Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления
Кафедра интеллектуальных информационных технологий

Отчет
к лабораторной работе No1.1
по дисциплине "Проектирование защищенных
интеллектуальных информационных систем"
на тему:

УПРАВЛЕНИЕ ДОСТУПОМ С ПОМОЩЬЮ СПИСКОВ КОНТРОЛЯ ДОСТУПА

Студент гр. 121702 А. Э. Кривецкий

Руководитель В. В. Захаров

Минск 2024

Ход работы

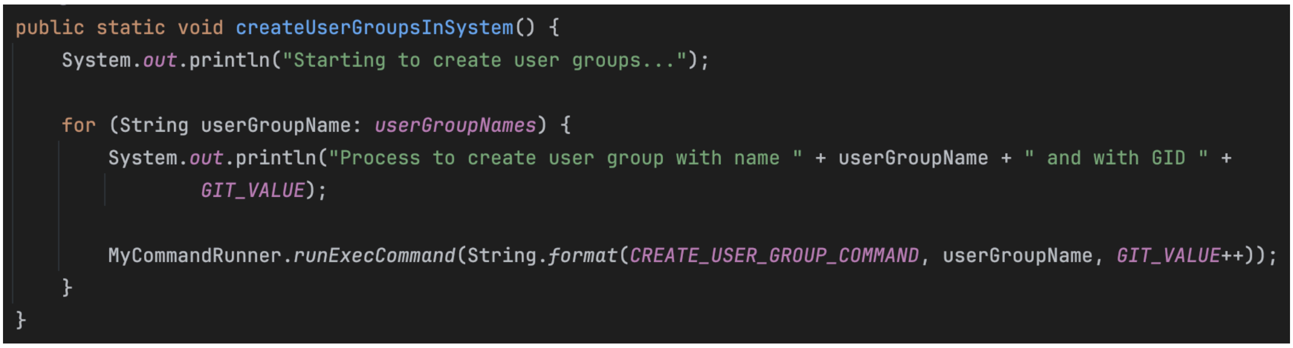
Постановка задачи:

1. Создание программы, которая позволяет настроить ACL для объектов файловой системы и проверить корректность её работы.

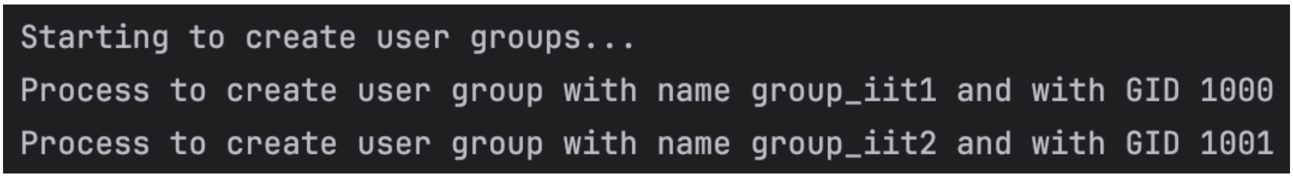
1. Оформление отчёта о проделанной работе.

Выполнение задания и скриншоты работы:

1. Создать группы пользователей group_iit1, group_iit2.

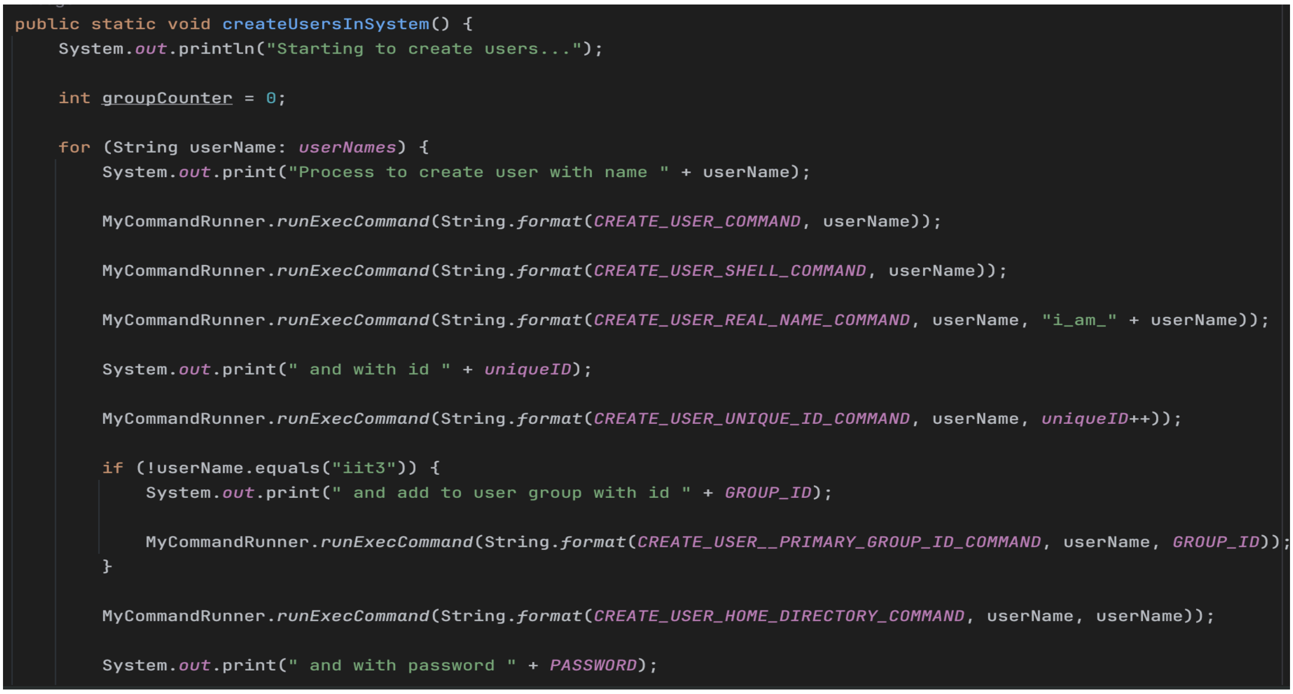


```java
public static void createUserGroupsInSystem() {
    System.out.println("Starting to create user groups...");

    for (String userGroupName: userGroupNames) {
        System.out.println("Process to create user group with name " + userGroupName + " and with GID " +
                GIT_VALUE);

        MyCommandRunner.runExecCommand(String.format(CREATE_USER_GROUP_COMMAND, userGroupName, GIT_VALUE++));
    }
}
```

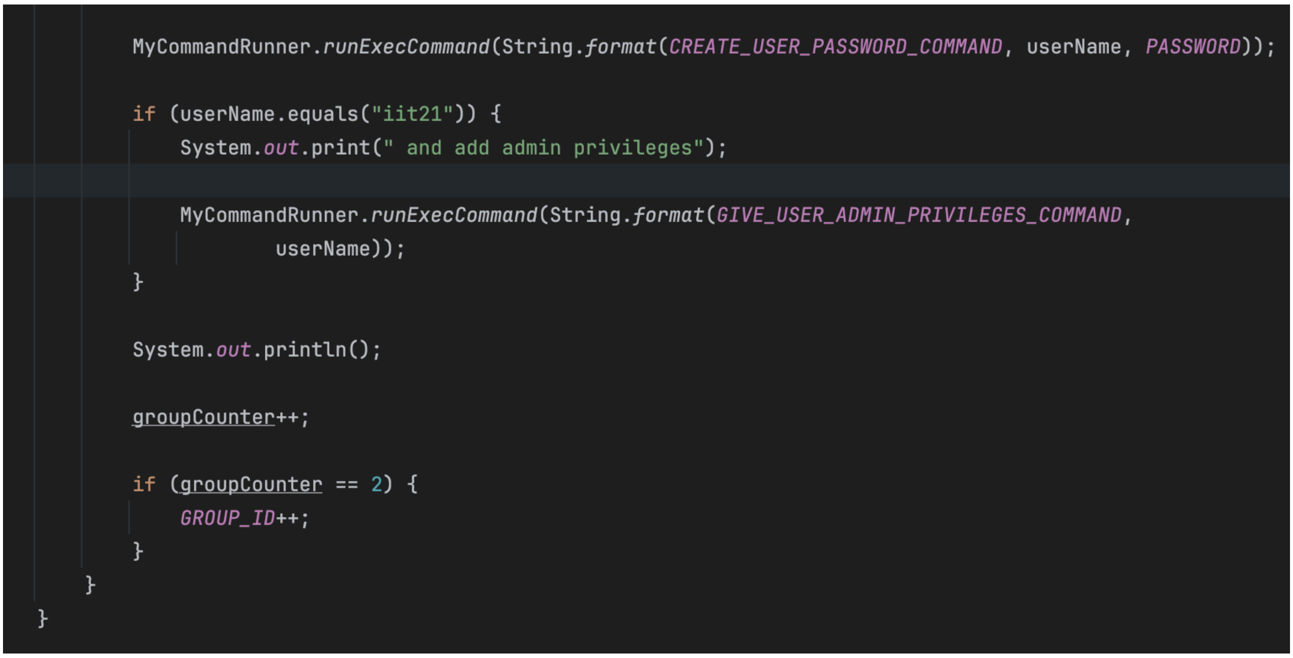


```
Starting to create user groups...
Process to create user group with name group_iit1 and with GID 1000
Process to create user group with name group_iit2 and with GID 1001
```

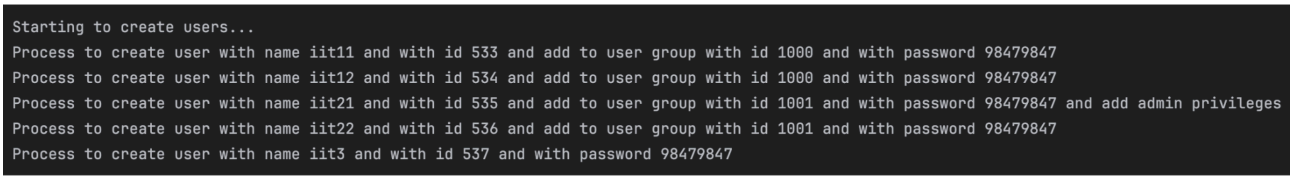
2-5. Создать пользователей и добавить их в группы.



```java
public static void createUsersInSystem() {
    System.out.println("Starting to create users...");

    int groupCounter = 0;

    for (String userName: userNames) {
        System.out.print("Process to create user with name " + userName);

        MyCommandRunner.runExecCommand(String.format(CREATE_USER_COMMAND, userName));

        MyCommandRunner.runExecCommand(String.format(CREATE_USER_SHELL_COMMAND, userName));

        MyCommandRunner.runExecCommand(String.format(CREATE_USER_REAL_NAME_COMMAND, userName, "i_am_" + userName));

        System.out.print(" and with id " + uniqueID);

        MyCommandRunner.runExecCommand(String.format(CREATE_USER_UNIQUE_ID_COMMAND, userName, uniqueID++));

        if (!userName.equals("iit3")) {
            System.out.print(" and add to user group with id " + GROUP_ID);

            MyCommandRunner.runExecCommand(String.format(CREATE_USER__PRIMARY_GROUP_ID_COMMAND, userName, GROUP_ID));
        }

        MyCommandRunner.runExecCommand(String.format(CREATE_USER_HOME_DIRECTORY_COMMAND, userName, userName));

        System.out.print(" and with password " + PASSWORD);
```

