

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС

Факультет управления

Кафедра Информатики и информационных технологий

Направление подготовки / специальность: 09.04.03 Прикладная информатика

Отчет по лабораторной работе

по дисциплине: Информационная безопасность

АВТОР

Обучающаяся 3 курса группы ИК-732
заочной формы обучения

(подпись) Насурллаев И. Е.
(фамилия, инициалы)

РУКОВОДИТЕЛИ

Гордеев Андрей Борисович
(ученая степень, ученое звание)

оценка _____
« _____ » _____ 2024 г.
(дата защиты)

(подпись) Гордеев А.Б..
(фамилия, инициалы)

Нижний Новгород, 2024г.

Оглавление

Лабораторная работа 1	3
Матрица доступа:	4
Логика получения прав.	5
Сценарий 1:.....	5
Сценарий 2:.....	7
Вывод	7
Практическая работа 1	8
Выполнение задания 1:	8
Выполнение задания 2:	10
Приложение	12
Класс User	12
Класс File	13
Класс Users.....	14
Класс Files	16
Модуль взаимодействия.....	20

Лабораторная работа 1

Название: Реализация дискреционной модели политики безопасности

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

Вариант: 7

Задание:

1. Получить данные о количестве пользователей и объектов компьютерной системы:
 - a. Кол-во пользователей – 9;
 - b. Кол-во объектов - 4.
2. Реализовать программный модуль, создающий матрицу доступа пользователей к объектам компьютерной системы.

Таблица 1. Пример матрицы доступа

Объект / Субъект	Файл_1	Файл_2	CD-RW	Дисковод
1. Администратор	Полные права	Полные права	Полные права	Полные права
2. Гость	Запрет	Чтение	Чтение	Запрет
3. Пользователь_1	Чтение, передача прав	Чтение, запись	Полные права	Запрет

3. Данный модуль должен выполнять следующие функции:
 - a. При запуске модуля должен запрашиваться идентификатор пользователя (проводится идентификация пользователя),
 - i. при успешной идентификации пользователя должен осуществляться вход в систему,
 - ii. при неуспешной – выводиться соответствующее сообщение.
 - b. При входе в систему после успешной идентификации пользователя на экране должен распечатываться список всех объектов системы с указанием перечня всех доступных прав доступа идентифицированного пользователя к данным объектам.
 - c. После вывода на экран перечня прав доступа пользователя к объектам компьютерной системы, необходимо организовать ожидание указаний пользователя на осуществление действий над объектами в компьютерной системе. После получения команды от пользователя, на экран необходимо вывести сообщение об успешности либо не успешности операции. При выполнении операции передачи прав (grant) должна модифицироваться матрица доступа. Программа должна поддерживать операцию выхода из системы (quit), после которой запрашивается идентификатор пользователя.

Код разработанного модуля представлен в приложении:

1. Класс User – параметры и действия для создания пользователей:
 - a. Методы *set* – установка параметров;
 - b. Методы *get* – получение значения параметров.
2. Класс File – параметры и действия для создания файлов:
 - a. Методы *set* – установка параметров;
 - b. Методы *get* – получение значения параметров.
3. Класс Users – взаимодействие с несколькими пользователями
4. Класс Files – взаимодействие с несколькими файлами
5. Модуль взаимодействия – интерфейс взаимодействия с пользователем.

Создадим пользователей:

```
u = Users()
u.generate_users(COUNT_USER)
u.print_all_persen()
```

Username	Password	Type
Матвей	Матвей	admin
Александр	Александр	user
Артём	Артём	user
Платон	Платон	user
Владимир	Владимир	user
Анна	Анна	user
Андрей	Андрей	user
Тимофей	Тимофей	user
Полина	Полина	guest

Матрица доступа:

```
f = Files()
f.generate_files(COUNT_OBJECT, u)
f.print_all_access_to_file()
```

Username	PersenData	Access	Object_1	Object_2
Матвей	Full	Full	Full	Full
Александр	Denied	r--	Denied	r--
Артём	Denied	r--	Denied	r--
Платон	Denied	r--	rw-	Denied
Владимир	Denied	r--	r--	Full
Анна	Denied	r--	rw-	r--
Андрей	Denied	r--	r--	r--
Тимофей	Denied	r--	Denied	rw-
Полина	Denied	Denied	Denied	Denied

Full – Полный доступ

Denied – Доступ запрещен

r – чтение разрешено

w – запись разрешена

s – передача доступов разрешена

Логика получения прав.

