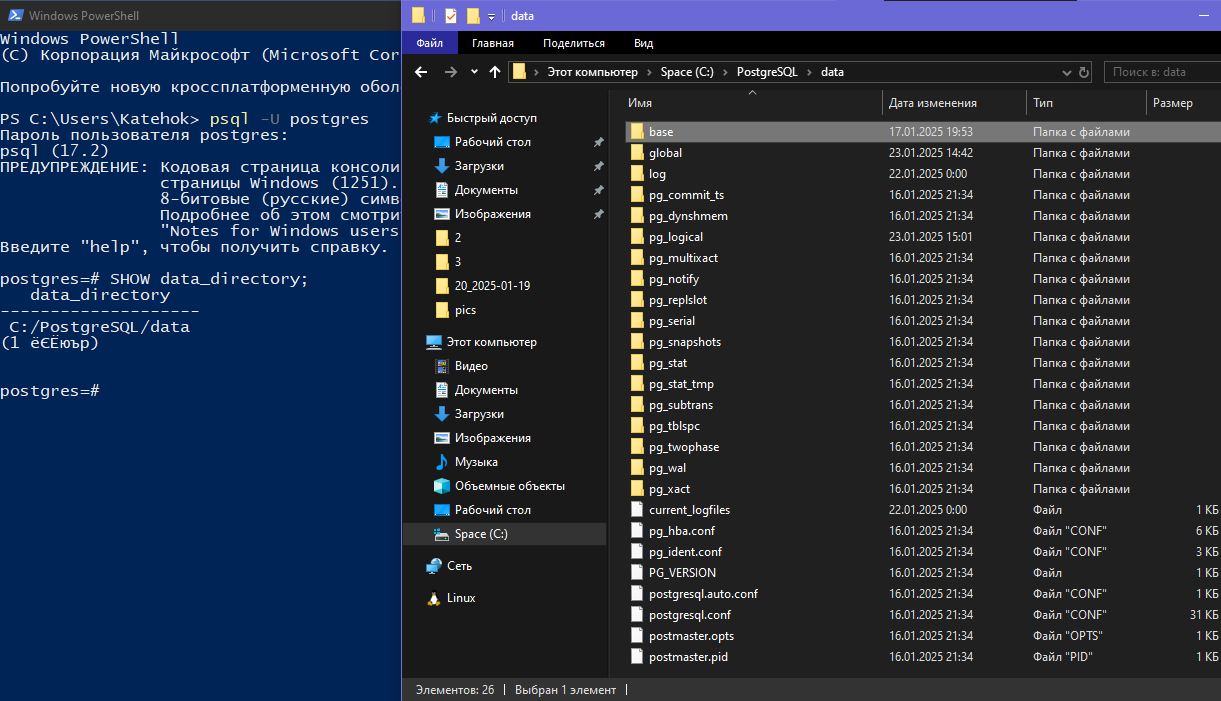
**1 Где БД?**

Подключимся к БД, используя PowerShell:

psql -U postgres

Запросим путь до директории с данными:

SHOW data\_directory;

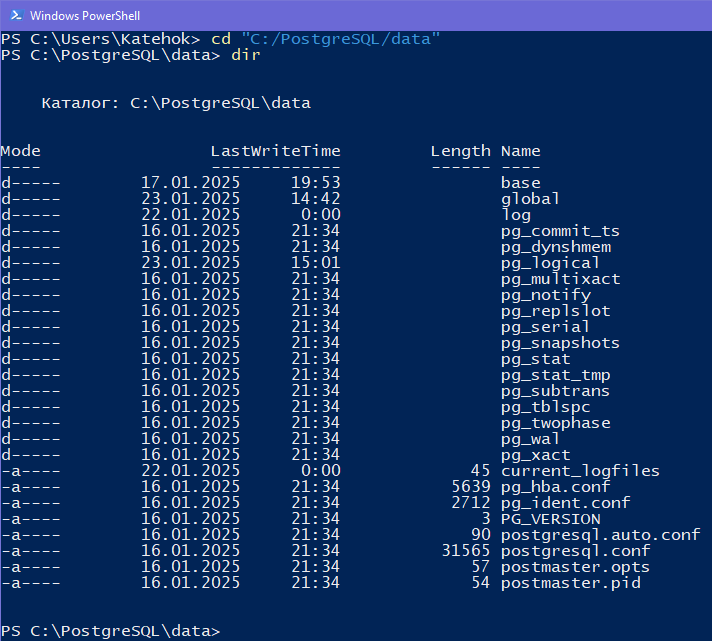


У меня данные БД postgres хранятсяпо пути C:\PostgreSQL\data.