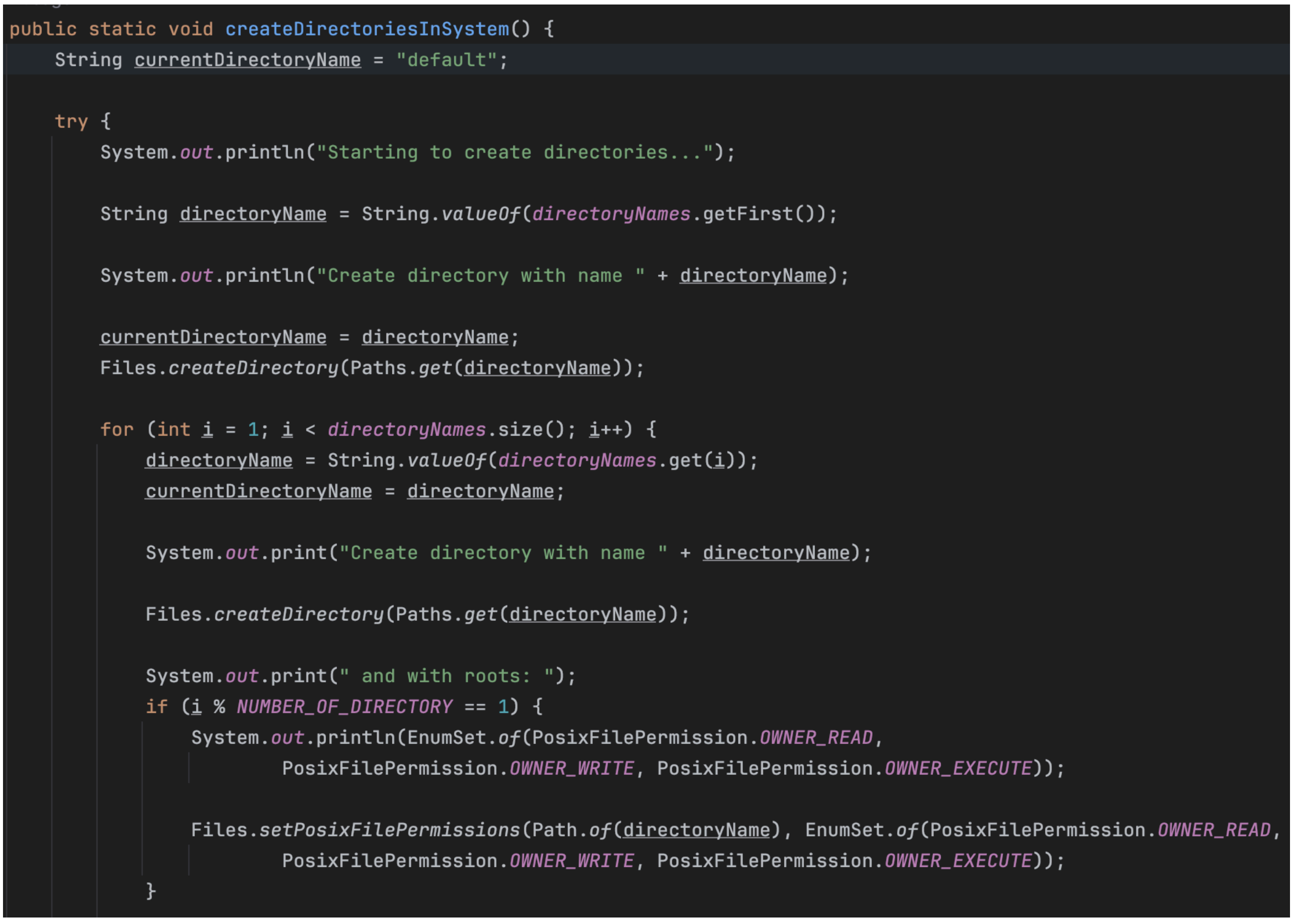


```
        MyCommandRunner.runExecCommand(String.format(CREATE_USER_PASSWORD_COMMAND, userName, PASSWORD));

        if (userName.equals("iit21")) {
            System.out.print(" and add admin privileges");

            MyCommandRunner.runExecCommand(String.format(GIVE_USER_ADMIN_PRIVILEGES_COMMAND,
                    userName));
        }

        System.out.println();

        groupCounter++;

        if (groupCounter == 2) {
            GROUP_ID++;
        }
    }
}
```

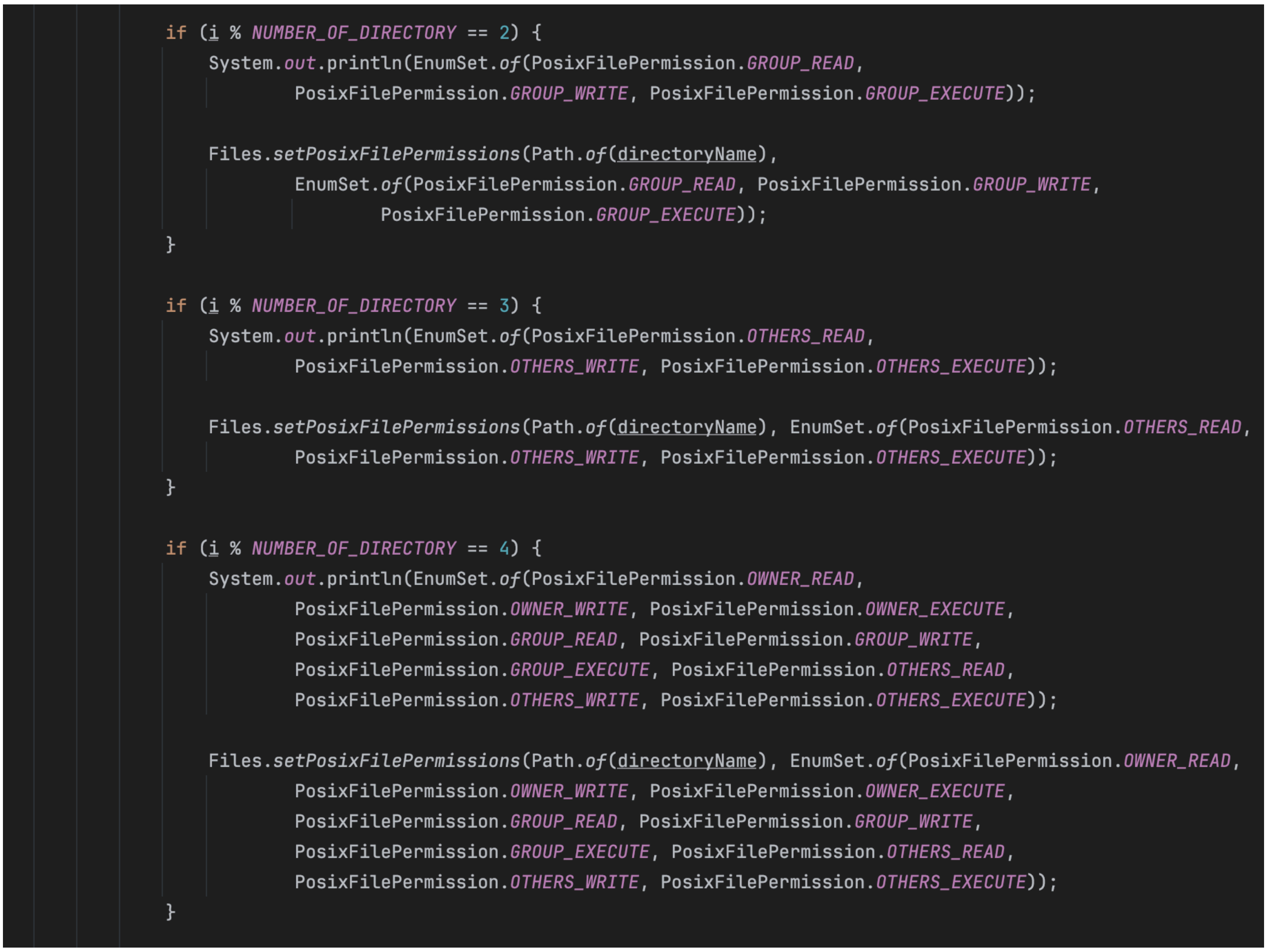


```
Starting to create users...
Process to create user with name iit11 and with id 533 and add to user group with id 1000 and with password 98479847
Process to create user with name iit12 and with id 534 and add to user group with id 1000 and with password 98479847
Process to create user with name iit21 and with id 535 and add to user group with id 1001 and with password 98479847 and add admin privileges
Process to create user with name iit22 and with id 536 and add to user group with id 1001 and with password 98479847
Process to create user with name iit3 and with id 537 and with password 98479847
```

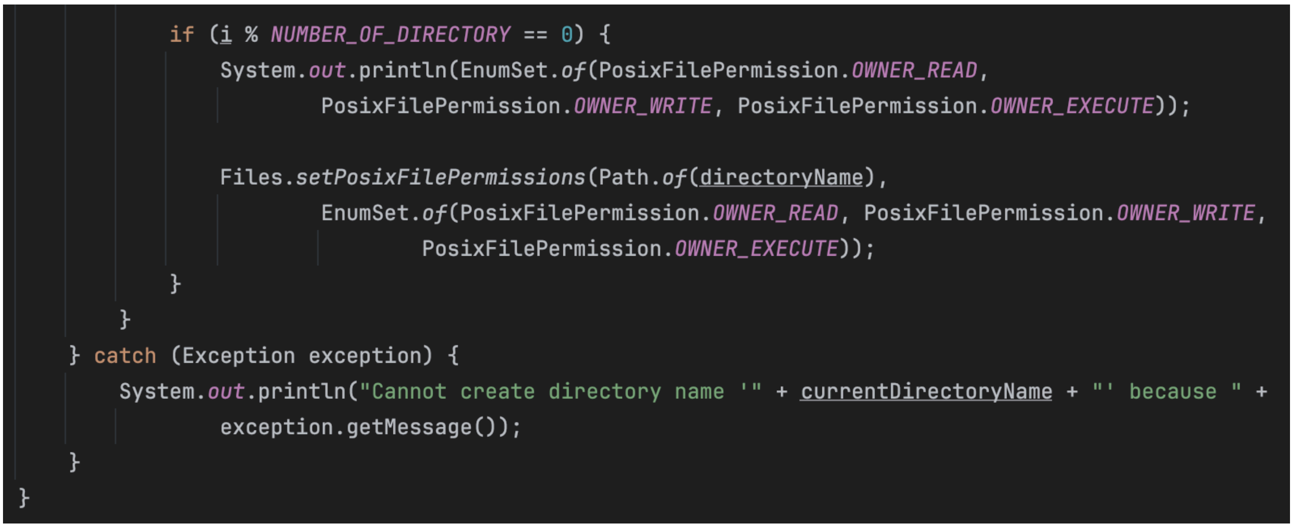
6-11. Создать папки и добавить им права.