Полный доступ может иметь администратор и владелец файла.

Доступ запрещен, соответствующими правами обладает гость и пользователь, которому запрещен доступ к файлу.

Сценарий 1:

Зайдем под администратором (в данном случае Матвей)

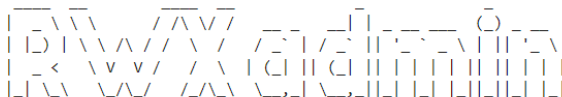


```
1 - Autorization
0 - Exit
```

```
Enetre value: 1
Insert you`re login and password
Login: Матвей
```

Passwor:

Авторизация прошла успешно, в названии программы появился тип пользователя и приветственное сообщение.



Hello Матвей!

Access for file

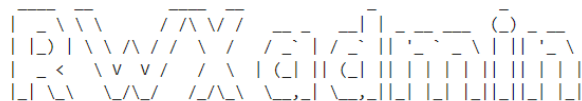
Username	PersenData	Access	Object_1	Object_2
Матвей	Full	Full	Full	Full
Александр	Denied	r--	Denied	r--
Артём	Denied	r--	Denied	r--
Платон	Denied	r--	rw-	Denied
Владимир	Denied	r--	r--	Full
Анна	Denied	r--	rw-	r--
Андрей	Denied	r--	r--	r--
Тимофей	Denied	r--	Denied	rw-
Полина	Denied	Denied	Denied	Denied

```
1 - Read
2 - Write
3 - Share
0 - Exit
```

Enetre value:

Теперь необходимо выбрать действие, допустим редактирование. Для этого необходимо ввести код действия (2).

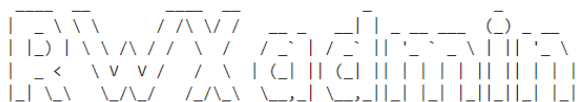
После чего открывается окно в котором надо ввести имя файла, например – PersenData



Write name file or enter '0' for exit

Enetre value:

Так как администратор имеет полный доступ ко всем объектам системы, получаем сообщение об успешном выполнении действия.



Success

USER: Матвей

FILE_NAME: PersenData

Enter '0' for rotate to back

Enetre value:

Сценарий 2:

Зайдем под пользователем (в данном случае Анна)

Попробуем произвести редактирование.



```
Success
USER:  Анна
FILE_NAME:  Object_1

Enter '0' for rotate to back
```

Видим сообщение об успешном выполнении действия.

Попробуем передать права другому пользователю, напомним у Анны таких прав нет.



```
Access denied
USER:  Анна
FILE_NAME:  Object_1

Enter '0' for rotate to back
```

Видим сообщение о том что в доступе отказано.

Вывод

Ознакомился с проблемами реализации политик безопасности в компьютерных системах на примере дискреционной модели.

Практическая работа 1

Название: Организация защиты информации на персональном компьютере

Цель: Приобретение практических навыков использования программных средств защиты для предотвращения утечки информации

Выполнение задания 1:

Создал docx документ и ввел информацию

СВОДНАЯ ВЕДОМОСТЬ №1
НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ
За 9-месяц 2024 года

Разрыв раздела (на текущей странице)

Табельный номер	ФИО	Код подразделения	Должность	Оклад	Процент премии	Начислено
1	Антонов А.А.	1	Начальник отдела	50 000	10	
2	Бородин Г.Д.	1	Инженер	20 000	20	
3	Воронина И.С.	2	Бухгалтер	30 000	10	
4	Громова А.А.	3	Инспектор	20 000	15	
5	Ефимов О.Н.	2	Главный бухгалтер	40 000	20	
6	Ильина П.А.	3	Инспектор	30 000	10	
7	Колосова В.В.	2	Бухгалтер	12 000	25	
8	Морозов В.Л.	2	Бухгалтер	15 000	30	
9	Титова А.Р.	3	Начальник отдела	40 000	5	
10	Фролов И.А.	2	Бухгалтер	20 000	10	

Разрыв раздела (на текущей странице)

Дата создания ведомости 09.09.2024

Руководитель организации Пупкин И.П.

