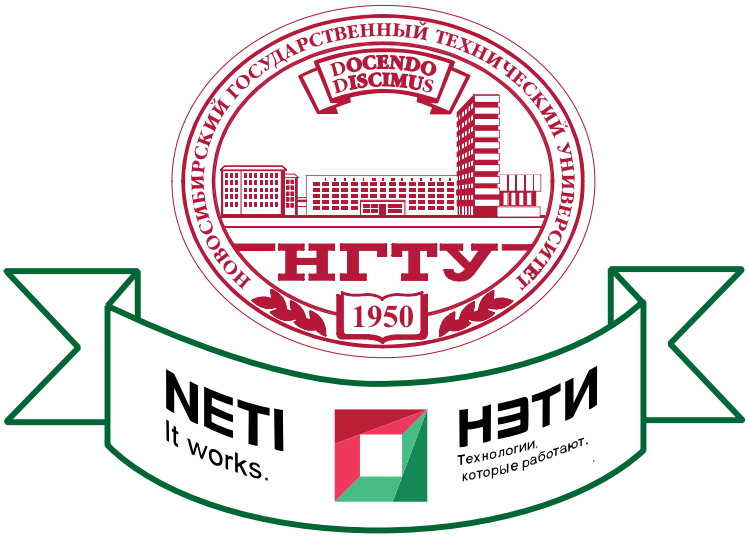
Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

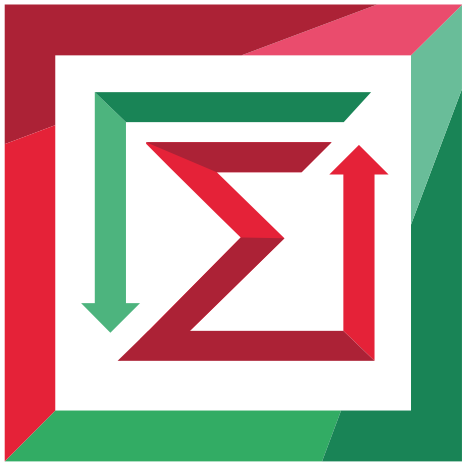


Кафедра теоретической и прикладной информатики

Лабораторная работа № 5

по дисциплине «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

РОЛЕВОЕ УПРАВЛЕНИЕ ДОСТУПОМ. РАЗРАБОТКА ЗАЩИЩЁННЫХ ПРИЛОЖЕНИЙ



Факультет:	ПМИ
Группа:	ПМИ-02
Вариант:	6
Студенты:	Сидоров Даниил,
	Дюков Богдан
Преподаватели:	Авдеенко Татьяна Владимировна,
	Кутузова Ирина Александровна.

Новосибирск

2026

1. Цель работы

Научиться тестировать полученную последовательность на равномерность и случайность. Познакомиться с концепцией ролевого управления доступом и способами защиты программного обеспечения от существующих угроз. Научиться разрабатывать приложения, которые используют ролевое управление доступом для разграничения полномочий пользователей. Получить навыки защиты разработанной программы от несанкционированного копирования и других угроз, которым может подвергаться программное обеспечение.

2. Задача

I. Реализовать приложение с графическим интерфейсом, удовлетворяющее следующим требованиям.

1. Приложение проводит аутентификацию пользователя.
2. Каждый пользователь программы должен относиться к какой-нибудь группе пользователей (роли), членам которой доступны различные функциональные возможности программы.
3. Программа должна принимать от пользователя некоторые данные и, возможно, после некоторой обработки, отображать их. При этом должна осуществляться защита от возможных атак на приложение. При разработке защиты нужно предположить, что приложение работает с базой данных, в которой сохраняет введенные пользователем данные.

II. Реализовать приложение-инсталлятор, позволяющее установить на компьютер пользователя приложение, реализованное в предыдущем пункте задания. Требования к приложению:

1. Приложение-инсталлятор совместно с устанавливаемым приложением должно обеспечивать защиту программного продукта от несанкционированного тиражирования.
2. Приложение-инсталлятор должно иметь защиту от возможных атак на него.

