Федеральное государственное бюджетное образовательное учреждение высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации».

## Кафедра «Прикладная Информатика»

ОТЧЕТ

О ПРОДЕЛАННОЙ ЛАБОРАТОРНОЙ РАБОТЕ №1

по курсу «Информационная безопасность»

Выполнил: студент группы

ИК-731

**(наименование группы)**

***Соколов Дмитрий Александрович***

**(Ф.И.О.)**

## Нижний Новгород

**2025 г.**

ЛАБОРАТОРНАЯ РАБОТА № 1

«Реализация дискреционной модели политики безопасности»

Цель работы: ознакомиться с проблемами реализации политик безопасности в компьютерных системах на примере дискреционной модели.

Вариант № 18

**1 Данные**

В соответствии с вариантом необходимо создать дискреционную модель на 4 пользователя и 4 объекта компьютерной системы.

**2 Реализация программного модуля**

Программный модуль, создающий матрицу доступа пользователей к объектам компьютерной системы, я разработал на языке программирования Python с использованием библиотек random и sys.

**2.1 Идентификаторы пользователей**

Каждый пользователь имеет уникальный идентификатор, например User1. Один из них назначается администратором и получает полные права.

Программа подразумевает 4 пользователя и 4 объекта доступа.

Создание идентификаторов пользователей и администратора:

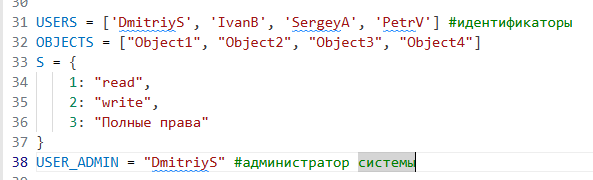


Рисунок 1 – Идентификаторы и параметры матрицы

**2.2 Заполнение матрицы доступа (программное)**

Матрица доступа представлена в виде словаря: matrix[user][object] = set(прав).

Права из множества S = {"read", "write"}

admin реализуется логикой передачи прав (передавать можно read или write). В программе реализована проверка, если есть доступ к read и write – это полный доступ.

Код:

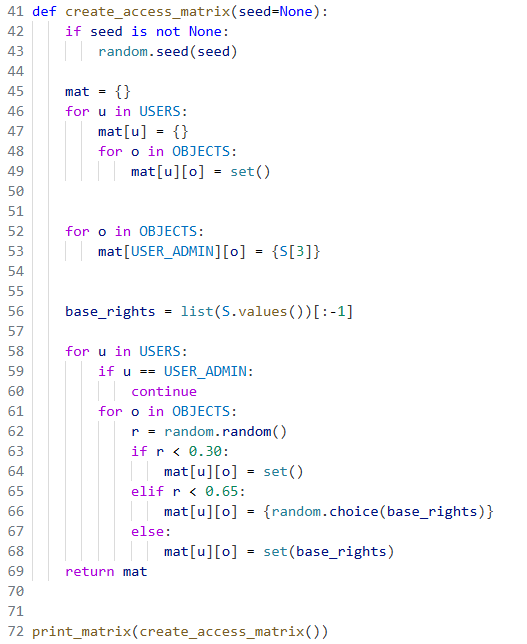


Рисунок 2 – Функция генерации матрицы

На данном этапе при запуске программы формируется следующая матрица (права, кроме администратора, выдаются рандомно). Сделано для проверки, не является основным функционалом.

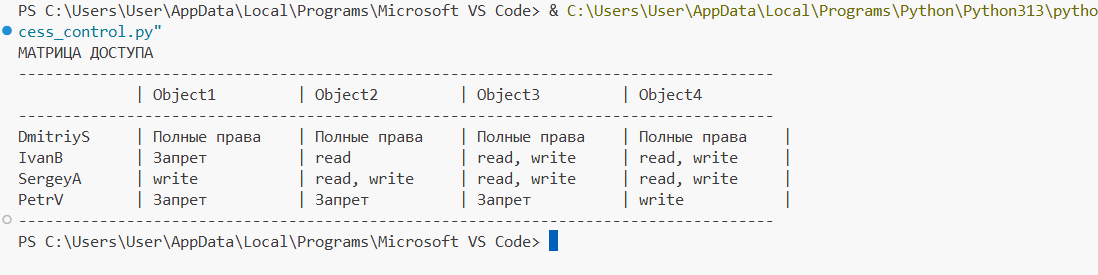


Рисунок 3 – Матрица (Запуск 1)

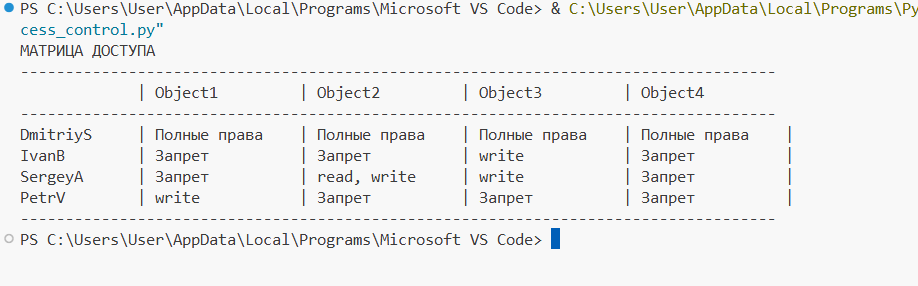


Рисунок 4 – Матрица (Запуск 2)

**3 Идентификация пользователя**

**3.1 Программный модуль**

Для вывода прав была создана функция show\_user\_rights.