Главный бухгалтер Коркин Г.Г.

Ограничил редактирование:

Защита раздела

Защищенные разделы:

- ☒ Раздел 1
- ☒ Раздел 2
- ☒ Раздел 3

Параметры...

2. Ограничения на редактирование

☒ Разрешить только указанный способ редактирования документа:

Ввод данных в поля форм

Выбор разделов...

Включил защиту не устанавливая пароль

Включить защиту

Способ защиты

☒ Пароль
(Документ не зашифрован. Пользователи-злоумышленники могут изменить файл и удалить пароль.)

Новый пароль (необязательно):

Подтверждение пароля:

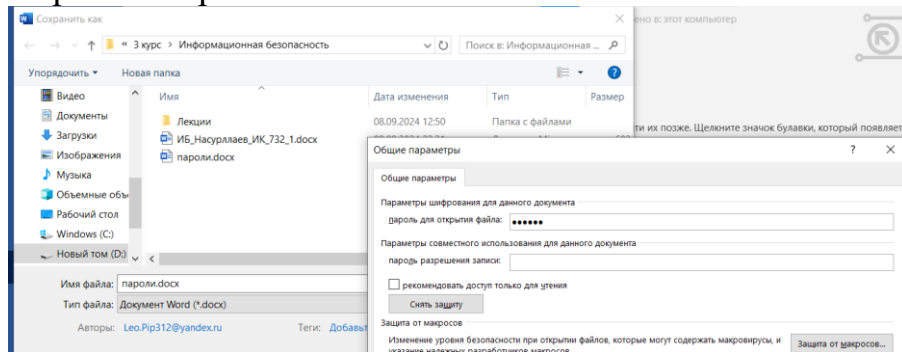
☐ Проверка подлинности пользователя
(Авторизованные владельцы могут снять защиту документа. Документ зашифрован, доступ ограничен.)

OK Отмена

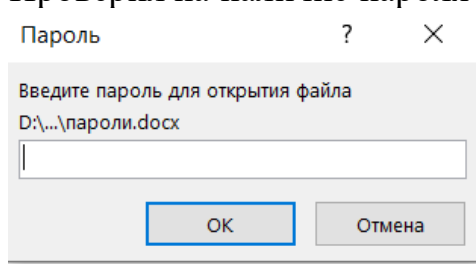
Проверил что редактировать можно только указанные поля

СВОДНАЯ ВЕДОМОСТЬ №1 ТАБЛИЦА ЗАРАБОТНОЙ ПЛАТЫ

Сохранил файл с добавлением пароля на открытие
Пароль – пароли



Проверил на наличие пароля при запуске файла



Выполнение задания 2:

Создал файл Организация с 3 листами:

1. Ведомость,
2. Подразделения,
3. Статистика

В лист Ведомость перенес данные из doc файла

Заполнил столбец «Начислено» при помощи формулы

G2		=E2*(1+F2/100)	
	A	B	C
	Табельный номер	ФИО	Код подразделения
1			
2	1	Антонов А.А.	1
3	2	Бородин Г.Д.	1
4	3	Воронина И.С.	2
5	4	Громова А.А.	3
6	5	Ефимов О.Н.	2
7	6	Ильина П.А.	3
8	7	Колосова В.В.	2
9	8	Морозов В.Л.	2
10	9	Титова А.Р.	3
11	10	Фролов И.А.	2
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			

Заполнил лист «Подразделение»

A	B	C
Подразделения		
Код подразделения	Наименование	Руководитель
1	Отдел сбыта	Антонов А.А.
2	Бухгалтерия	Ефимов О.Н.
3	Отдел кадров	Титова А.Р.