III. Протестировать правильность работы разработанных приложений.

3. Разработанное программное средство

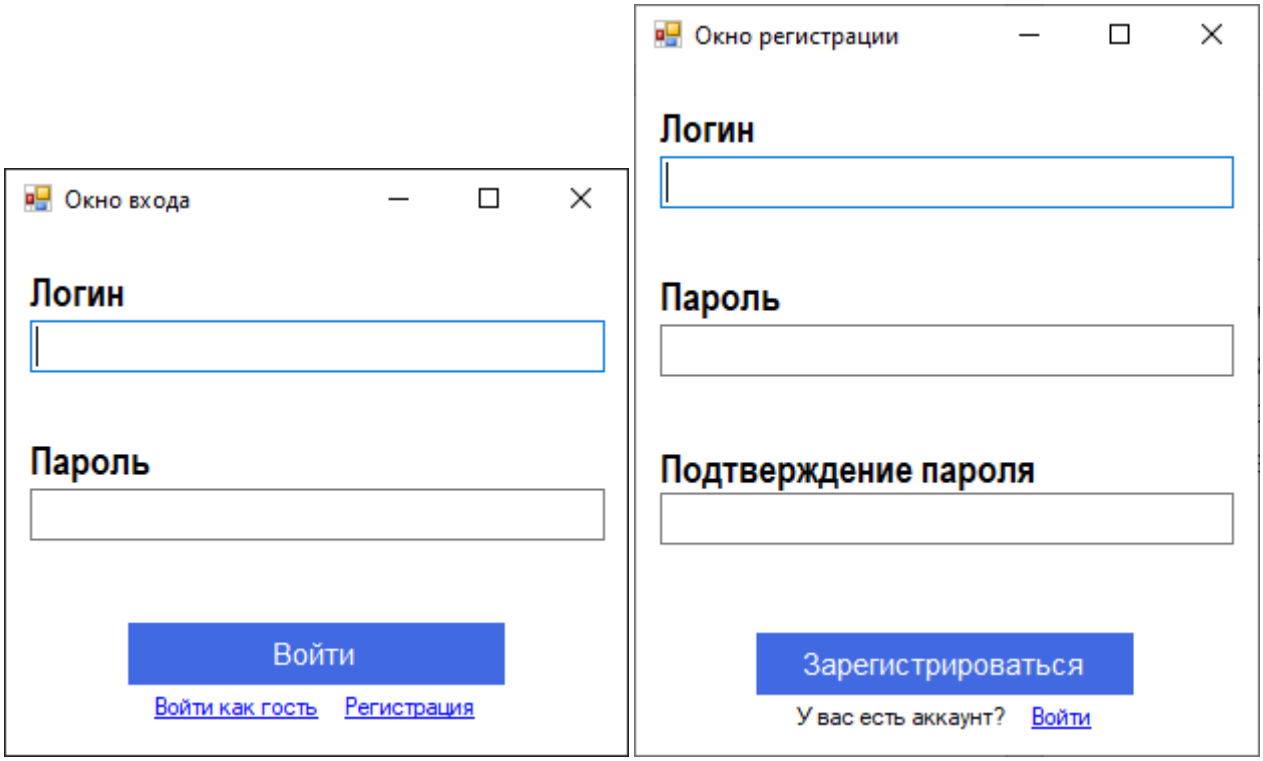
Для демонстрации концепции ролевого доступа было разработано приложение, позволяющее просматривать каталог фильмов, добавлять их в избранное и др. Всего в программе доступно 4 группы пользователей (роли):

Группа пользователей	**Возможности**
Гость	Просмотр каталога фильмов
Обычный пользователь	Просмотр каталога фильмов, добавление до 3 фильмов в избранное
Премиум пользователь	Просмотр каталога фильмов, добавление до 6 фильмов в избранное

Администратор	Просмотр каталога фильмов, добавление любого числа фильмов в избранное, модификация каталога фильмов, управление ключами активации премиум аккаунта

Разработанное программное средство представляет собой приложение Windows Forms, также используется база данных для хранения данных пользователей и фильмов.

Окно входа и регистрации:



При входе в аккаунт требуется ввести логин и пароль. Введенные данные проверяются следующим образом:

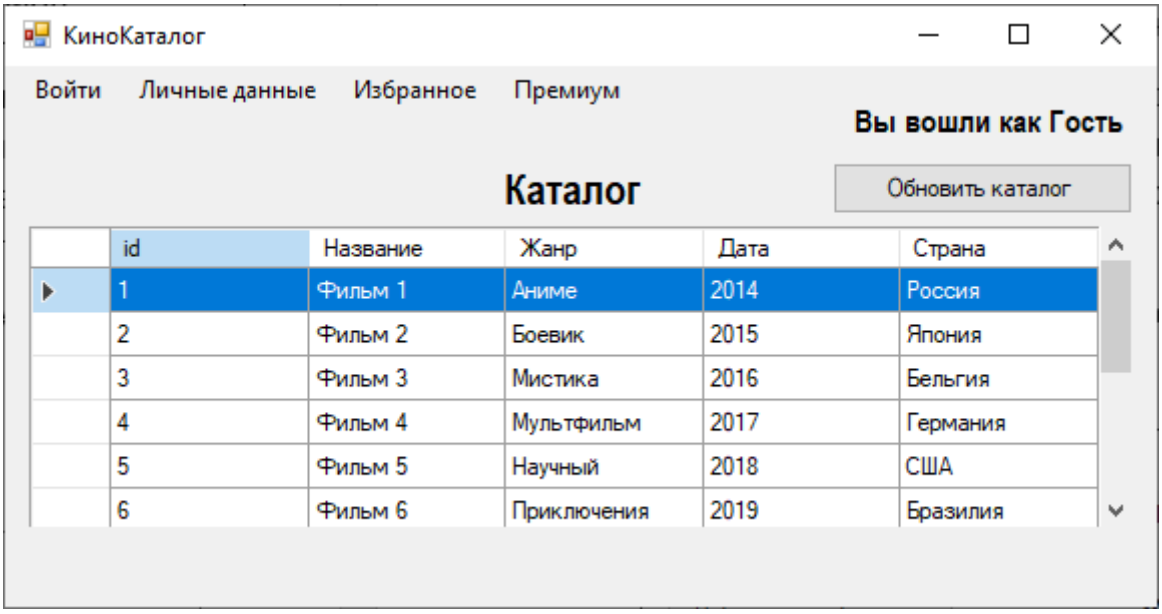
1) Введенный логин находится в базе данных и в случае успеха происходит получение всей остальное пользовательской информации: хешированный пароль, 16-байтовая соль, роль и ID пользователя.
2) Введенный пароль хешируется алгоритмом RIPEMD-160 с использованием полученной соли и сравнивается с захешированным паролем из базы данных. Если пароли совпадают, то вход успешен и открывается главное меню.

В случае, если аккаунта нет, то можно совершить регистрацию. Требуется ввести логин, пароль и подтверждение пароля. Если все сделано правильно (например, логин уникален и пароли совпадают), то соответствующие данные сохраняются в базе данных и пользователь перейдет в окно входа, где после повторного ввода сможет перейти в главное меню.

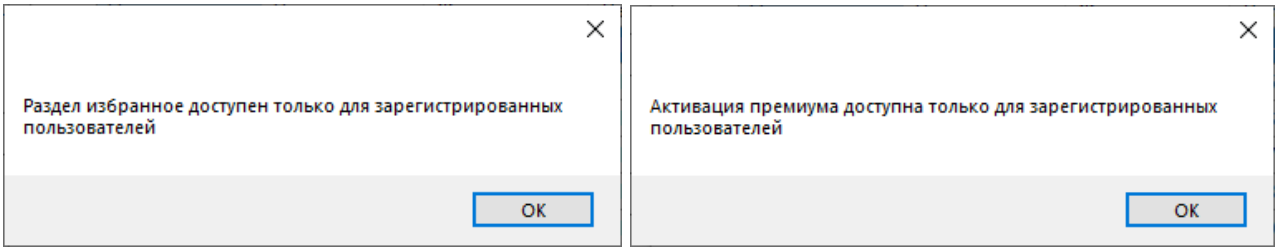
Следует подчеркнуть, что пароль будет хеширован с помощью алгоритма RIPEMD-160 с использованием случайно сгенерированной 16-байтовой соли (она тоже сохраняется в базу данных), а роль будет выбрана автоматически – обычный пользователь.

Опция “Войти как гость” позволяет перейти в главное меню без входа в аккаунт.

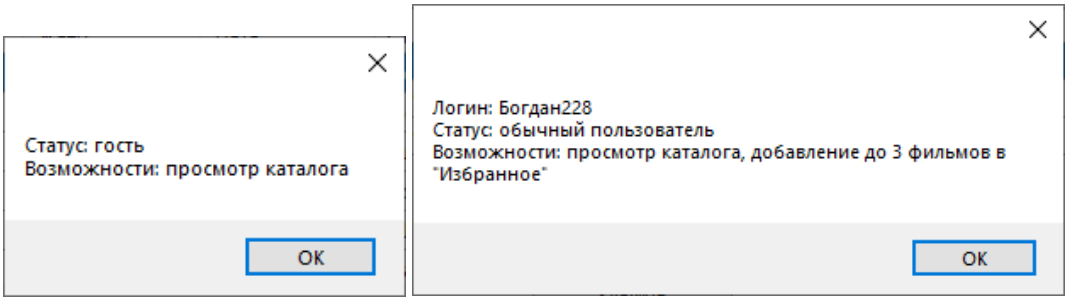
Интерфейс главного меню (гость):