**2 Что в директории?**

Основные файлы и папки:

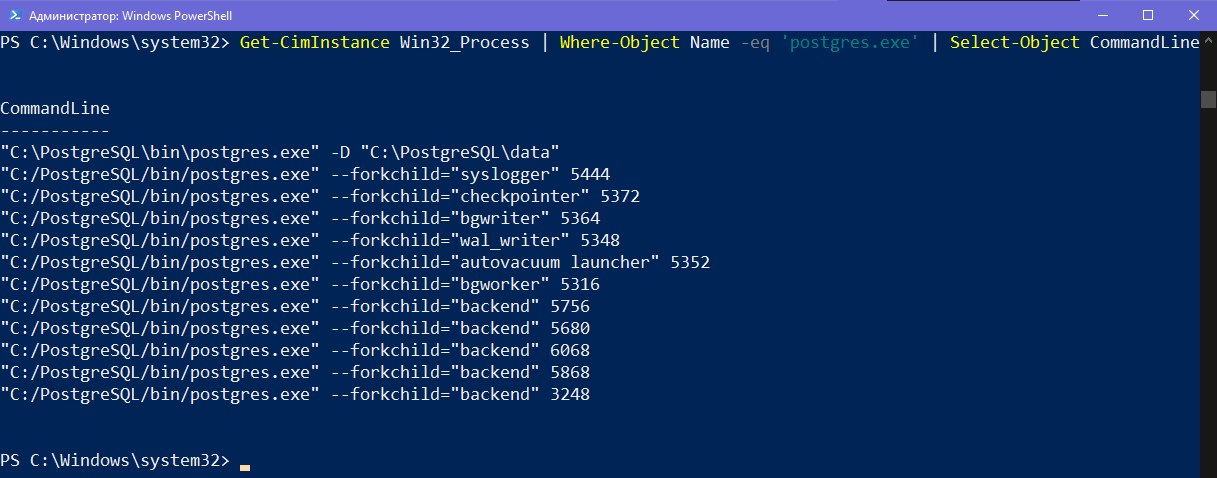
* base/: Содержит данные всех баз данных в кластере. Каждая база данных представлена подкаталогом с уникальным идентификатором.
* global/: Хранит глобальные метаданные кластера, такие как роли, таблицы настройки и системные каталоги.
* pg\_wal/: Каталог для WAL-журналов (Write-Ahead Logging), которые используются для восстановления базы данных в случае сбоя.
* pg\_hba.conf: Файл настройки правил доступа к базам данных (Host-Based Authentication).
* postgresql.conf: Основной файл конфигурации PostgreSQL. Здесь задаются параметры, такие как размер буферов, количество соединений, настройки логирования и многое другое.
* pg\_ident.conf: Используется для сопоставления системных пользователей и ролей PostgreSQL.
* log/: Папка, в которой хранятся журналы работы PostgreSQL (если логирование настроено).
* PG\_VERSION: Текстовый файл, содержащий номер версии PostgreSQL, установленной в этом кластере.
* postmaster.pid: Указывает, что кластер запущен, и содержит информацию о процессе `postmaster`.



**3 Как запущен экземпляр PostgreSQL?**

Запустив PowerShell с правами администратора, выполнил:

Get-CimInstance Win32\_Process | Where-Object Name -eq 'postgres.exe' | Select-Object CommandLine



CommandLine

-----------

"C:\PostgreSQL\bin\postgres.exe" -D "C:\PostgreSQL\data"

"C:/PostgreSQL/bin/postgres.exe" --forkchild="syslogger" 5444

"C:/PostgreSQL/bin/postgres.exe" --forkchild="checkpointer" 5372

"C:/PostgreSQL/bin/postgres.exe" --forkchild="bgwriter" 5364

"C:/PostgreSQL/bin/postgres.exe" --forkchild="wal\_writer" 5348

"C:/PostgreSQL/bin/postgres.exe" --forkchild="autovacuum launcher" 5352

"C:/PostgreSQL/bin/postgres.exe" --forkchild="bgworker" 5316

"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 5756

"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 5680

"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 6068

"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 5868

"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 3248

Первый выведенный процесс – основной, запущен с указанием директории данных.