Заполнил лист «Статистика»

Код подразделения	2
	Статистика
Наименование подразделения	Бухгалтерия
Руководитель	Ефимов О.Н.
Число сотрудников	5
Средняя зарплата	23400
Наибольшая зарплата	40000
Наименьшая зарплата	12000

Заполнил формулы в листе «Статистика»

2		Статистика
3	Наименование подразделения	Бухгалтерия
4	Руководитель	ВПР(B1;Подразделения!A3:C5;3)
5	Число сотрудников	СЧЁТЕСЛИ(Ведомость!C2:C11;B1)
6	Средняя зарплата	СУММЕСЛИ(Ведомость!C2:C11;B1;Ведомость!E2:E11)/B5
7	Наибольшая зарплата	МАКСЕСЛИ(Ведомость!E2:E11;Ведомость!C2:C11;B1)
8	Наименьшая зарплата	МИНЕСЛИ(Ведомость!E2:E11;Ведомость!C2:C11;B1)

Приложение

Класс User

```
class User:

    user_name = ''
    password = ''
    type = ''

    def __init__(self,
                  user_name = '',
                  password = '',
                  type = 'user'
                  ):

        self.user_name = user_name
        self.password = password
        self.type = type

    def set_username(self, name):
        self.user_name = name

    def get_username(self):
        return self.user_name

    def set_password(self, password):
        self.password = password

    def get_password(self):
        return self.password

    def set_type(self, type):
        self.type = type

    def get_type(self):
        return self.type
```

Класс File

```
class File:

    file_name = ''
    rolls = {}

    def __init__(self,
                  file_name = '',
                  rolls = {},
                  ):

        self.file_name = file_name
        self.rolls = rolls

    def set_filename(self, name):
        self.file_name = name

    def get_filename(self):
        return self.file_name

    def set_rolls(self, rolls):
        self.rolls = rolls

    def get_rolls(self):
        return self.rolls
```

Класс Users

```
class Users:

    # Список пользователей

    list_users = []

    BAD_CODE = -1

    SUCESS = 200

    def __init__(self):
        self.list_users = []

    def generate_users(self, count):
        def _generate_name():
            PATH = r'./List_names.txt'
            try:
                f = open(PATH, 'r', encoding='utf-8')
                names = [name.replace('\n', '') for name in f.readlines()]
                f.close()
            except:
                names = ['Иван', 'Кристина', 'Ксения',
                        'Тимофей', 'Виктория', 'Мария',
                        'Александр', 'Алексей', 'Максим']
            name = choice(names)
            names.remove(name)
            return name

        self.list_users = [userClass.User() for i in range(0, count)]

        for index, user in enumerate(self.list_users):
            name = _generate_name()
            user.set_username(name)
            user.set_password(name)
            if index == 0:
                user.set_type('admin')
            if index == len(self.list_users)-1:
                user.set_type('guest')

    def print_all_persen(self):
        # ВЫВОД
```

```

len_usn = max([len(user.user_name) for user in self.list_users] +
[8])
len_pass = max([len(user.password) for user in self.list_users] +
[8])
len_type = max([len(user.type) for user in self.list_users] + [5])

sep_row = '|'+'-'*(len_usn+4)+'|'+'-'*(len_pass+4)+'|'+'-'
'* (len_type+4)+'|'
print(sep_row)
print('| Username', ' '*(len_usn-8+1),
      '| Password', ' '*(len_pass-8+1),
      '| Type', ' '*(len_type-4+1),
      '| '
      )
print(sep_row)
for u in self.list_users:
    name = u.get_username()
    password = u.get_password()
    type = u.get_type()

    print(f"| {name}", ' '*(len_usn-len(name)+1), '|', end='')
    print(f" {password}", ' '*(len_pass-len(name)+1), '|', end='')
    print(f" {type}", ' '*(len_type-len(type)+1), '|')
print(sep_row)

def authorization(self, username, password):
    list_usernames = [user.user_name for user in self.list_users]
    if username not in list_usernames: return self.BAD_CODE, 1

    user = self.list_users[list_usernames.index(username)]

    if user.get_password() == password: return self.SUCCESS, user
    else: return self.BAD_CODE, 2