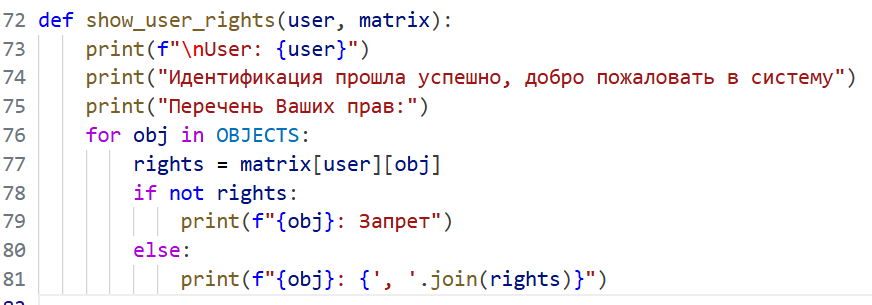


Рисунок 5 – Вывод прав пользователя

В данной функции нет проверки пользователя, тк она реализована в функции main().

**3.2 Вывод прав доступа**

Часть идентификации:

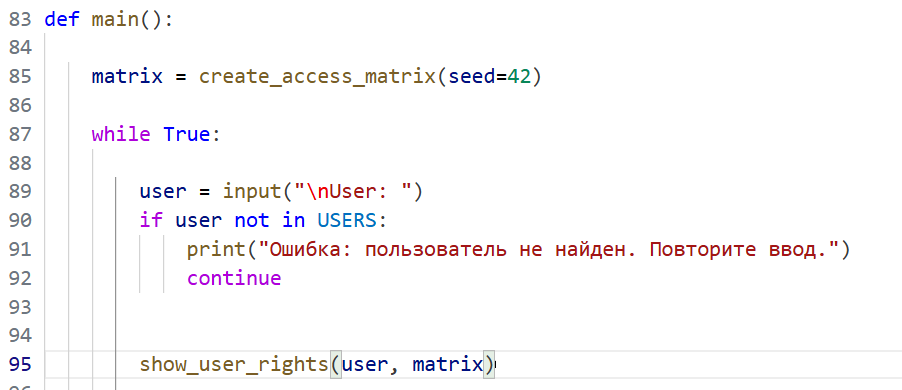


Рисунок 6 – Идентификация

Если данные ошибочны:

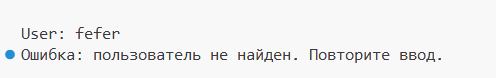


Рисунок 7 - Ошибка

Верный пользователь (администратор):

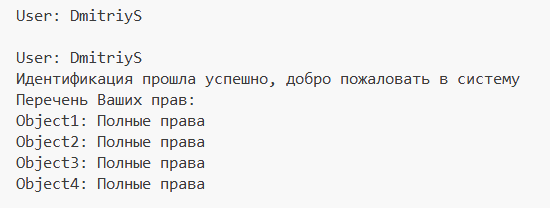


Рисунок 8 - Администратор

Верный пользователь (обычный):

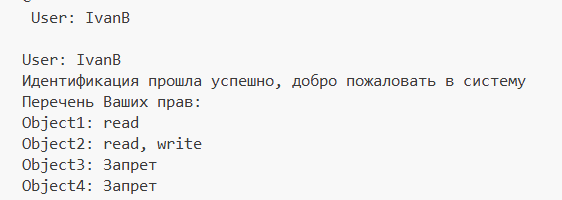


Рисунок 9 – Пользователь

**3.3 Действия пользователей**

Основная логика программы, использующая выше созданные функции.