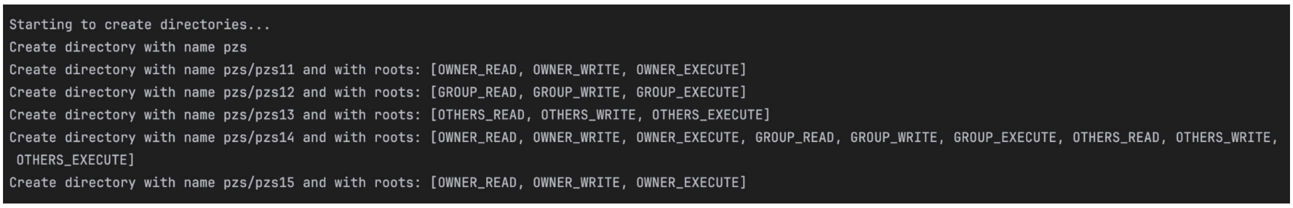
```
public static void createDirectoriesInSystem() {
    String currentDirectoryName = "default";

    try {
        System.out.println("Starting to create directories...");

        String directoryName = String.valueOf(directoryNames.getFirst());

        System.out.println("Create directory with name " + directoryName);

        currentDirectoryName = directoryName;
        Files.createDirectory(Paths.get(directoryName));

        for (int i = 1; i < directoryNames.size(); i++) {
            directoryName = String.valueOf(directoryNames.get(i));
            currentDirectoryName = directoryName;

            System.out.print("Create directory with name " + directoryName);

            Files.createDirectory(Paths.get(directoryName));

            System.out.print(" and with roots: ");
            if (i % NUMBER_OF_DIRECTORY == 1) {
                System.out.println(EnumSet.of(PosixFilePermission.OWNER_READ,
                        PosixFilePermission.OWNER_WRITE, PosixFilePermission.OWNER_EXECUTE));

                Files.setPosixFilePermissions(Path.of(directoryName), EnumSet.of(PosixFilePermission.OWNER_READ,
                        PosixFilePermission.OWNER_WRITE, PosixFilePermission.OWNER_EXECUTE));
            }
```



```java
            if (i % NUMBER_OF_DIRECTORY == 2) {
                System.out.println(EnumSet.of(PosixFilePermission.GROUP_READ,
                        PosixFilePermission.GROUP_WRITE, PosixFilePermission.GROUP_EXECUTE));

                Files.setPosixFilePermissions(Path.of(directoryName),
                        EnumSet.of(PosixFilePermission.GROUP_READ, PosixFilePermission.GROUP_WRITE,
                                PosixFilePermission.GROUP_EXECUTE));
            }

            if (i % NUMBER_OF_DIRECTORY == 3) {
                System.out.println(EnumSet.of(PosixFilePermission.OTHERS_READ,
                        PosixFilePermission.OTHERS_WRITE, PosixFilePermission.OTHERS_EXECUTE));

                Files.setPosixFilePermissions(Path.of(directoryName), EnumSet.of(PosixFilePermission.OTHERS_READ,
                        PosixFilePermission.OTHERS_WRITE, PosixFilePermission.OTHERS_EXECUTE));
            }

            if (i % NUMBER_OF_DIRECTORY == 4) {
                System.out.println(EnumSet.of(PosixFilePermission.OWNER_READ,
                        PosixFilePermission.OWNER_WRITE, PosixFilePermission.OWNER_EXECUTE,
                        PosixFilePermission.GROUP_READ, PosixFilePermission.GROUP_WRITE,
                        PosixFilePermission.GROUP_EXECUTE, PosixFilePermission.OTHERS_READ,
                        PosixFilePermission.OTHERS_WRITE, PosixFilePermission.OTHERS_EXECUTE));

                Files.setPosixFilePermissions(Path.of(directoryName), EnumSet.of(PosixFilePermission.OWNER_READ,
                        PosixFilePermission.OWNER_WRITE, PosixFilePermission.OWNER_EXECUTE,
                        PosixFilePermission.GROUP_READ, PosixFilePermission.GROUP_WRITE,
                        PosixFilePermission.GROUP_EXECUTE, PosixFilePermission.OTHERS_READ,
                        PosixFilePermission.OTHERS_WRITE, PosixFilePermission.OTHERS_EXECUTE));
            }
```

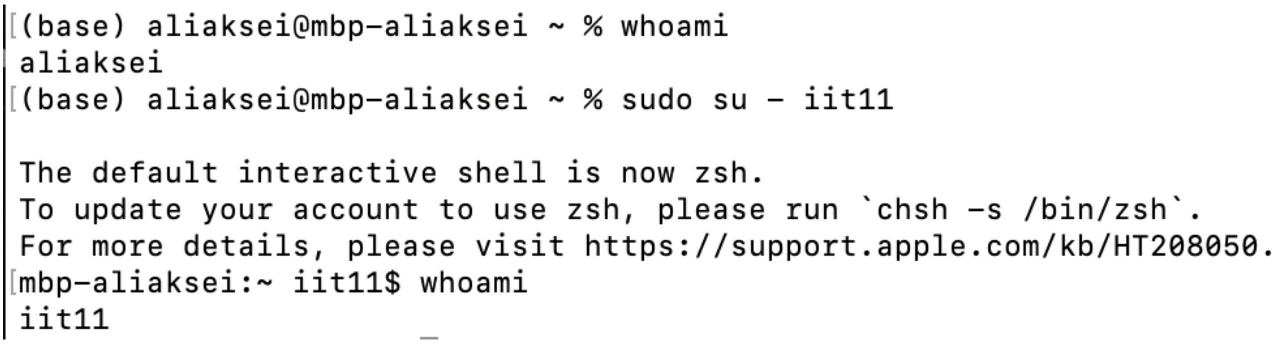


```java
            if (i % NUMBER_OF_DIRECTORY == 0) {
                System.out.println(EnumSet.of(PosixFilePermission.OWNER_READ,
                        PosixFilePermission.OWNER_WRITE, PosixFilePermission.OWNER_EXECUTE));

                Files.setPosixFilePermissions(Path.of(directoryName),
                        EnumSet.of(PosixFilePermission.OWNER_READ, PosixFilePermission.OWNER_WRITE,
                                PosixFilePermission.OWNER_EXECUTE));
            }
        }
    } catch (Exception exception) {
        System.out.println("Cannot create directory name '" + currentDirectoryName + "' because " +
                exception.getMessage());
    }
}
```



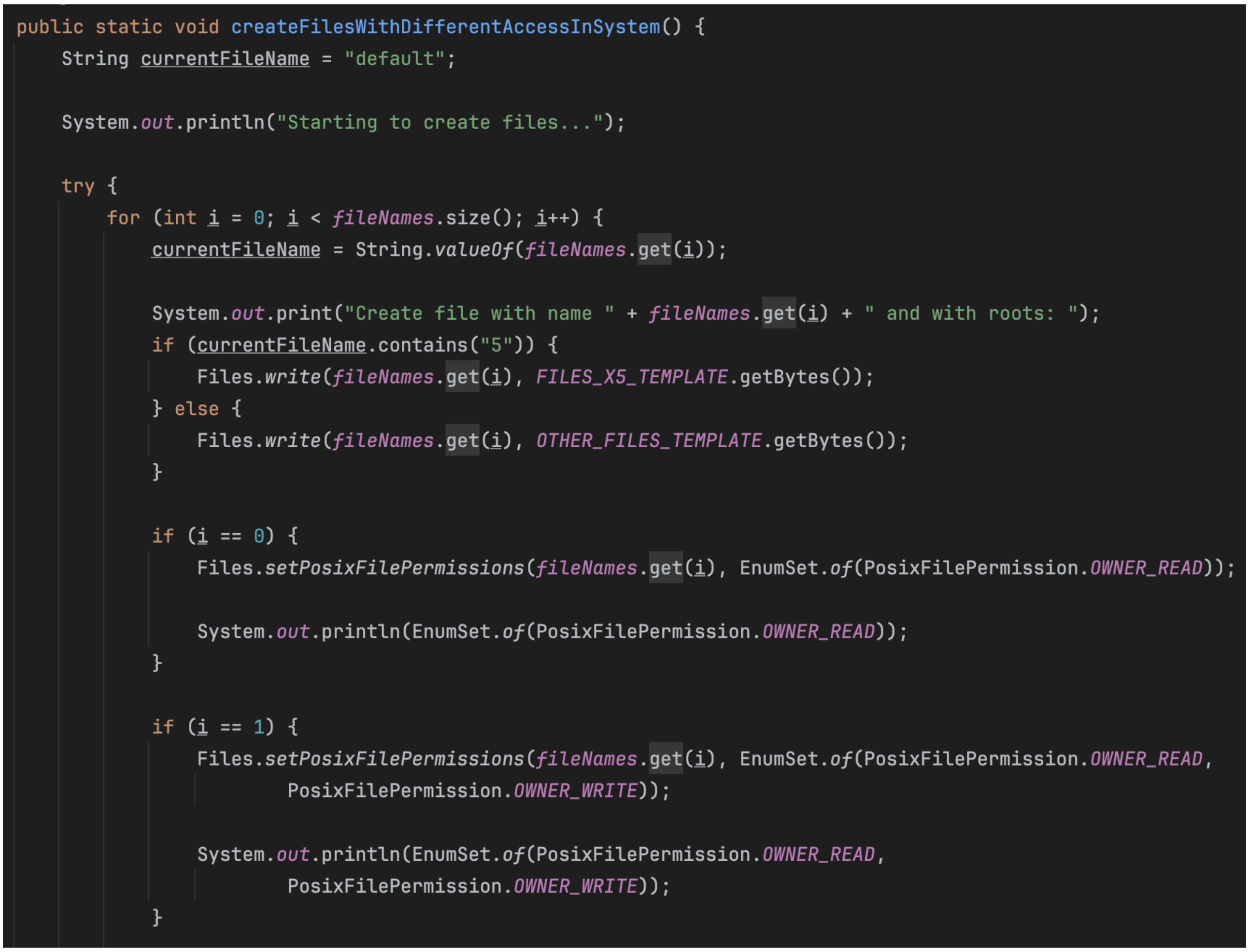
```
Starting to create directories...
Create directory with name pzs
Create directory with name pzs/pzs11 and with roots: [OWNER_READ, OWNER_WRITE, OWNER_EXECUTE]
Create directory with name pzs/pzs12 and with roots: [GROUP_READ, GROUP_WRITE, GROUP_EXECUTE]
Create directory with name pzs/pzs13 and with roots: [OTHERS_READ, OTHERS_WRITE, OTHERS_EXECUTE]
Create directory with name pzs/pzs14 and with roots: [OWNER_READ, OWNER_WRITE, OWNER_EXECUTE, GROUP_READ, GROUP_WRITE, GROUP_EXECUTE, OTHERS_READ, OTHERS_WRITE,
 OTHERS_EXECUTE]
Create directory with name pzs/pzs15 and with roots: [OWNER_READ, OWNER_WRITE, OWNER_EXECUTE]
```

12. Сменить текущего пользователя на iit11.

