Практические задание.

Создайте систему управления пользователями с различными уровнями доступа.

1. Родительский класс User

- Содержит username, email и role (роль пользователя: "admin" или "user").
- Метод get_info(), который возвращает информацию о пользователе (но без пароля).

2. Дочерний класс UserAccount

- Наследует User и добавляет инкапсулированный пароль (__password).
- Реализует методы:

  set_password(new_password), который устанавливает новый пароль.

  check_password(password), который проверяет правильность пароля.

  reset_password(new_password), который позволяет сбросить пароль только если пользователь – админ.

  increase_failed_attempts(), который увеличивает счётчик неудачных попыток входа. Если три попытки подряд неверные – блокируем аккаунт.

Дополнительные пояснения

1. Что такое инкапсуляция и зачем она здесь?

Инкапсуляция – это сокрытие данных от прямого доступа, чтобы защитить их от случайного или намеренного изменения.
В этом задании:

- Пароль (__password) скрыт, и его нельзя прочитать напрямую (user1.__password вызовет ошибку).
- Счётчик неудачных попыток (__failed_attempts) тоже закрыт, чтобы нельзя было сбросить его вручную.
- Переменная __blocked предотвращает попытки входа после трёх неудачных попыток.

2. Почему используется родительский класс User?

Родительский класс User нужен, чтобы:

- Разделить **основные данные пользователя** (логин, email, роль) и **функции аккаунта** (пароль, вход).
- Позволить в будущем создать **другие типы пользователей** без изменения основного кода (например, GuestUser, ModeratorUser).
- Сделать код **чистым и удобным для расширения** (принцип **DRY – Don't Repeat Yourself**).

3. Как работают проверки и ограничения?

1. **Установка пароля**
   - Если аккаунт заблокирован, установить новый пароль нельзя.
   - При установке пароля счётчик неудачных попыток сбрасывается.
2. **Проверка пароля**
   - Если пароль неверный, увеличивается счётчик __failed_attempts.
   - После трёх неудачных попыток аккаунт блокируется (__blocked = True).
3. **Сброс пароля**
   - Только администратор может сбросить пароль (reset_password()).
   - При сбросе пароля счётчик ошибок обнуляется, и аккаунт разблокируется.

Пример входных данных:

```
# === Пример использования ===

user1 = UserAccount("Alice", "alice@example.com")

admin = UserAccount("AdminUser", "admin@example.com", role="admin")



# Устанавливаем пароль

user1.set_password("securePass123")



# Проверка входа

user1.check_password("wrongPass")  # Попытка 1

user1.check_password("wrongPass")  # Попытка 2

user1.check_password("wrongPass")  # Попытка 3 (БЛОКИРОВКА)



# Попытка входа после блокировки

user1.check_password("securePass123")  # Уже не работает
```

```
# Сброс пароля админом
admin.reset_password("newAdminPass")  # Нет ошибки, но не меняет пароль user1
user1.reset_password("newUserPass")  # Ошибка, т.к. user1 не админ
admin.set_password("adminSecurePass")  # Меняем свой пароль
admin.check_password("adminSecurePass")  # Успешный вход
```

Выходные данные:

```
Пароль успешно установлен.
Неверный пароль! Попытка 1/3.
Неверный пароль! Попытка 2/3.
Неверный пароль! Попытка 3/3.
Аккаунт заблокирован из-за 3 неудачных попыток.
Аккаунт заблокирован.
Пароль сброшен администратором.
У вас нет прав для сброса пароля!
Пароль успешно установлен.
Вход выполнен успешно!
```