Остальные процессы, начинающиеся с --forkchild, — это рабочие процессы PostgreSQL. Вот их назначения:

* syslogger: Запись журналов.
* checkpointer: Обеспечение сохранности данных через контрольные точки.
* bgwriter: Запись данных на диск в фоновом режиме.
* wal\_writer: Работа с WAL (Write-Ahead Logging).
* autovacuum launcher: Удаление "мусорных" данных и предотвращение переполнения транзакций.
* bgworker: Фоновые задачи.
* backend: Процессы клиентов, подключающихся к серверу (каждый клиент имеет свой backend-процесс).

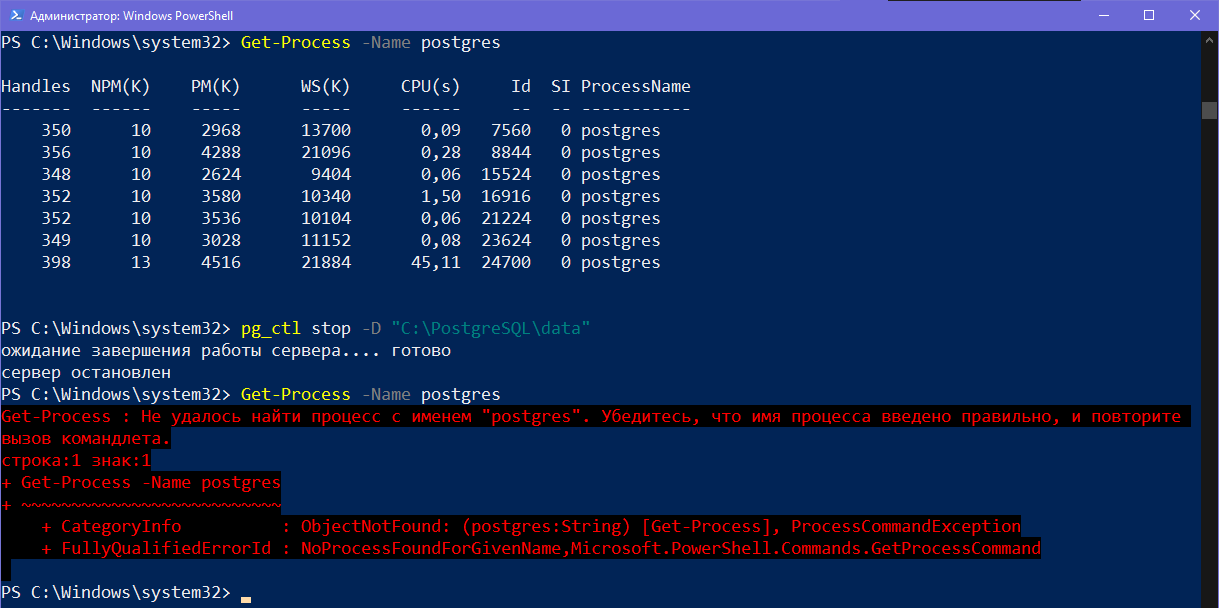
**4 Штатное завершение работы**

С помощью pg\_ctl – утилиты для управления сервером PostgreSql завершим его работу:

pg\_ctl stop -D "C:\PostgreSQL\data"

Эта команда:

* Завершает работу сервера PostgreSQL.
* Гарантирует, что все данные будут записаны в файловую систему.
* Закрывает все клиентские соединения.