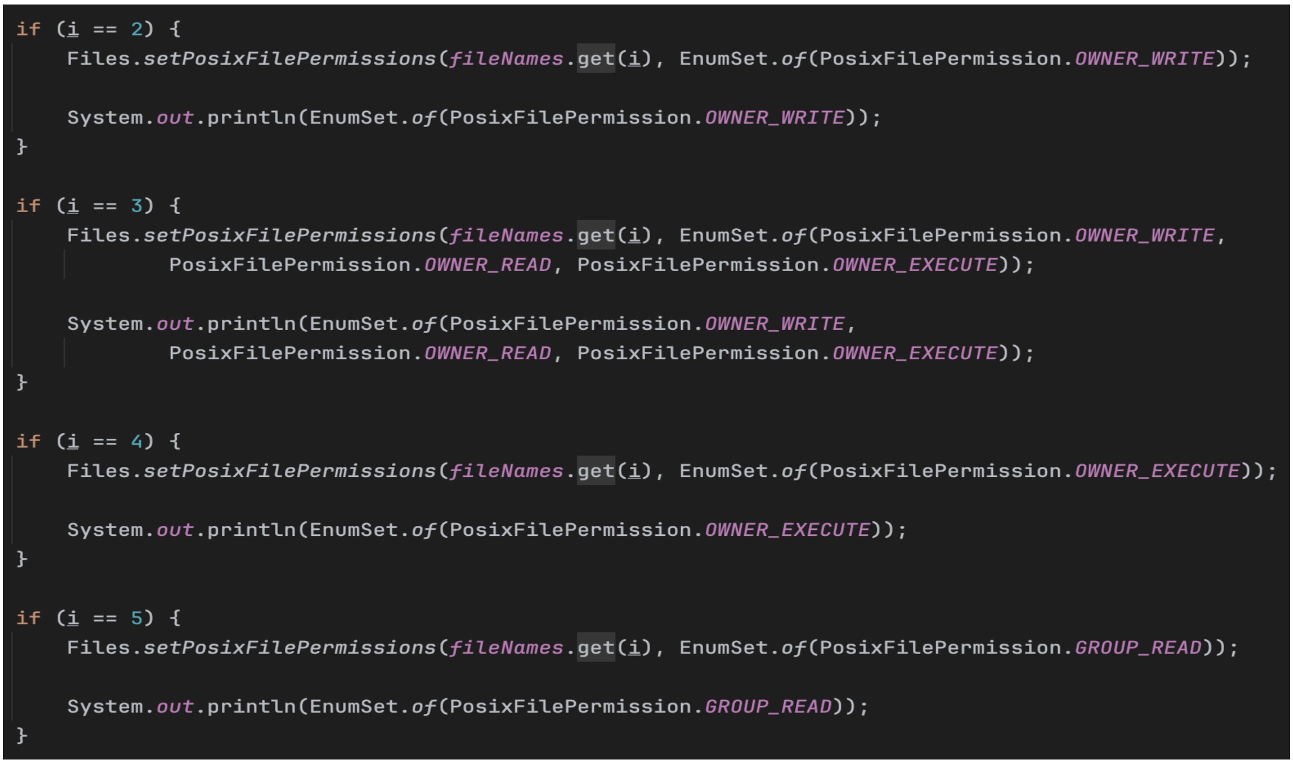


```
(base) aliaksei@mbp-aliaksei ~ % whoami
aliaksei
(base) aliaksei@mbp-aliaksei ~ % sudo su - iit11

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
mbp-aliaksei:~ iit11$ whoami
iit11
```

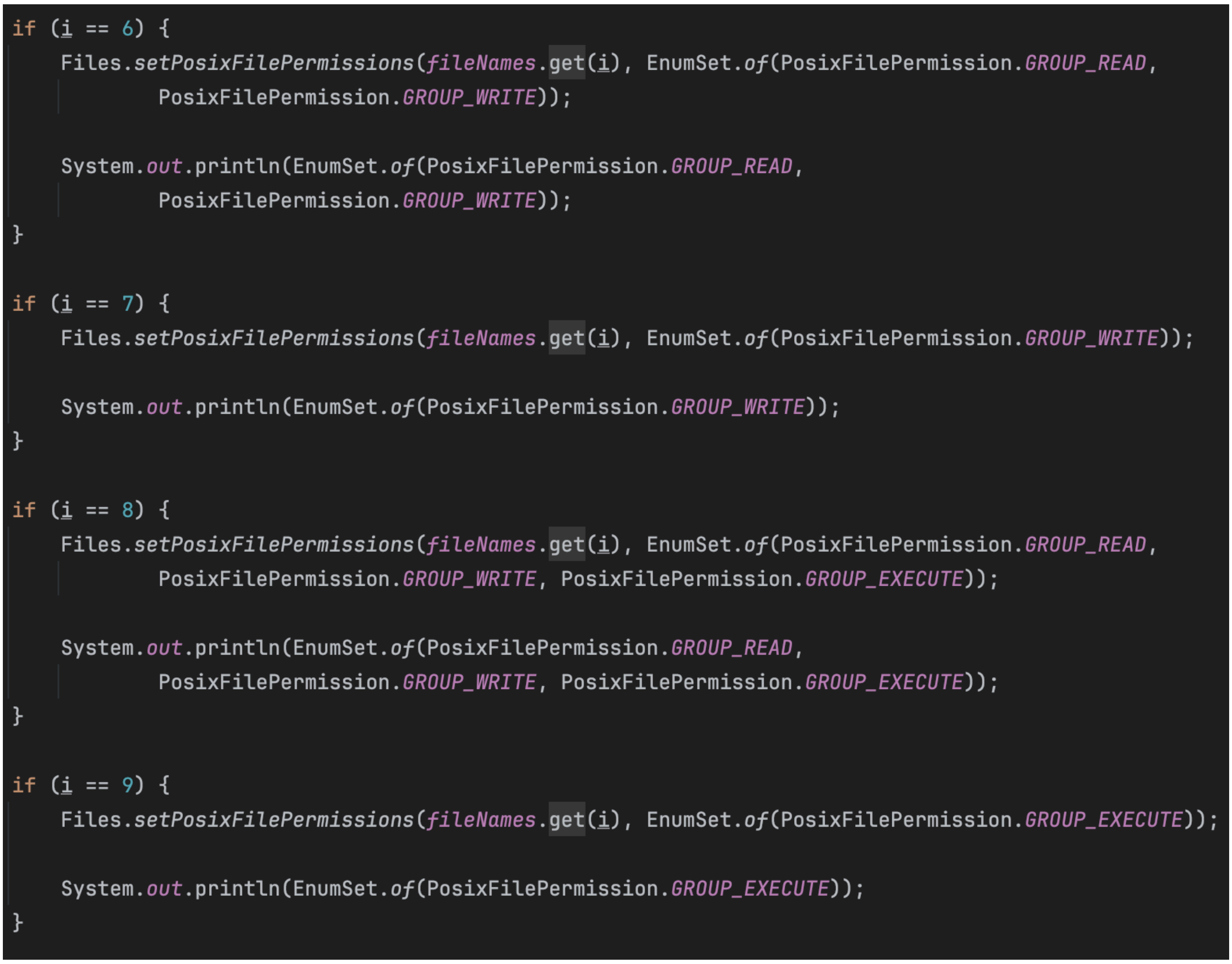
13. Создание файлов с определёнными правами и определёнными данными.



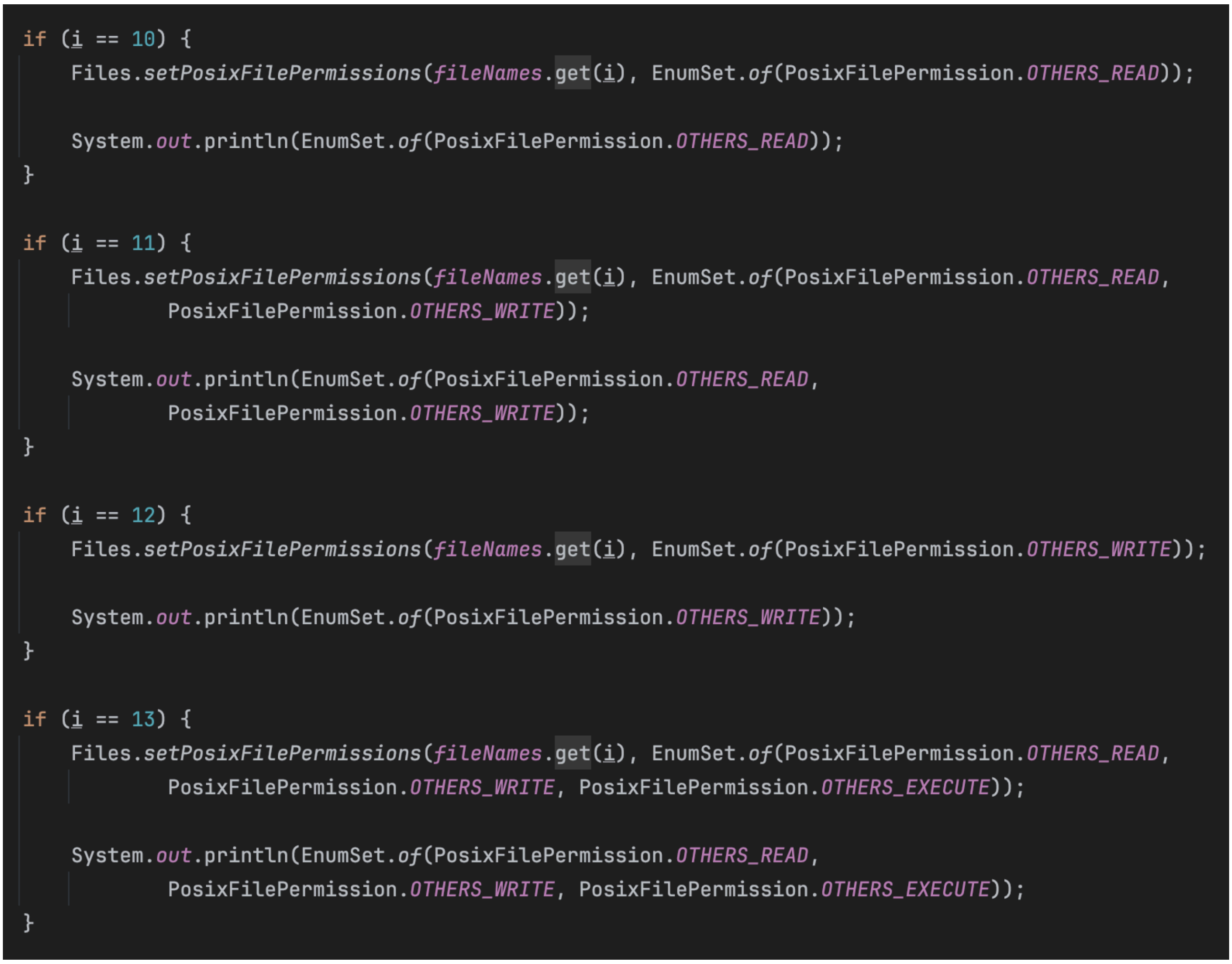
```
public static void createFilesWithDifferentAccessInSystem() {
    String currentFileName = "default";

    System.out.println("Starting to create files...");

    try {
        for (int i = 0; i < fileNames.size(); i++) {
            currentFileName = String.valueOf(fileNames.get(i));

            System.out.print("Create file with name " + fileNames.get(i) + " and with roots: ");
            if (currentFileName.contains("5")) {
                Files.write(fileNames.get(i), FILES_X5_TEMPLATE.getBytes());
            } else {
                Files.write(fileNames.get(i), OTHER_FILES_TEMPLATE.getBytes());
            }

            if (i == 0) {
                Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_READ));

                System.out.println(EnumSet.of(PosixFilePermission.OWNER_READ));
            }

            if (i == 1) {
                Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_READ,
                        PosixFilePermission.OWNER_WRITE));

                System.out.println(EnumSet.of(PosixFilePermission.OWNER_READ,
                        PosixFilePermission.OWNER_WRITE));
            }
```