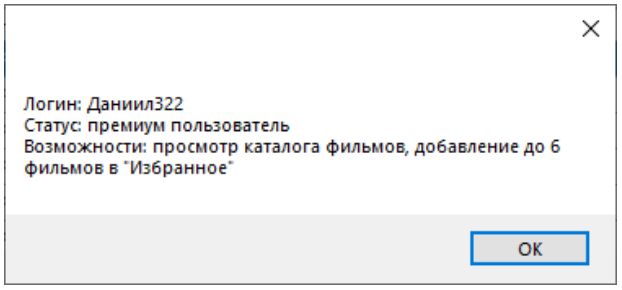


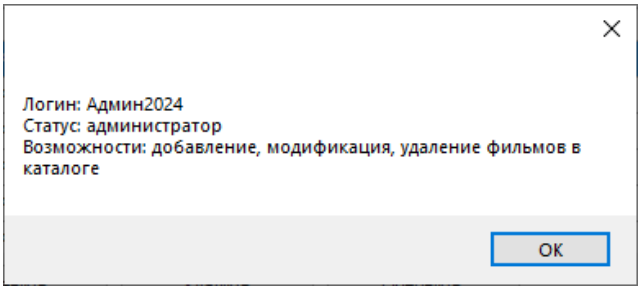
Гость может только просматривать каталог фильмов. Кнопки меню “Избранное” и “Премиум” ему недоступны, а в случае нажатие появляются пояснительные окна:



Кнопка меню “Личные данные” у гостя, обычного пользователя, премиум пользователя и администратора соответственно:

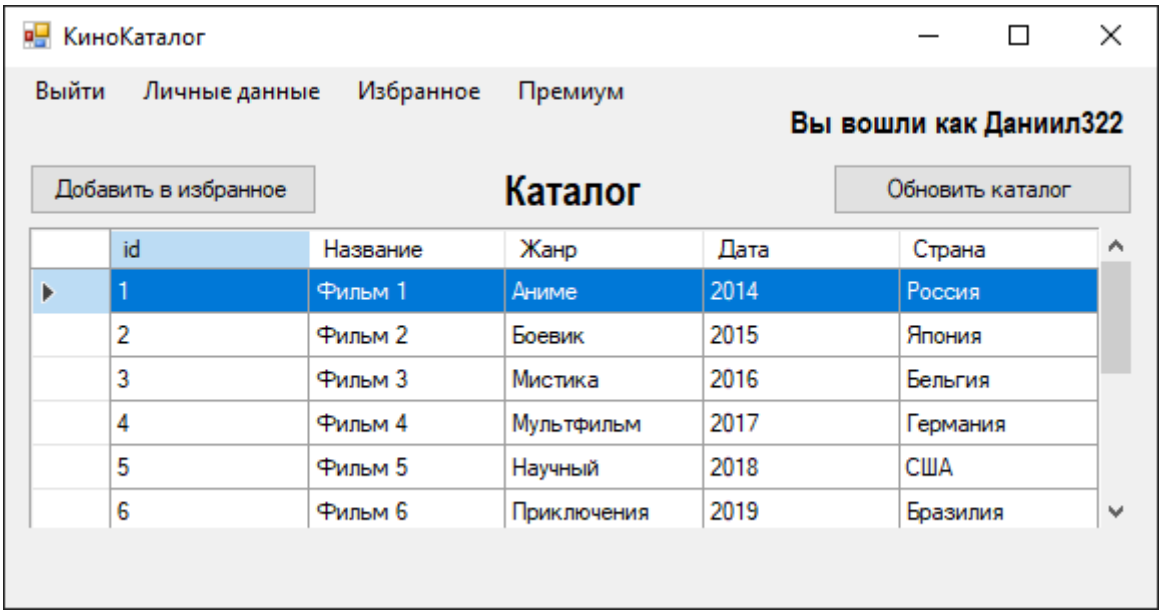






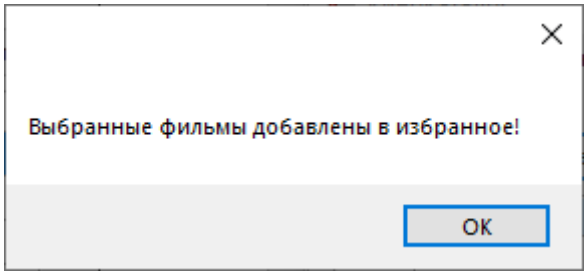
Эти окна показывают основную информацию о пользователях и их возможностях.

Интерфейс главного меню (обычный и премиум пользователь):

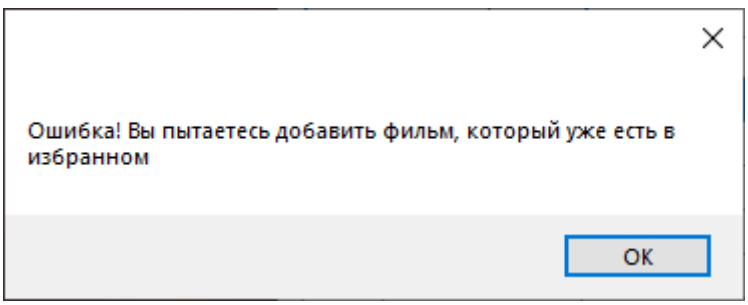


В отличие от гостя, у зарегистрированных пользователей появляется возможность не только просматривать каталог, но и добавлять фильмы в избранное. Можно добавлять сразу несколько фильмов за раз.