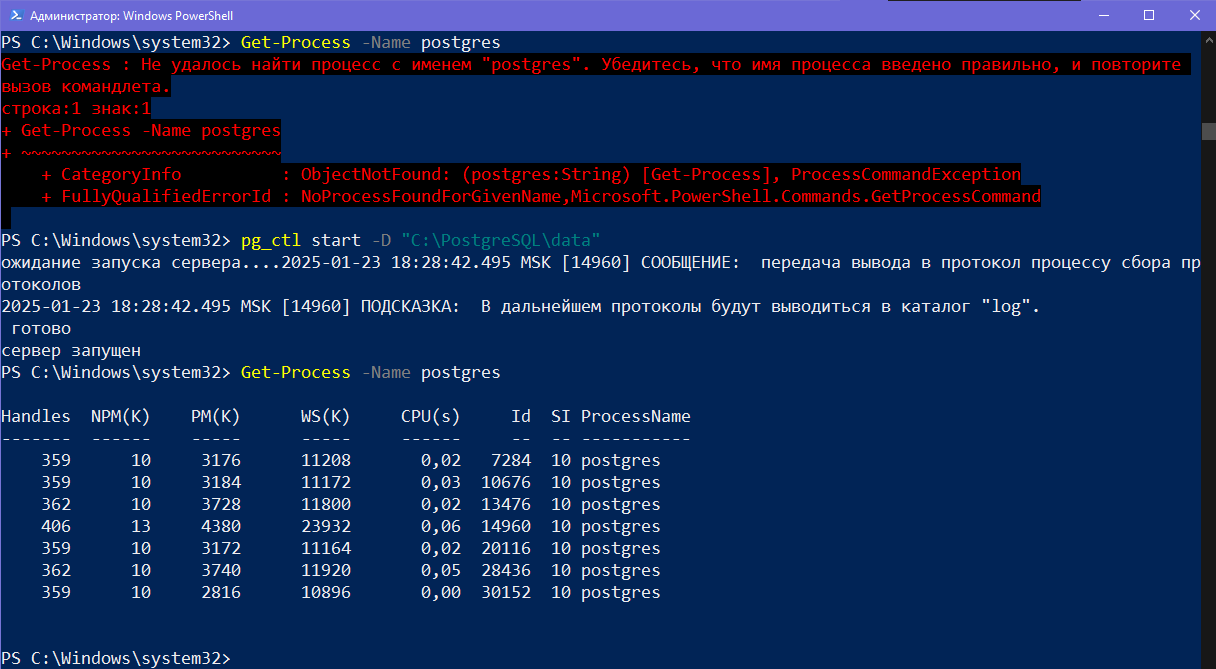


Как видно из скриншота, больше нет активных процессов СУБД.

**5 Запуск вручную**

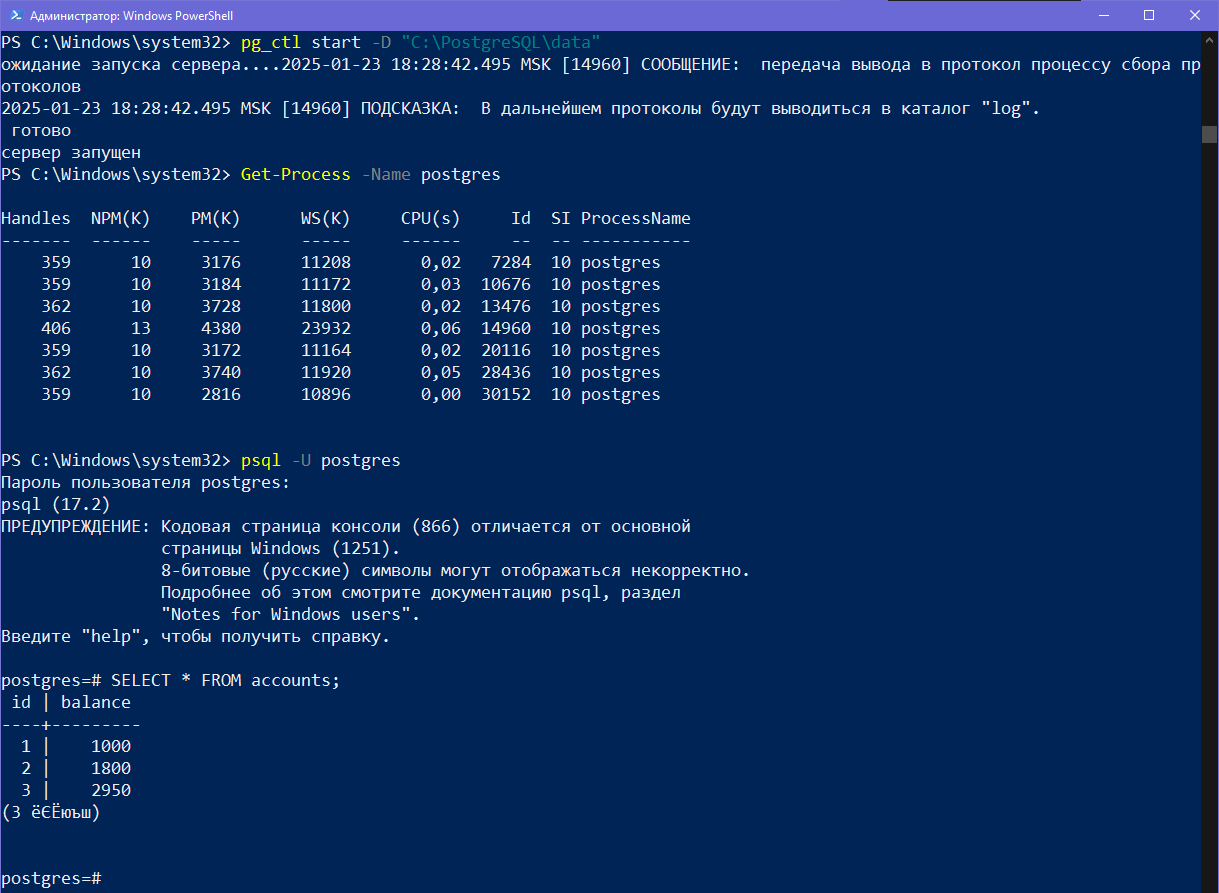
С помощью все той же утилиты запустим процесс, указав нужную директорию:

pg\_ctl start -D "C:\PostgreSQL\data"



**6 Проверка**

Подключимся к БД и выполним запрос.



**7 Создание новой БД**

Создать новую БД (при необходимости можно явно указать владельца):

CREATE DATABASE test\_db;

Посмотрим владельца:

SELECT d.datname, r.rolname AS owner

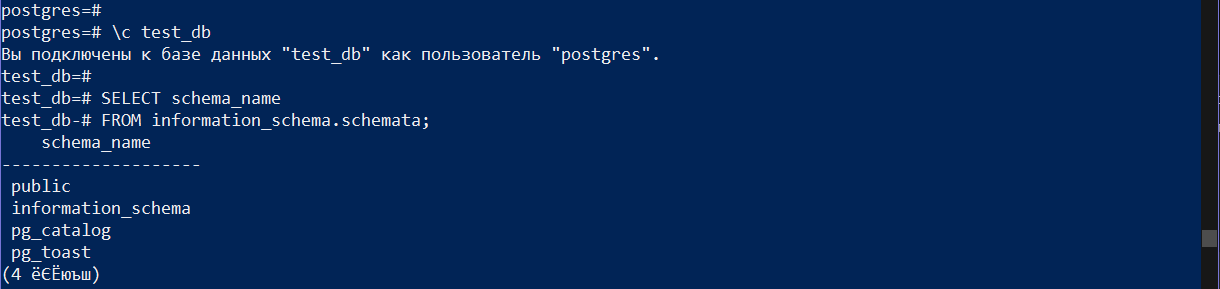
FROM pg\_database d

JOIN pg\_roles r ON d.datdba = r.oid

WHERE d.datname = 'test\_db';



Подключимся к новой БД и проверим, какие в ней есть схемы.



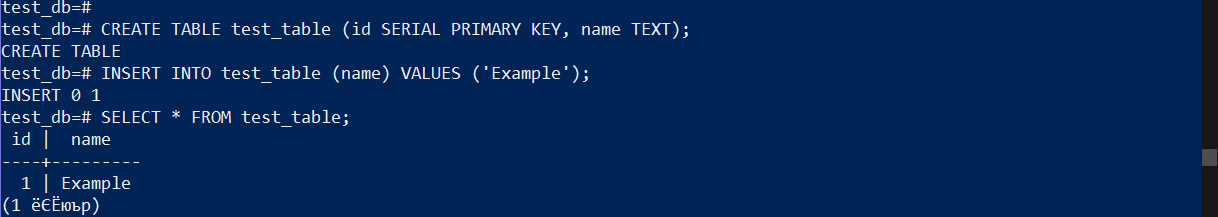
В новой БД есть 4 схемы, 3 из которых – для системных объектов. Посмотрим, какие в public схеме есть таблицы.