Рисунок 10 – Основная логика программы

**Алгоритм работы программы**

1. Создание матрицы на основе входных данных, с помощью функции create\_matrix, использует список пользователей, список администраторов системы и список прав пользователей – на выходе матрица распределения прав.
2. Реализация основной логики:

* Программа ожидает ввода имени пользователя;
* Если пользователь не найден – повторный ввод;
* Если пользователь найден, то вывод его прав;
* После вывода прав программа ожидает действие;
* Команда quit – выход из текущего пользователя, можно зайти под другим пользователем;
* Команды read и write – действия над одним из объектов;
* После ввода команды нужно ввести номер объекта, при не верном номере объект, выведет соответствующее сообщение;
* Далее идет проверка доступа к объекту по пользователю и правам, если пользователь не соответствует требованиям – выводится ошибка доступа, иначе «Операция прошла успешно»

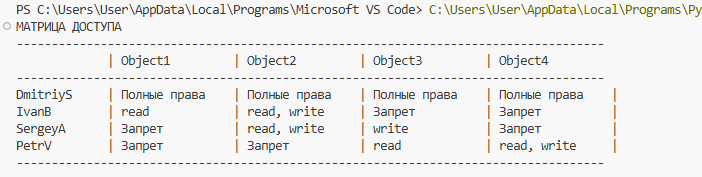
1. Дополнительная команда grant – для смены прав:

* Доступна только администратору, при вводе обычным пользователем – не пройдет проверку;
* После команды grand будет запрошен номер объекта для передачи прав;
* Затем какому пользователю;

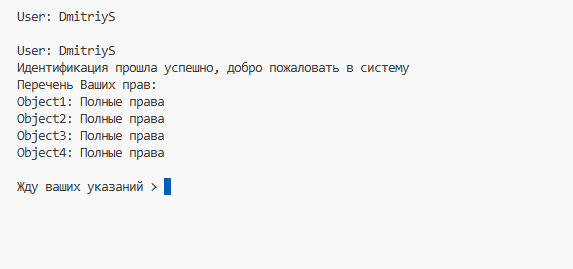
1. При неизвестных командах будет выводиться: “Неизвестная команда. Доступные: read, write, grant, quit”

**Результаты работы программы**

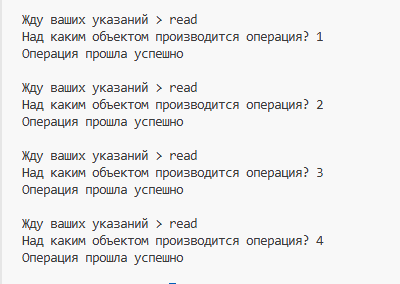
Исходная матрица:



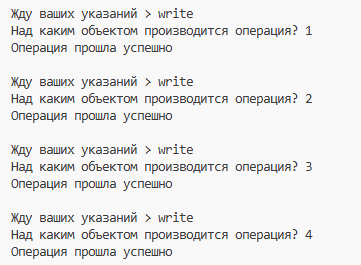
Ввод пользователя (Администратор):



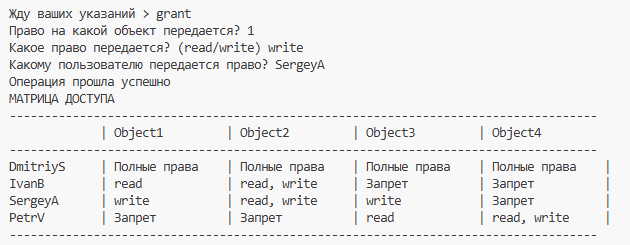
Команда read (Доступна для всех объектов):



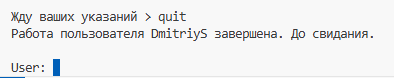
Команда write (Доступна для всех объектов):



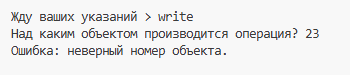
Команда grant, на скриншоте показано, что для объекта 1, выданы права write, пользователю SergeyA, также вывелась измененная матрица:



Команда quit:

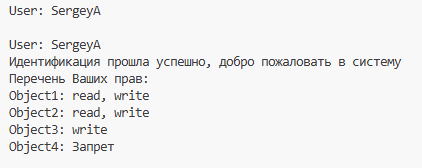


Ошибки:

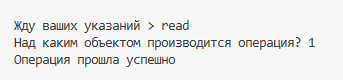




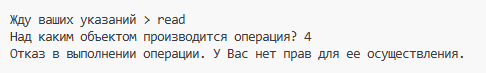
Ввод пользователя (Обычный пользователь):



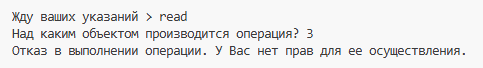
Чтение 1 объекта:



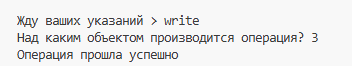
Попытка чтения 4 объекта:



Попытка чтения 3 объекта:



Попытка записи в 3 объект:



**Вывод**

Реализованная мной дискреционная модель безопасности подтвердила свой основной принцип: права доступа определяются владельцами объектов, которые могут делегировать полномочия другим пользователям.

Матрица доступа успешно продемонстрировала управление правами, где каждый пользователь имеет индивидуальный набор правил.

Моя программа показала работу механизмов контроля доступа и передачи прав, что соответствует требованиям коммерческих организаций для разграничения доступа к информации. Модель эффективно предотвращает несанкционированный доступ, сохраняя при этом простоту управления правами для владельцев ресурсов.