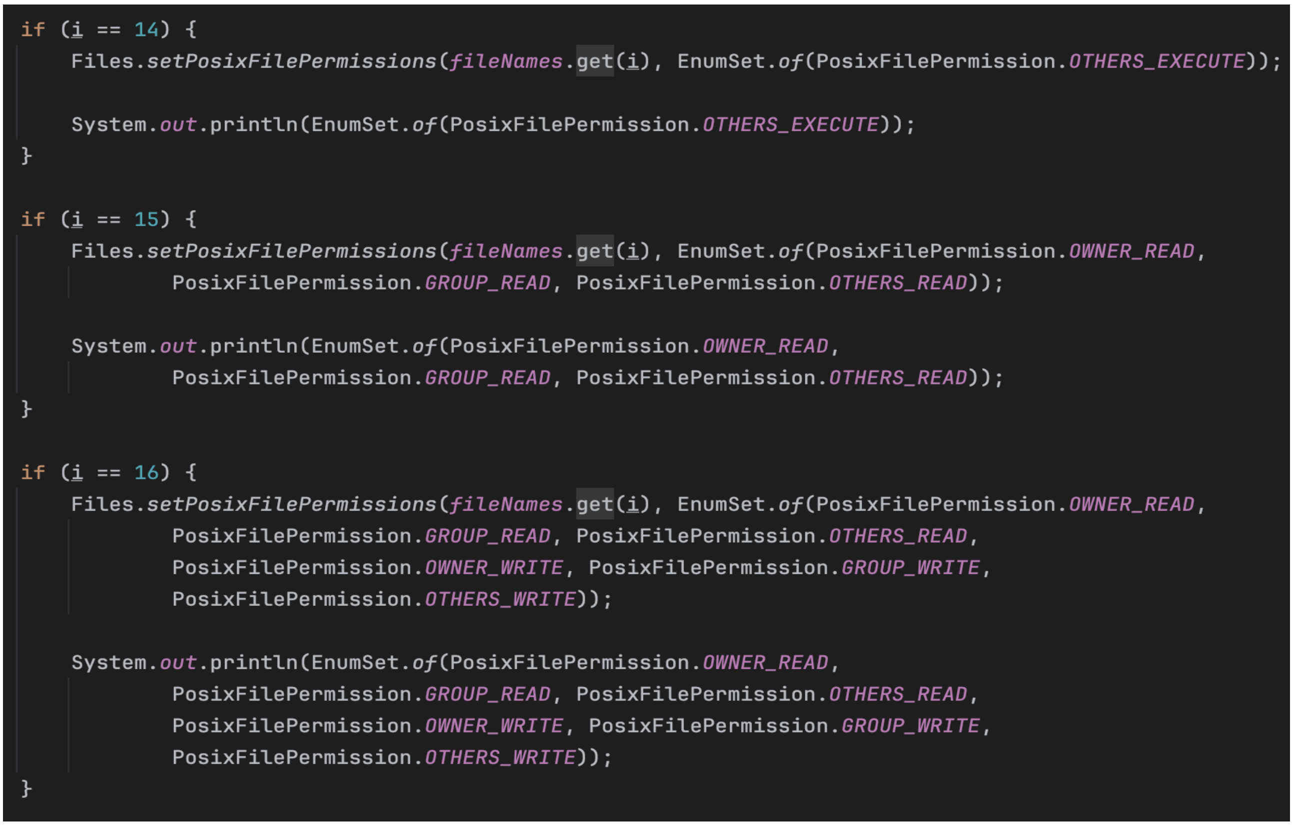
```java
if (i == 2) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_WRITE));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_WRITE));
}

if (i == 3) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_WRITE,
            PosixFilePermission.OWNER_READ, PosixFilePermission.OWNER_EXECUTE));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_WRITE,
            PosixFilePermission.OWNER_READ, PosixFilePermission.OWNER_EXECUTE));
}

if (i == 4) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_EXECUTE));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_EXECUTE));
}

if (i == 5) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.GROUP_READ));

    System.out.println(EnumSet.of(PosixFilePermission.GROUP_READ));
}
```



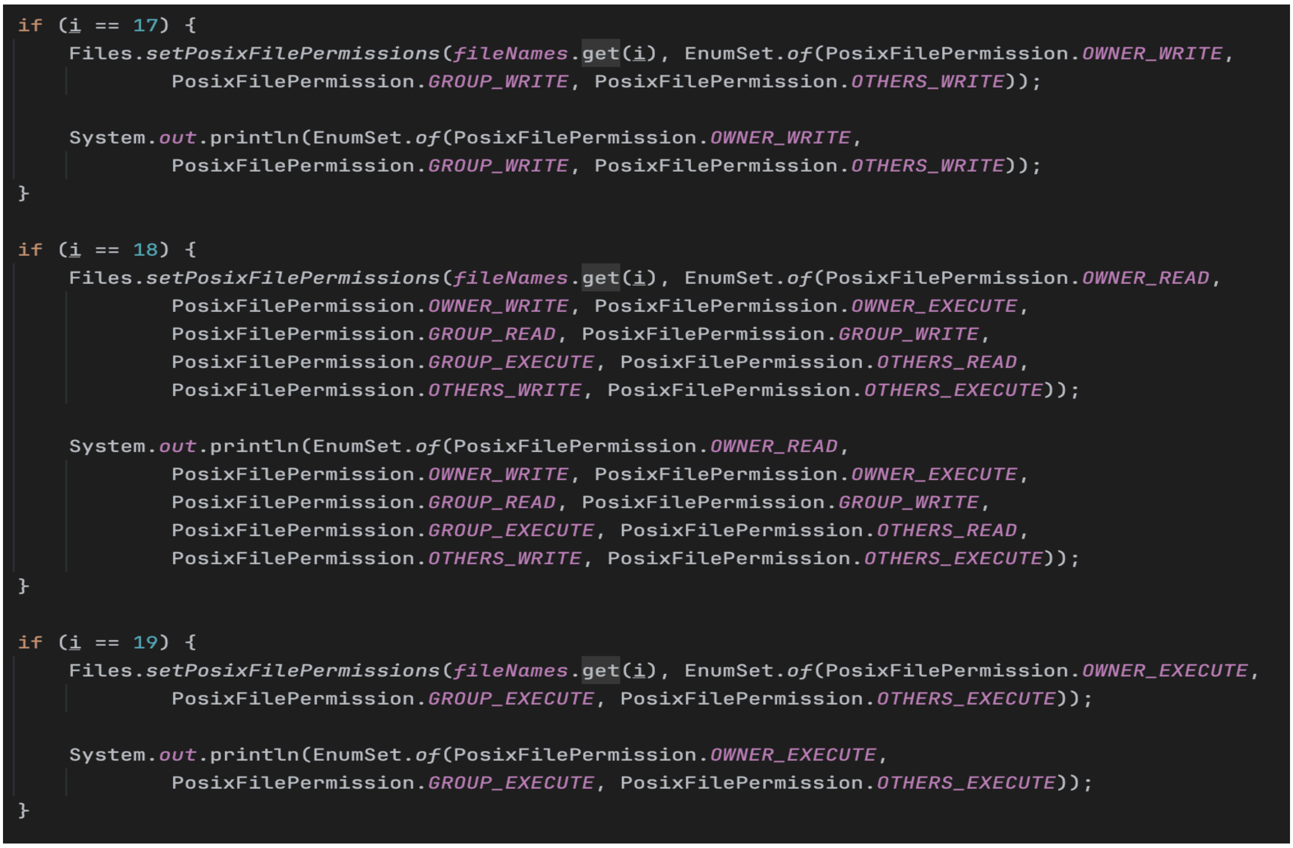
```java
if (i == 6) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.GROUP_READ,
            PosixFilePermission.GROUP_WRITE));

    System.out.println(EnumSet.of(PosixFilePermission.GROUP_READ,
            PosixFilePermission.GROUP_WRITE));
}

if (i == 7) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.GROUP_WRITE));

    System.out.println(EnumSet.of(PosixFilePermission.GROUP_WRITE));
}

if (i == 8) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.GROUP_READ,
            PosixFilePermission.GROUP_WRITE, PosixFilePermission.GROUP_EXECUTE));

    System.out.println(EnumSet.of(PosixFilePermission.GROUP_READ,
            PosixFilePermission.GROUP_WRITE, PosixFilePermission.GROUP_EXECUTE));
}

if (i == 9) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.GROUP_EXECUTE));

    System.out.println(EnumSet.of(PosixFilePermission.GROUP_EXECUTE));
}
```



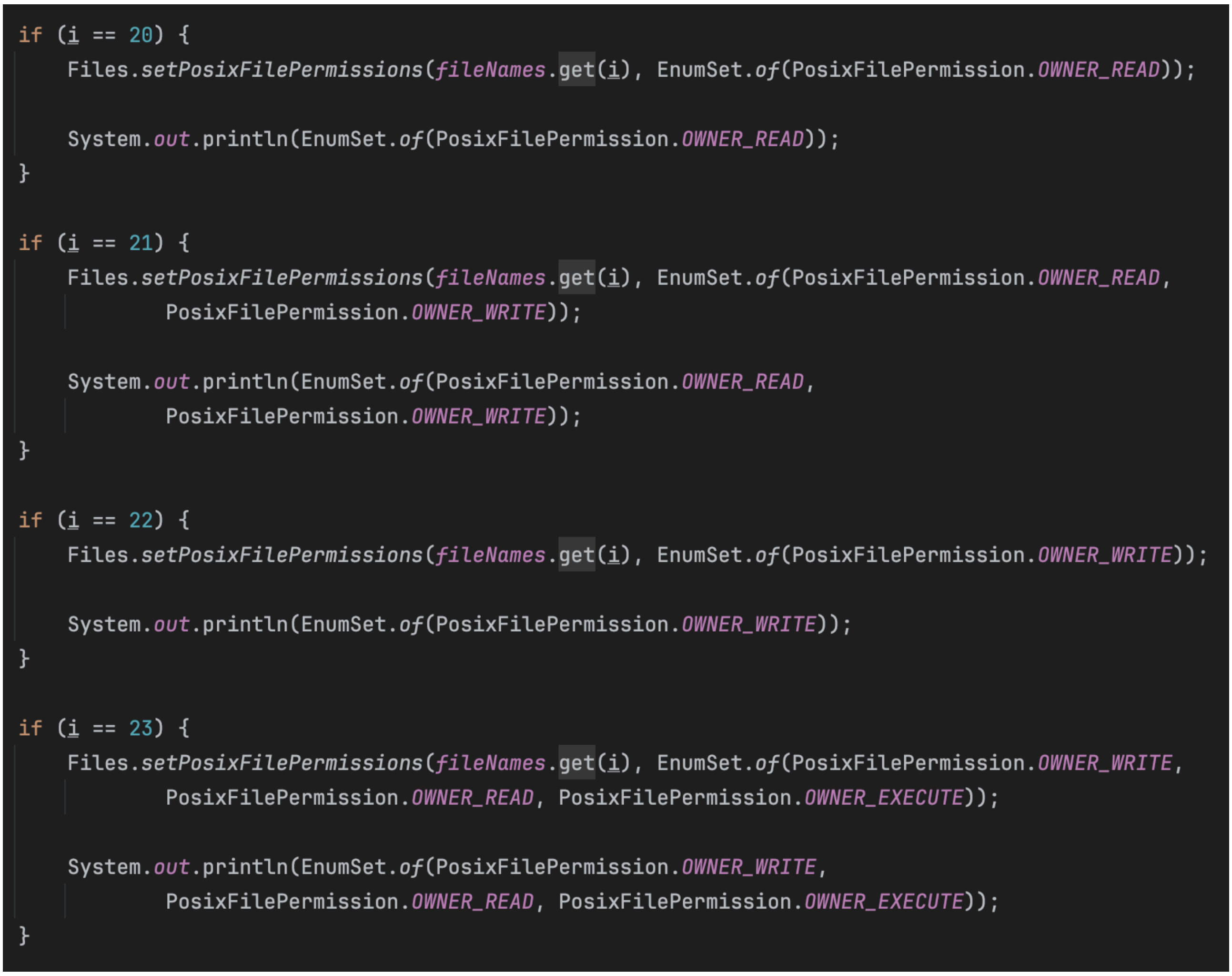
```java
if (i == 10) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OTHERS_READ));

    System.out.println(EnumSet.of(PosixFilePermission.OTHERS_READ));
}

if (i == 11) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OTHERS_READ,
            PosixFilePermission.OTHERS_WRITE));

    System.out.println(EnumSet.of(PosixFilePermission.OTHERS_READ,
            PosixFilePermission.OTHERS_WRITE));
}

if (i == 12) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OTHERS_WRITE));

    System.out.println(EnumSet.of(PosixFilePermission.OTHERS_WRITE));
}

if (i == 13) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OTHERS_READ,
            PosixFilePermission.OTHERS_WRITE, PosixFilePermission.OTHERS_EXECUTE));

    System.out.println(EnumSet.of(PosixFilePermission.OTHERS_READ,
            PosixFilePermission.OTHERS_WRITE, PosixFilePermission.OTHERS_EXECUTE));
}
```



