

Лабораторная работа №10

Lab 10.1 Создайте базовый класс User, у которого есть:

1. метод `__init__`, принимающий имя пользователя и его роль. Их необходимо сохранить в атрибуты экземпляра `name` и `role` соответственно

Затем создайте класс Access, у которого есть:

1. приватный атрибут класса `__access_list`, в котором хранится список ['admin', 'developer']
2. приватный статик-метод `__check_access`, который принимает название роли и возвращает True, если роль находится в списке `__access_list`, иначе - False
3. публичный статик-метод `get_access`, который должен принимать экземпляр класса User и проверять есть ли доступ у данного пользователя к ресурсу при помощи метода `__check_access`. Если у пользователя достаточно прав, выведите на экран сообщение «User <name>: success», если прав недостаточно - «AccessDenied». Если передается тип данных отличный от экземпляр класса User необходимо вывести сообщение «AccessTypeError»

```
user1 = User('batya99', 'admin')
```

```
Access.get_access(user1) # печатает "User batya99: success"
```

```
zaya = User('milaya_zaya999', 'user')
```

```
Access.get_access(zaya) # печатает AccessDenied
```

```
Access.get_access(5) # печатает AccessTypeError
```

Lab 10.2 Задача Корзина - ситуация создания файлов и перемещения их в корзину.

Создайте класс File, у которого есть:

1. метод `__init__`, который должен принимать на вход имя файла и записывать его в атрибут `name`. Также необходимо создать атрибуты `in_trash`, `is_deleted` и записать в них значение False
2. метод `restore_from_trash`, который печатает фразу «Файл {name} восстановлен из корзины» и проставляет атрибут `in_trash` в значение False
3. метод `remove`, который печатает фразу «Файл {name} был удален» и проставляет атрибут `is_deleted` в значение True
4. метод `read`, который
 - печатает фразу «ErrorReadFileDeleted({name})», если атрибут `is_deleted` истин, и выходит из метода
 - печатает фразу «ErrorReadFileTrashed({name})», если атрибут `in_trash` истин, и выходит из метода
 - печатает фразу «Прочитали все содержимое файла {self.name}» если файл не удален и не в корзине
5. метод `write`, который принимает значение `content` для записи и
 - печатает фразу «ErrorWriteFileDeleted({name})», если атрибут `is_deleted` истин, и выходит из метода

- печатает фразу «ErrorWriteFileTrashed({name})», если атрибут `in_trash` истин, и выходит из метода
- печатает фразу «Записали значение {content} в файл {self.name}», если файл не удален и не в корзине

Далее создайте класс `Trash` у которого есть:

1. атрибут класса `content` изначально равный пустому списку
2. статик-метод `add`, который принимает файл и сохраняет его в корзину: для этого нужно добавить его в атрибут `content` и проставить файлу атрибут `in_trash` значение `True`. Если в метод `add` передается не экземпляр класса `File`, необходимо вывести сообщение «В корзину добавлять можно только файл»
3. статик-метод `clear`, который запускает процесс очистки файлов в корзине. Необходимо для всех файлов, хранящийся в атрибуте `content`, в порядке их добавления в корзину вызвать метод файла `remove`. Атрибут `content` после очистки должен стать пустым списком. Сама процедура очистки должна начинаться фразой «Очищаем корзину» и заканчиваться фразой «Корзина пуста»
4. статик-метод `restore`, который запускает процесс восстановления файлов из корзины. Необходимо для всех файлов, хранящийся в атрибуте `content`, в порядке их добавления в корзину вызвать метод файла `restore_from_trash`. Атрибут `content` после очистки должен стать пустым списком. Сама процедура восстановления должна начинаться фразой «Восстанавливаем файлы из корзины» и заканчиваться фразой «Корзина пуста»

Придумайте экземпляр класса и реализуйте через него работу (вызов) каждого созданного вами метода.