```

Класс Files

```
class Files:

    # Список файлов
    list_files = []
    BAD_CODE = -1
    SUCESS = 200

    def __init__(self):
        self.list_files = []

    def generate_files(self, count, users):
        def _generate_rolls_user():
            acess = [True, False]
            for index in range(0,3):
                user = {'r':'', 'w':'', 's':''}
                value = choice(acess)
                if value == False:
                    return {'r':False, 'w':False, 's':False}
                user['r'] = value
                value = choice(acess)
                if value == False:
                    return {'r':True, 'w':False, 's':False}
                user['w'] = value
                user['s'] = choice(acess)
            return user

        self.list_files = [fileClass.File() for i in range(0, count)]

        admin_access = {'r':True, 'w':True, 's':True}
        user_access = {'r':True, 'w':False, 's':False}
        guest_access = {'r':False, 'w':False, 's':False}

        for index, file in enumerate(self.list_files):
            if index == 0:
                file.set_filename('PersenData')

                admin = {user.get_username():admin_access for user in
users.list_users if user.get_type() == 'admin'}
```



```

        guest = {user.get_username():guest_access for user in
users.list_users if user.get_type() != 'admin'}

        file.set_rolls(
            dict(list(admin.items())+list(guest.items()))
        )

    elif index == 1:
        file.set_filename('Access')

        admin = {user.get_username():admin_access for user in
users.list_users if user.get_type() == 'admin'}

        user = {user.get_username():user_access for user in
users.list_users if user.get_type() == 'user'}

        guest = {user.get_username():guest_access for user in
users.list_users if user.get_type() != 'admin' and user.get_type() != 'user'}

        file.set_rolls(
dict(list(admin.items())+list(user.items())+list(guest.items()))
        )
    else:
        file.set_filename(f'Object_{index-1}')

        admin = {user.get_username():admin_access for user in
users.list_users if user.get_type() == 'admin'}

        user = {user.get_username():generate_rolls_user() for user
in users.list_users if user.get_type() == 'user'}

        guest = {user.get_username():guest_access for user in
users.list_users if user.get_type() != 'admin' and user.get_type() != 'user'}

        file.set_rolls(
dict(list(admin.items())+list(user.items())+list(guest.items()))
        )

def print_all_access_to_file(self):
    # ВЫВОД
    usns = list(self.list_files[0].rolls.keys())
    len_usn = max([len(user) for user in usns] + [8])
    len_pers = len("PersenData")
    len_acc = len("Access")

```

```

        sep_row = '|'+'-'*(len_usn+4)+'|'+'-'*(len_pers+4)+'|'+'-'
        *(len_acc+6)+'|'
        print(sep_row, end='')

    for index, file in enumerate(self.list_files):
        if index < 2:
            continue
        print('-'*(len(file.file_name)+2), end='|')
    print()

    print('| Username', ' '*(len_usn-8+1),
          '| PersenData', ' '*(len_pers-10+1),
          '| Access ', ' '*(len_acc-6+2),
          '|', end=' ')

    for index, file in enumerate(self.list_files):
        if index < 2:
            continue

        print(f'{file.file_name}', end = ' | ')

    print()
    print(sep_row, end='')

    for index, file in enumerate(self.list_files):
        if index < 2:
            continue
        print('-'*(len(file.file_name)+2), end='|')
    print()

    for user in usns:
        print(f"| {user}", ' '*(len_usn-len(user)+2), end='|')
        for index, file in enumerate(self.list_files):
            access = list(file.rolls[user].values())
            if sum(access) == 3:
                if index == 0:
                    print(" Full", ' '*(len_pers-len("Full")+2),end='|')