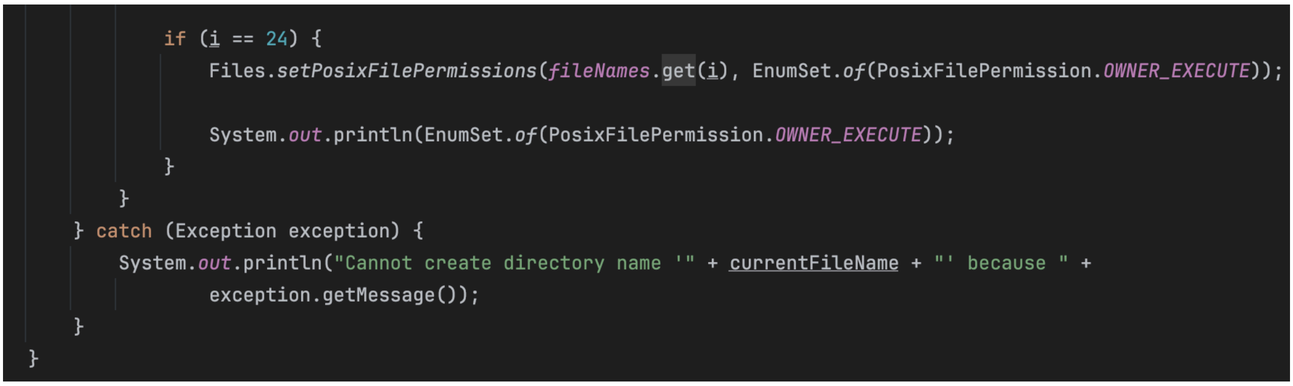
```java
if (i == 14) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OTHERS_EXECUTE));

    System.out.println(EnumSet.of(PosixFilePermission.OTHERS_EXECUTE));
}

if (i == 15) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_READ,
            PosixFilePermission.GROUP_READ, PosixFilePermission.OTHERS_READ));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_READ,
            PosixFilePermission.GROUP_READ, PosixFilePermission.OTHERS_READ));
}

if (i == 16) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_READ,
            PosixFilePermission.GROUP_READ, PosixFilePermission.OTHERS_READ,
            PosixFilePermission.OWNER_WRITE, PosixFilePermission.GROUP_WRITE,
            PosixFilePermission.OTHERS_WRITE));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_READ,
            PosixFilePermission.GROUP_READ, PosixFilePermission.OTHERS_READ,
            PosixFilePermission.OWNER_WRITE, PosixFilePermission.GROUP_WRITE,
            PosixFilePermission.OTHERS_WRITE));
}
```



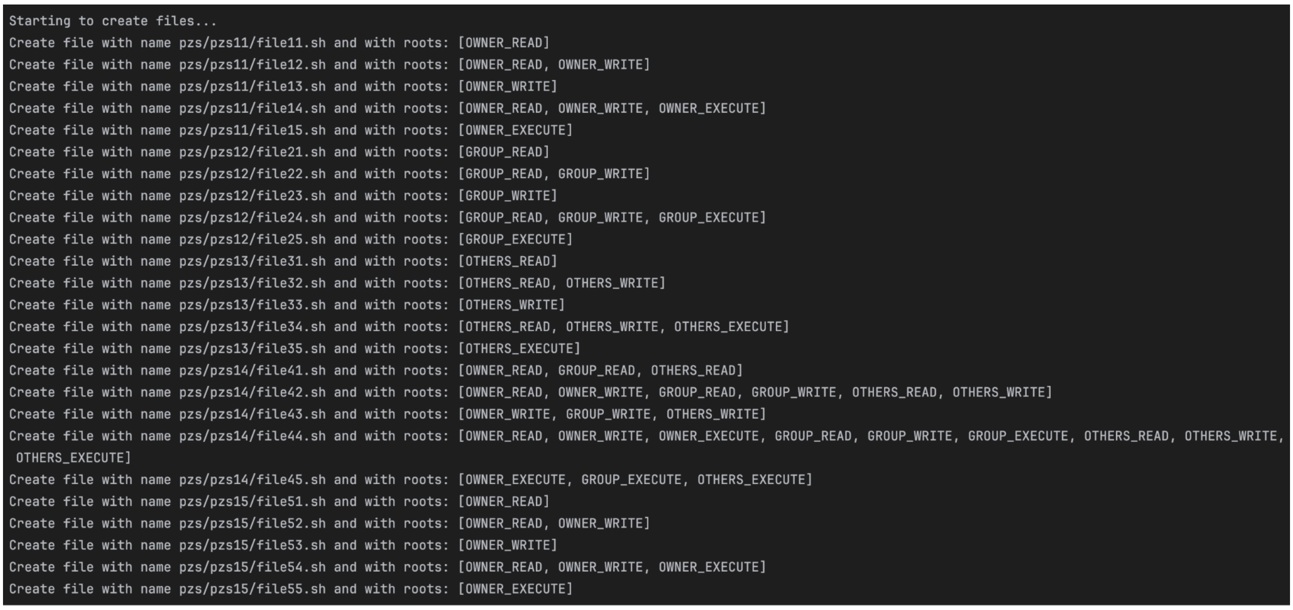
```
if (i == 17) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_WRITE,
            PosixFilePermission.GROUP_WRITE, PosixFilePermission.OTHERS_WRITE));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_WRITE,
            PosixFilePermission.GROUP_WRITE, PosixFilePermission.OTHERS_WRITE));
}

if (i == 18) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_READ,
            PosixFilePermission.OWNER_WRITE, PosixFilePermission.OWNER_EXECUTE,
            PosixFilePermission.GROUP_READ, PosixFilePermission.GROUP_WRITE,
            PosixFilePermission.GROUP_EXECUTE, PosixFilePermission.OTHERS_READ,
            PosixFilePermission.OTHERS_WRITE, PosixFilePermission.OTHERS_EXECUTE));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_READ,
            PosixFilePermission.OWNER_WRITE, PosixFilePermission.OWNER_EXECUTE,
            PosixFilePermission.GROUP_READ, PosixFilePermission.GROUP_WRITE,
            PosixFilePermission.GROUP_EXECUTE, PosixFilePermission.OTHERS_READ,
            PosixFilePermission.OTHERS_WRITE, PosixFilePermission.OTHERS_EXECUTE));
}

if (i == 19) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_EXECUTE,
            PosixFilePermission.GROUP_EXECUTE, PosixFilePermission.OTHERS_EXECUTE));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_EXECUTE,
            PosixFilePermission.GROUP_EXECUTE, PosixFilePermission.OTHERS_EXECUTE));
}
```



```
if (i == 20) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_READ));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_READ));
}

if (i == 21) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_READ,
            PosixFilePermission.OWNER_WRITE));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_READ,
            PosixFilePermission.OWNER_WRITE));
}

if (i == 22) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_WRITE));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_WRITE));
}

if (i == 23) {
    Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_WRITE,
            PosixFilePermission.OWNER_READ, PosixFilePermission.OWNER_EXECUTE));

    System.out.println(EnumSet.of(PosixFilePermission.OWNER_WRITE,
            PosixFilePermission.OWNER_READ, PosixFilePermission.OWNER_EXECUTE));
}
```

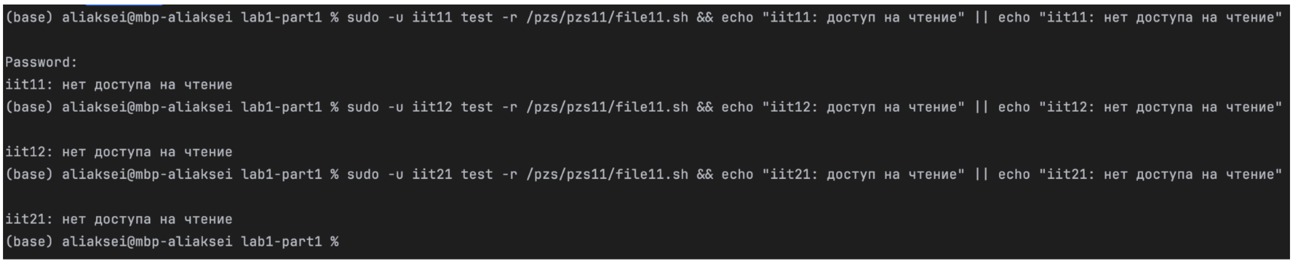


```
            if (i == 24) {
                Files.setPosixFilePermissions(fileNames.get(i), EnumSet.of(PosixFilePermission.OWNER_EXECUTE));

                System.out.println(EnumSet.of(PosixFilePermission.OWNER_EXECUTE));
            }
        }
    } catch (Exception exception) {
        System.out.println("Cannot create directory name '" + currentFileName + "' because " +
                exception.getMessage());
    }
}
```