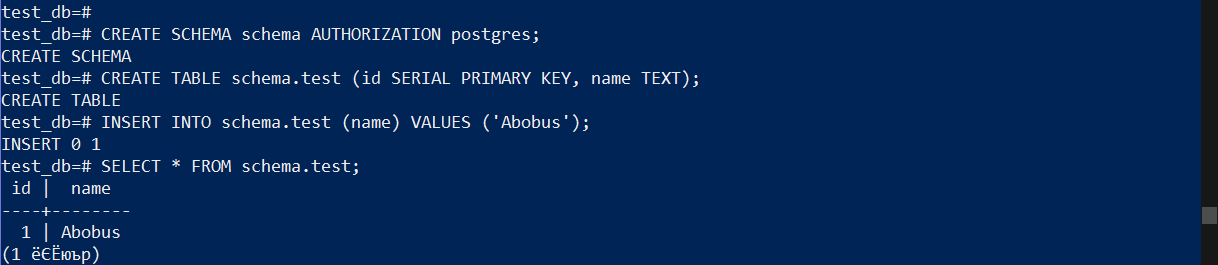
Ожидаемо, так как БД новая, в паблик-схеме пока нет никаких таблиц.

**8 Создание пробных объектов**

Создадим таблицу, и добавим в нее одну запись.



Создадим схему, в ней – таблицу, добавим в таблицу запись.



**9 Без доступа к старой**

Попробуем получить доступ к таблице account, созданной в одной из предыдущих работ.

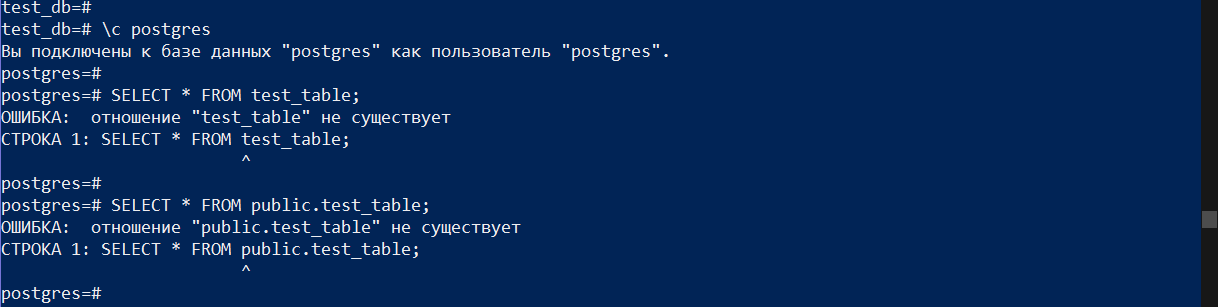


Безуспешно: в данной БД такой таблицы не существует.

**10 Без доступа из старой**

Подключимся к postgres и попробуем получить доступ к одной из новых таблиц.

Безуспешно: мы не можем обращаться к объекту в базе данных, не будучи подключенными к ней.



**Заключение**

В ходе работы успешно выполнены задачи по управлению кластером баз данных PostgreSQL и проверке его работоспособности.

Выяснено местоположение кластера баз данных (`C:/PostgreSQL/data`) и проанализировано его содержимое, включая конфигурационные файлы и служебные данные.

Определена командная строка запуска PostgreSQL и выполнено штатное завершение работы сервера. Сервер успешно перезапущен вручную и проверен на работоспособность.

Создана новая база данных, проверено её содержание и подтверждена изоляция между базами данных. Пробные объекты, созданные в новой базе, оказались недоступны из других баз, что соответствует принципам PostgreSQL.

Данная работа позволила закрепить навыки по настройке, управлению и проверке изоляции данных в PostgreSQL. Все задачи выполнены успешно.

Приложение

* [отчет (docx)](https://github.com/KATEHOK/DBS-labs-7/blob/main/3/report.docx);
* [отчет (pdf)](https://github.com/KATEHOK/DBS-labs-7/blob/main/3/report.pdf).