```

```

        elif index == 1:
            print(" Full", ' '*(len_acc-len("Full")+4),end='|')
        else:
            print(" Full", ' '*(len(file.file_name)-len("Full")),
end='|')

        elif sum(access) == 0:
            if index == 0:
                print(" Denied", ' '*(len_pers-
len("Denied")+2),end='|')
            elif index == 1:
                print(" Denied", ' '*(len_acc-len("Denied")+4),
end='|')
            else:
                print(" Denied", ' '*(len(file.file_name)-
len("Denied")), end='|')
            else:
                for index_role, value in enumerate(access):
                    if value == False: access[index_role] = '-'
                    elif index_role == 0: access[index_role] = 'r'
                    elif index_role == 1: access[index_role] = 'w'
                    elif index_role == 2: access[index_role] = 's'
                access = "".join(access)
                if index == 1:
                    print(f' {access}', ' '*(len_acc-len(access)+4),
end='|')
                else:
                    print(f' {access}', ' '*(len(file.file_name)-
len(access)), end='|')
                print()
            print(sep_row, end='')
        for index, file in enumerate(self.list_files):
            if index < 2:
                continue
            print('-'*(len(file.file_name)+2), end='|')

def access_verificate(self, username, filename, action):
    list_filenames = [file.file_name for file in self.list_files]
    file = self.list_files[list_filenames.index(filename)]
    if file.rolls[username][action]: return self.SUCCESS
    else: return self.BAD_CODE

```

Модуль взаимодействия

```
import sys

sys.intern('./__init__.py')

import User as userClass
import File as fileClass

from IPython.display import clear_output
from time import sleep
from tqdm import tqdm
from random import choice
from art import tprint

COUNT_USER = 9
COUNT_OBJECT = 4
flag_exit = 1

u = Users()
u.generate_users(COUNT_USER)

f = Files()
f.generate_files(COUNT_OBJECT, u)
flag_exit = 1
page = 1

while flag_exit != 0:
    if page == 1:
        tprint("RWX")
        print("""
1 - Autorization
0 - Exit

""")

    elif page == 2:
        clear_output()
        tprint(f"RWX {user.get_type()}")

        print(f"Hello {user.get_username()}!")
        print()
```

```

        print("Access for file")
        f.print_all_access_to_file()

        print()
        print("""
1 - Read
2 - Write
3 - Share
0 - Exit

        """)

    elif page == 3:
        clear_output()
        tprint(f"RWX {user.get_type()}")

        print("""
Write name file or enter '0' for exit

        """)

    elif page == 4:
        clear_output()
        tprint(f"RWX {user.get_type()}")

        if code == f.SUCESS: print("Success")
        else: print("Access denied")

        print("USER: ", user.get_username())
        print("FILE_NAME: ", name_file)

        print()
        print("""Enter '0' for rotate to back""")

        print()

    if page == 3:
        try:
            name_file = input("Enetre value:", )

```

```

except:
    clear_output()
    continue
else:
    try:
        value = int(input("Entre value:", ))
    except:
        clear_output()
        continue

if page <= 2 and value == 0:
    print("Bay!")
    flag_exit = 0

elif value == 1 and page == 1:
    print("Insert you`re login and password")
    login = input("Login: ", )
    password = input("Passwor: ",)

    code, user = u.authorization(login, password)
    if code == u.BAD_CODE: print("Uncorect username or password")
    elif code == u.SUCESS:
        for i in tqdm(range(0,100)):
            sleep(0.01)
        page += 1
    else: print("Unknown error")

elif page == 2 and value != 0:
    action = value
    value = ''
    page +=1

elif page == 4 and value == 0:
    page = 2

```

```

elif page == 3 and name_file == '0':
    page -= 1

elif page == 3:
    if user.get_type == 'admin':
        print("Success")
        if name_file == 'PersenData':
            u.print_all_persen()
        elif name_file == 'Access':
            f.print_all_access_to_file()

    elif user.get_type == 'Guest':
        print("Access denied")

    else:
        # READ
        if action == 1:
            code = f.access_verificate(user.get_username(), name_file,
'r')

        # Write
        elif action == 2:
            code = f.access_verificate(user.get_username(), name_file,
'w')

        # Share
        elif action == 3:
            code = f.access_verificate(user.get_username(), name_file,
's')

    page += 1

```