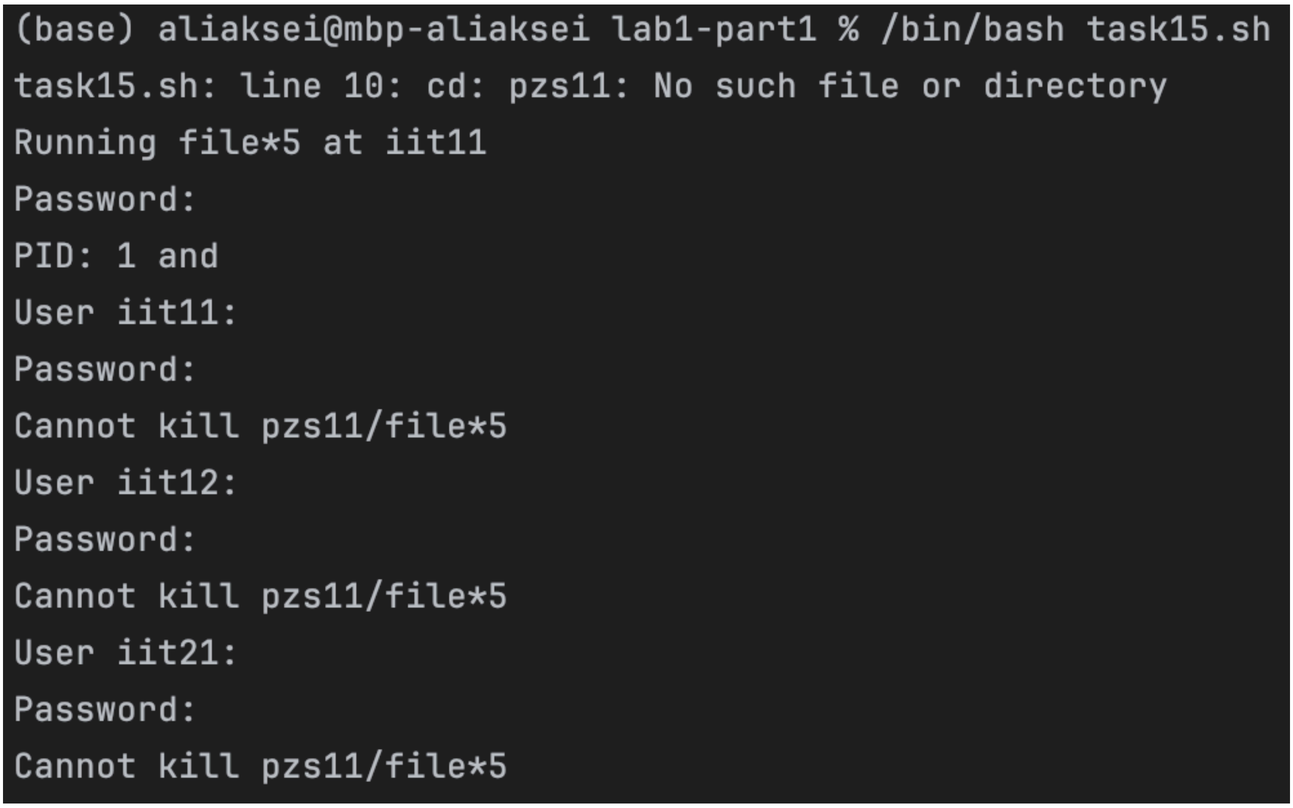
```
Starting to create files...
Create file with name pzs/pzs11/file11.sh and with roots: [OWNER_READ]
Create file with name pzs/pzs11/file12.sh and with roots: [OWNER_READ, OWNER_WRITE]
Create file with name pzs/pzs11/file13.sh and with roots: [OWNER_WRITE]
Create file with name pzs/pzs11/file14.sh and with roots: [OWNER_READ, OWNER_WRITE, OWNER_EXECUTE]
Create file with name pzs/pzs11/file15.sh and with roots: [OWNER_EXECUTE]
Create file with name pzs/pzs12/file21.sh and with roots: [GROUP_READ]
Create file with name pzs/pzs12/file22.sh and with roots: [GROUP_READ, GROUP_WRITE]
Create file with name pzs/pzs12/file23.sh and with roots: [GROUP_WRITE]
Create file with name pzs/pzs12/file24.sh and with roots: [GROUP_READ, GROUP_WRITE, GROUP_EXECUTE]
Create file with name pzs/pzs12/file25.sh and with roots: [GROUP_EXECUTE]
Create file with name pzs/pzs13/file31.sh and with roots: [OTHERS_READ]
Create file with name pzs/pzs13/file32.sh and with roots: [OTHERS_READ, OTHERS_WRITE]
Create file with name pzs/pzs13/file33.sh and with roots: [OTHERS_WRITE]
Create file with name pzs/pzs13/file34.sh and with roots: [OTHERS_READ, OTHERS_WRITE, OTHERS_EXECUTE]
Create file with name pzs/pzs13/file35.sh and with roots: [OTHERS_EXECUTE]
Create file with name pzs/pzs14/file41.sh and with roots: [OWNER_READ, GROUP_READ, OTHERS_READ]
Create file with name pzs/pzs14/file42.sh and with roots: [OWNER_READ, OWNER_WRITE, GROUP_READ, GROUP_WRITE, OTHERS_READ, OTHERS_WRITE]
Create file with name pzs/pzs14/file43.sh and with roots: [OWNER_WRITE, GROUP_WRITE, OTHERS_WRITE]
Create file with name pzs/pzs14/file44.sh and with roots: [OWNER_READ, OWNER_WRITE, OWNER_EXECUTE, GROUP_READ, GROUP_WRITE, GROUP_EXECUTE, OTHERS_READ, OTHERS_WRITE,
 OTHERS_EXECUTE]
Create file with name pzs/pzs14/file45.sh and with roots: [OWNER_EXECUTE, GROUP_EXECUTE, OTHERS_EXECUTE]
Create file with name pzs/pzs15/file51.sh and with roots: [OWNER_READ]
Create file with name pzs/pzs15/file52.sh and with roots: [OWNER_READ, OWNER_WRITE]
Create file with name pzs/pzs15/file53.sh and with roots: [OWNER_WRITE]
Create file with name pzs/pzs15/file54.sh and with roots: [OWNER_READ, OWNER_WRITE, OWNER_EXECUTE]
Create file with name pzs/pzs15/file55.sh and with roots: [OWNER_EXECUTE]
```

14. Для каждого из созданных файлов проверить, можно ли прочитать, редактировать, запустить файл пользователям iit11, iit12, iit21, iit22, iit3, суперпользователем (root).



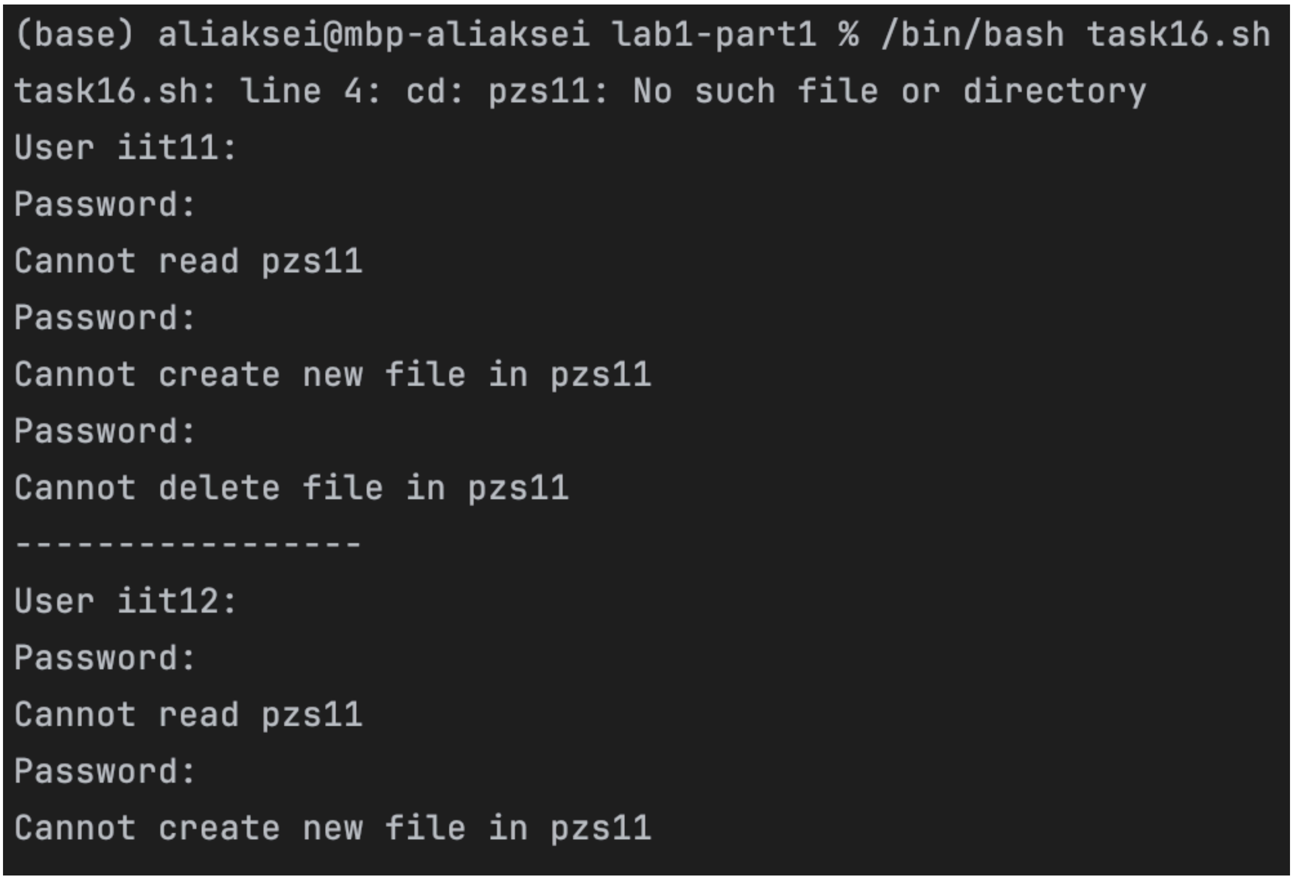
```
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo -u iit11 test -r /pzs/pzs11/file11.sh && echo "iit11: доступ на чтение" || echo "iit11: нет доступа на чтение"

Password:
iit11: нет доступа на чтение
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo -u iit12 test -r /pzs/pzs11/file11.sh && echo "iit12: доступ на чтение" || echo "iit12: нет доступа на чтение"

iit12: нет доступа на чтение
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo -u iit21 test -r /pzs/pzs11/file11.sh && echo "iit21: доступ на чтение" || echo "iit21: нет доступа на чтение"

iit21: нет доступа на чтение
(base) aliaksei@mbp-aliaksei lab1-part1 %
```

15. Запустить каждый из файлов, которые удовлетворяют шаблону «file*x*5» пользователем iit11. Проверить, можно ли остановить запущенный процесс пользователям iit11, iit12, iit21, iit22, iit3, суперпользователем (root).



```
(base) aliaksei@mbp-aliaksei lab1-part1 % /bin/bash task15.sh
task15.sh: line 10: cd: pzs11: No such file or directory
Running file*5 at iit11
Password:
PID: 1 and
User iit11:
Password:
Cannot kill pzs11/file*5
User iit12:
Password:
Cannot kill pzs11/file*5
User iit21:
Password:
Cannot kill pzs11/file*5
```

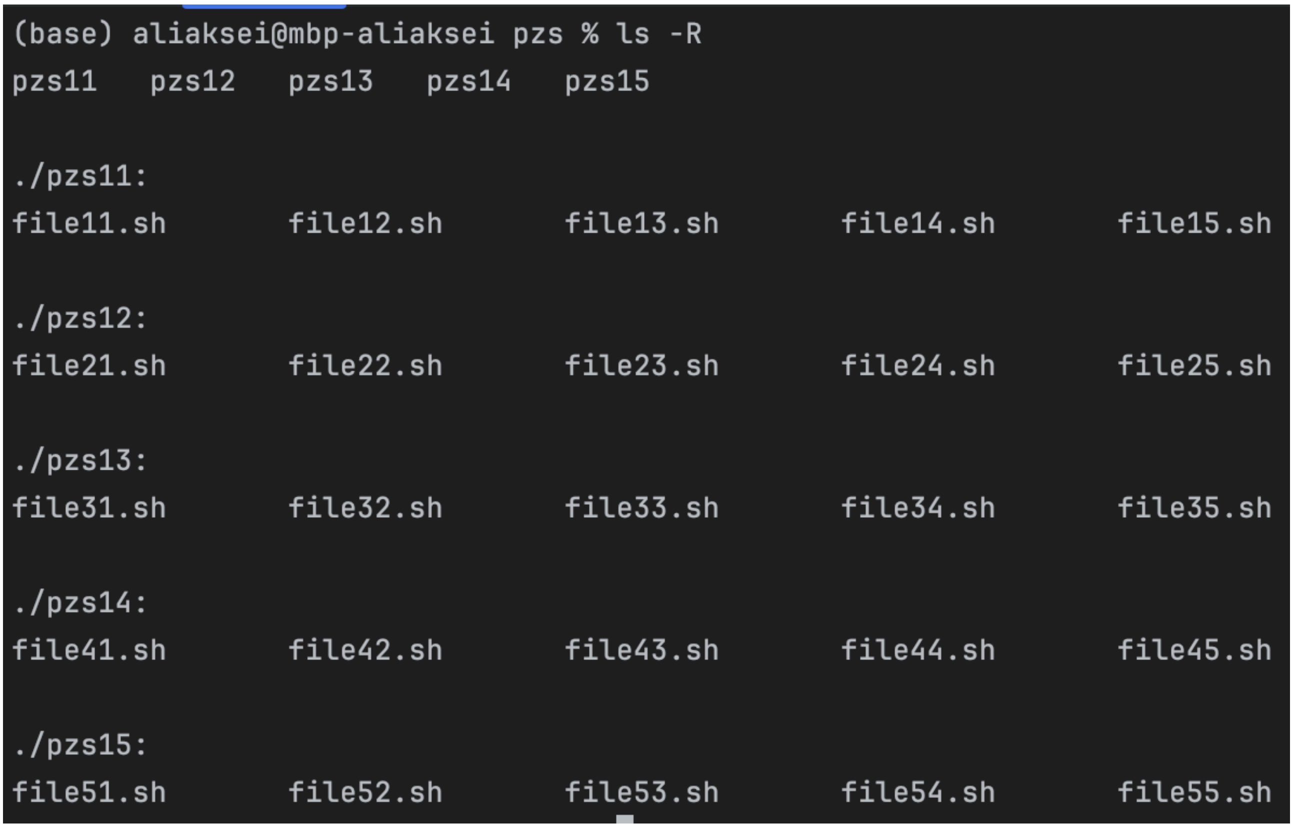
16. В каждой из созданных папок проверить, можно ли прочитать содержимое папок, создать новые файлы, удалить каждый из существующих файлов.



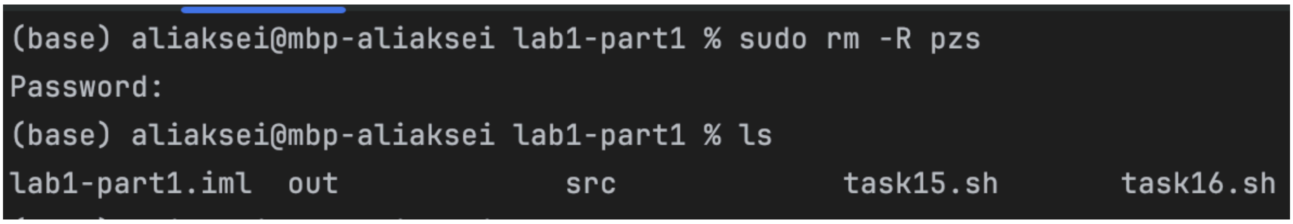
```
(base) aliaksei@mbp-aliaksei lab1-part1 % /bin/bash task16.sh
task16.sh: line 4: cd: pzs11: No such file or directory
User iit11:
Password:
Cannot read pzs11
Password:
Cannot create new file in pzs11
Password:
Cannot delete file in pzs11
-----------------
User iit12:
Password:
Cannot read pzs11
Password:
Cannot create new file in pzs11
```

17. Удалить созданные файлы, папки, пользователей iit11, iit12,

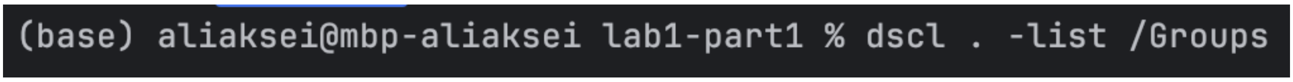
iit21, iit22, iit3, группы groupt_iit1, group_iit2.



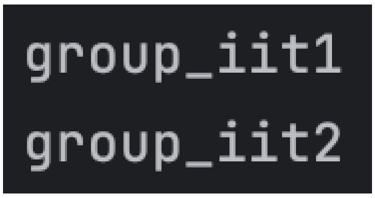
```
(base) aliaksei@mbp-aliaksei pzs % ls -R
pzs11   pzs12   pzs13   pzs14   pzs15

./pzs11:
file11.sh       file12.sh       file13.sh       file14.sh       file15.sh

./pzs12:
file21.sh       file22.sh       file23.sh       file24.sh       file25.sh

./pzs13:
file31.sh       file32.sh       file33.sh       file34.sh       file35.sh

./pzs14:
file41.sh       file42.sh       file43.sh       file44.sh       file45.sh

./pzs15:
file51.sh       file52.sh       file53.sh       file54.sh       file55.sh
```



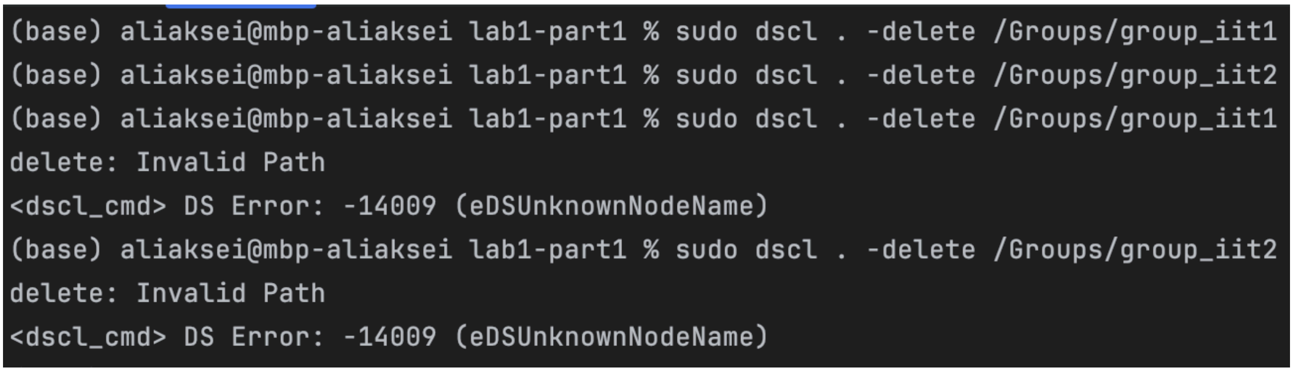
```
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo rm -R pzs
Password:
(base) aliaksei@mbp-aliaksei lab1-part1 % ls
lab1-part1.iml  out             src             task15.sh       task16.sh
```



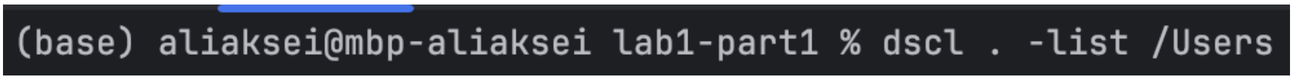
```
(base) aliaksei@mbp-aliaksei lab1-part1 % dscl . -list /Groups
```



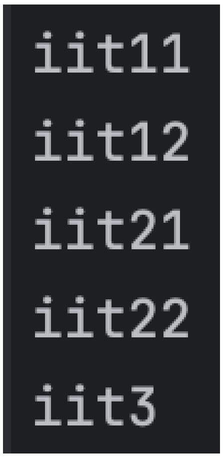
```
group_iit1
group_iit2
```



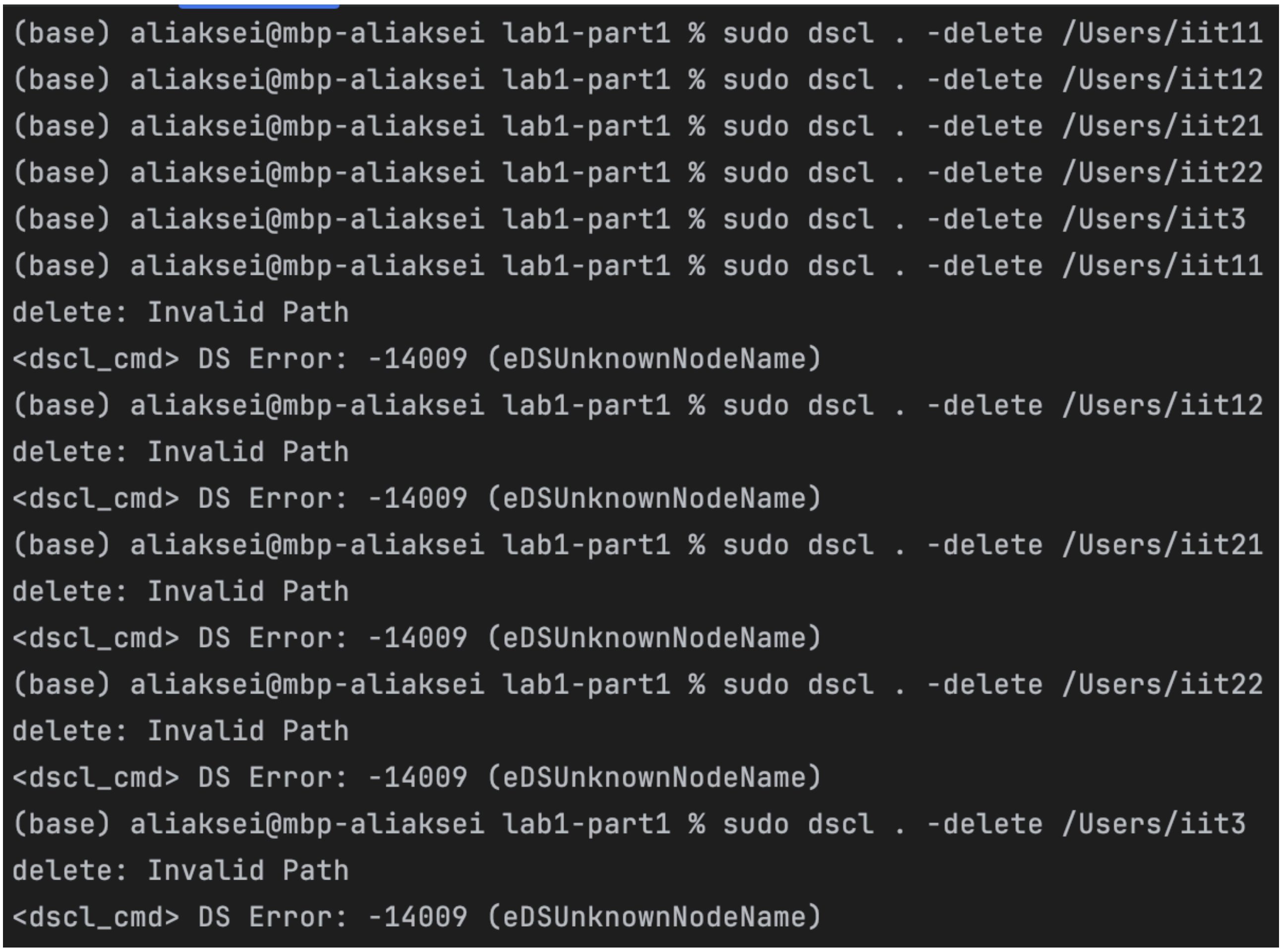
```
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Groups/group_iit1
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Groups/group_iit2
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Groups/group_iit1
delete: Invalid Path
<dscl_cmd> DS Error: -14009 (eDSUnknownNodeName)
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Groups/group_iit2
delete: Invalid Path
<dscl_cmd> DS Error: -14009 (eDSUnknownNodeName)
```



```
(base) aliaksei@mbp-aliaksei lab1-part1 % dscl . -list /Users
```



```
iit11
iit12
iit21
iit22
iit3
```



```
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Users/iit11
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Users/iit12
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Users/iit21
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Users/iit22
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Users/iit3
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Users/iit11
delete: Invalid Path
<dscl_cmd> DS Error: -14009 (eDSUnknownNodeName)
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Users/iit12
delete: Invalid Path
<dscl_cmd> DS Error: -14009 (eDSUnknownNodeName)
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Users/iit21
delete: Invalid Path
<dscl_cmd> DS Error: -14009 (eDSUnknownNodeName)
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Users/iit22
delete: Invalid Path
<dscl_cmd> DS Error: -14009 (eDSUnknownNodeName)
(base) aliaksei@mbp-aliaksei lab1-part1 % sudo dscl . -delete /Users/iit3
delete: Invalid Path
<dscl_cmd> DS Error: -14009 (eDSUnknownNodeName)
```

Вывод

В ходе работы я создание программу, которая позволяет настроить ACL для объектов файловой системы и проверить корректность её работы используя bash команды и файлы