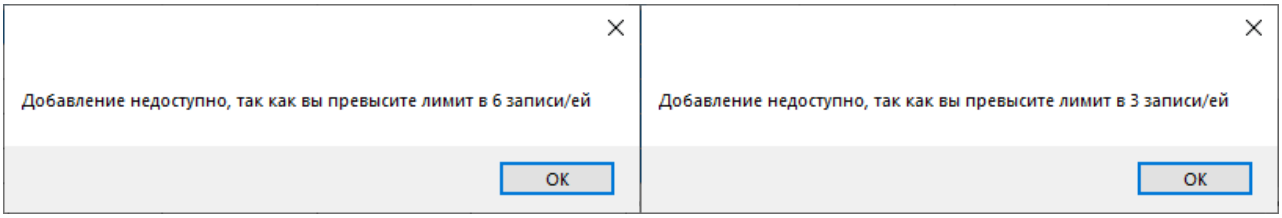
Успешное добавление фильма:



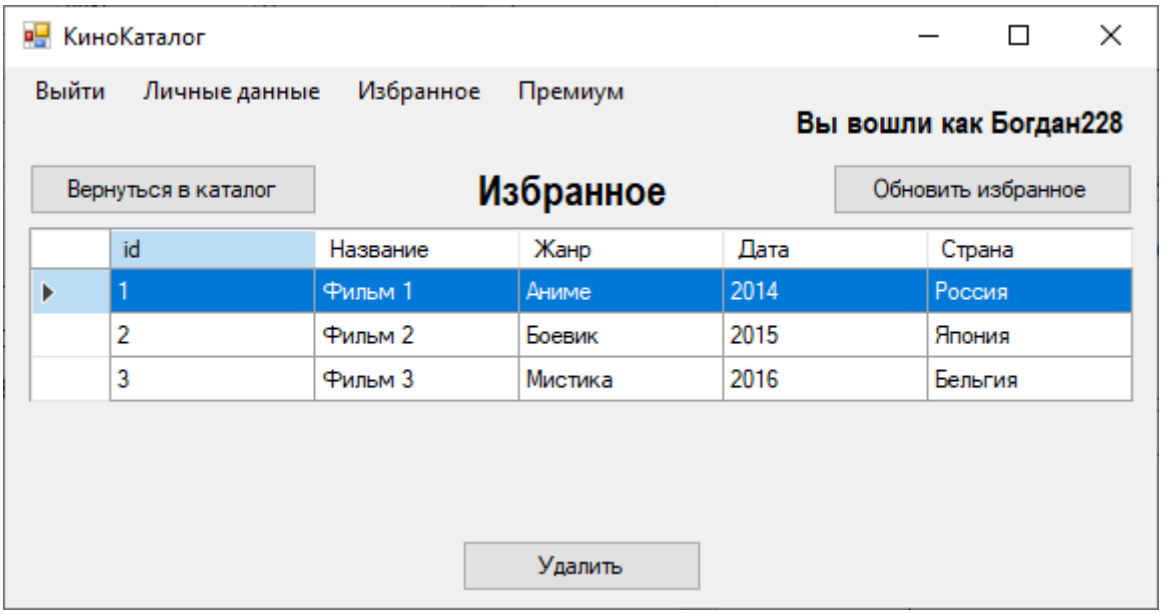
Попытка добавить фильм, который уже есть в избранном:



Максимальное число записей в избранном определяется ролью пользователя (премиум пользователь – 6 записей, обычный пользователь – 3 записи, администратор - безлимит):

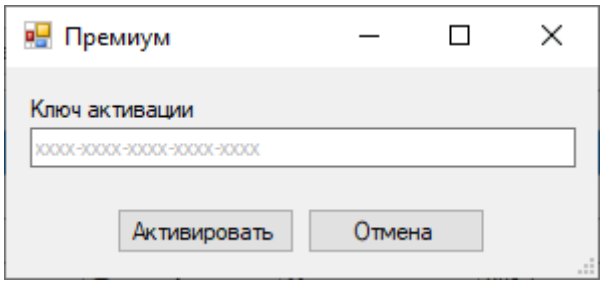


Интерфейс окна “Избранное”:

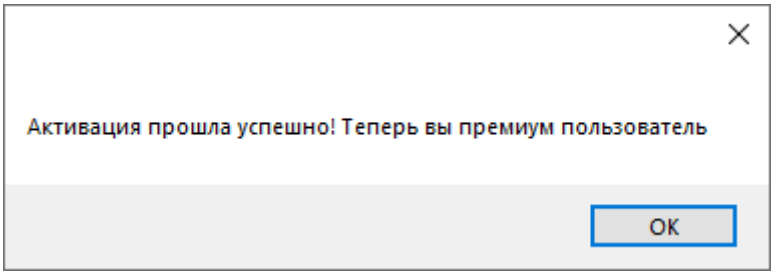


Мы можем просматривать сохраненные фильмы, удалять ненужные записи и возвращаться в основной каталог.

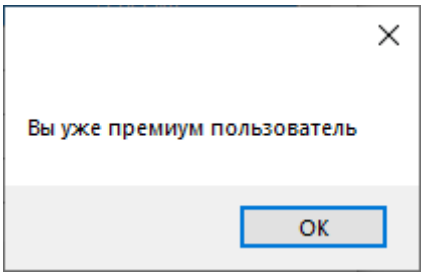
Кнопка меню “Премиум” открывает окно с активацией премиум аккаунта:



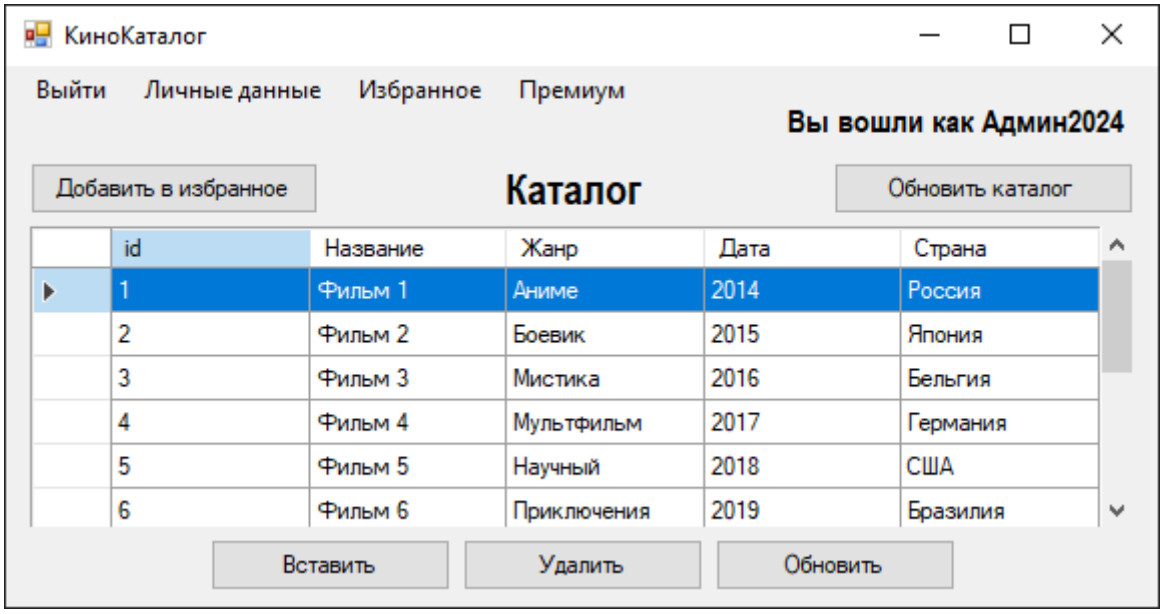
Ключи генерируется администратором (процесс будет рассмотрен ниже) и хранятся в базе данных. В случае ввода корректного ключа, аккаунт пользователя становится премиальным, а сам пользователь сможет добавлять больше фильмов в избранное.



Если премиум пользователь попробует активировать премиум аккаунт:

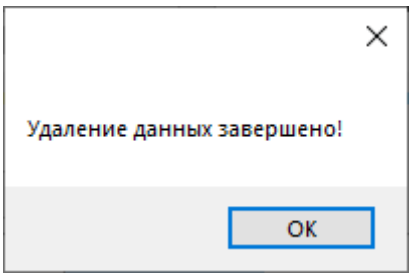


Теперь рассмотрим возможности администратора. Главное меню:

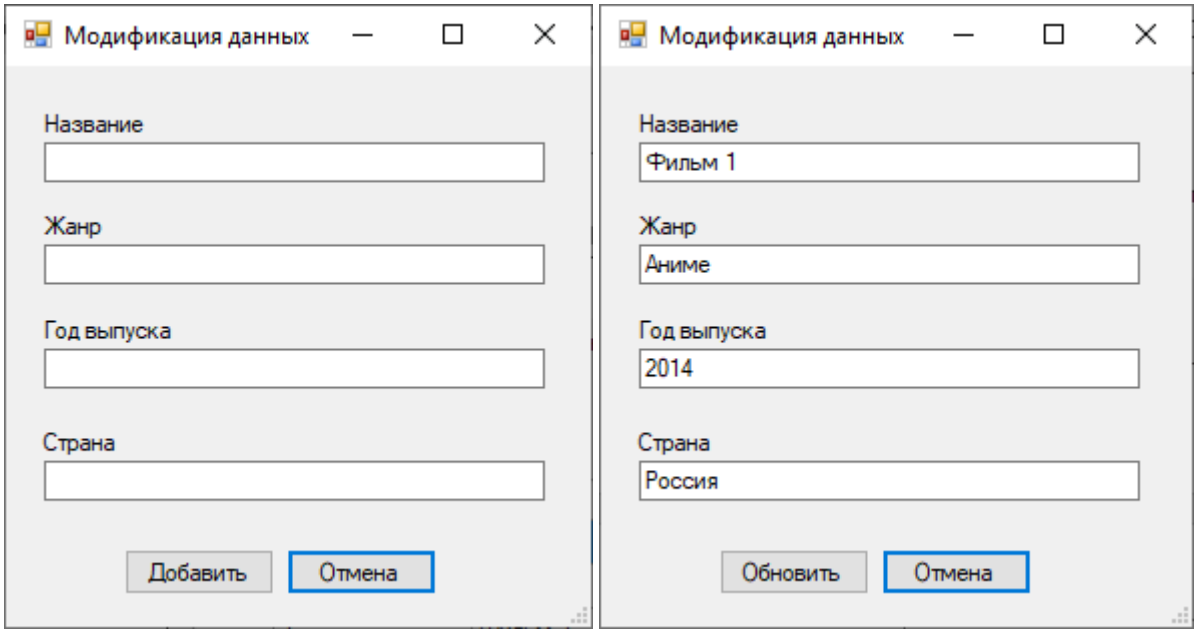


В отличие от остальных групп пользователей, администратор может модифицировать каталог (вставлять новые записи, удалять или обновлять существующие).

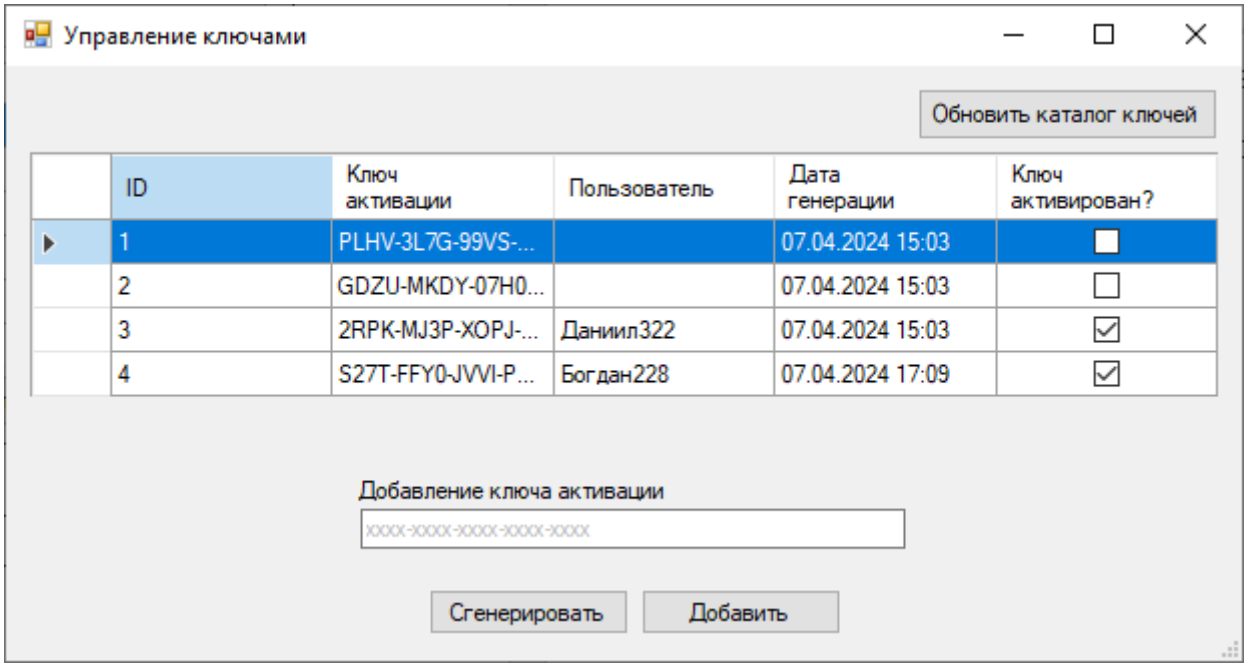
Удаление записей сопровождается сообщением:



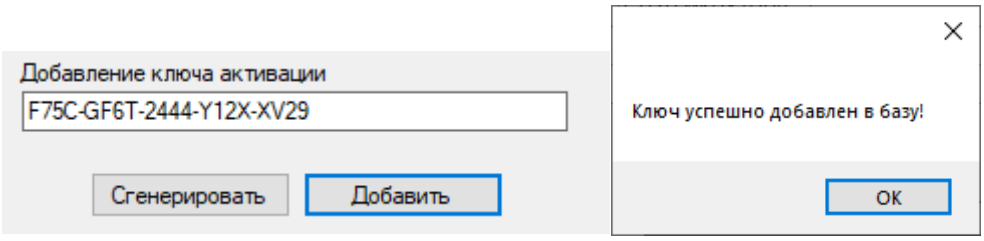
Окна вставки и обновления записей:



Кнопка меню “Премиум” открывает администратору окно с управлением ключами активации:



В этом окне администратор может просматривать список как свободных, так и уже активированных ключей с соответствующими владельцами. Также он может генерировать новые ключи. Вся информация хранится в базе данных. Сгенерируем и добавим новый ключ активации:



Наблюдаем его наличие в базе данных:

	ID	Ключ активации	Пользователь	Дата генерации	Ключ активирован?
▶	1	PLHV-3L7G-99VS-...		07.04.2024 15:03	☐
	2	GDZU-MKDY-07H0...		07.04.2024 15:03	☐
	3	2RPK-MJ3P-XOPJ-...	Даниил322	07.04.2024 15:03	☑
	4	S27T-FFY0-JVVI-P...	Богдан228	07.04.2024 17:09	☑
	5	F75C-GF6T-2444-Y...		07.04.2024 21:32	☐

4. Описание защиты от возможных атак на приложение

Атака «переполнение буфера»

Атака «переполнение буфера» — это тип атаки на компьютерные системы, при которой злоумышленник использует ошибку переполнения буфера для выполнения произвольного кода на целевой системе.

Для защиты от такого рода атак мы тщательно контролируем длину вводимых данных в каждом текстовом поле приложения. Например, максимальная длина логина – 30 символов, пароля – 64 символа. Ключ активации имеет строгий формат “xxxx-xxxx-xxxx-xxxx-xxxx”, то есть ограничен длиной 24 символа.

Атака «SQL-инъекции»

SQL-инъекция происходит, когда злоумышленник вводит вредоносные SQL-команды в пользовательский ввод, который затем исполняется базой данных. Это может привести к несанкционированному доступу к данным, их изменению или удалению.

Для предотвращения SQL-инъекций мы использовали параметризованные SQL-запросы, потому что они разделяют SQL-код от данных. Вместо того чтобы вставлять пользовательский ввод напрямую в SQL-код, параметризованный запрос использует параметры для представления этих данных. Затем эти параметры заменяются на пользовательский ввод перед выполнением запроса.

Приведем пример функции, которая выполняет получение фильмов из избранного пользователя с идентификатором userId:

```
public DataTable GetFavoriteMovies(int userId)
{
   DataTable dt = new DataTable();

   using (var cmd = new NpgsqlCommand(@"SELECT m.*
                              FROM pmib0706.Movies m
                              JOIN pmib0706.Favorites f ON m.ID = f.MovieID
                              WHERE f.UserID = @userId", connection))
   {
```

```
            cmd.Parameters.AddWithValue("userId", userId);

            using (NpgsqlDataAdapter da = new NpgsqlDataAdapter(cmd))
            {
                da.Fill(dt);
            }

        }

        return dt;
    }
```

В этом примере @userId - это параметр, который заменяется на значение переменной userId. Поскольку значение userId никогда не вставляется напрямую в SQL-код, нет возможности для SQL-инъекции.

Минимизация привилегий

Принцип минимальных привилегий — это основная идея в информационной безопасности, согласно которой доступ к ресурсам в системе должен быть организован таким образом, чтобы любая сущность внутри этой системы имела доступ только к тем ресурсам, которые минимально необходимы для успешного выполнения рабочей цели этой сущности, — и ни к каким другим.

Если не следовать этому принципу, то может возникнуть угроза безопасности, связанная с выполнением запросов к базе данных. Например, злоумышленник может удалить базу данных или отдельную таблицу.

Для решение этой проблемы в приложении подключение к базе данных происходит не под учетной записью администратора.

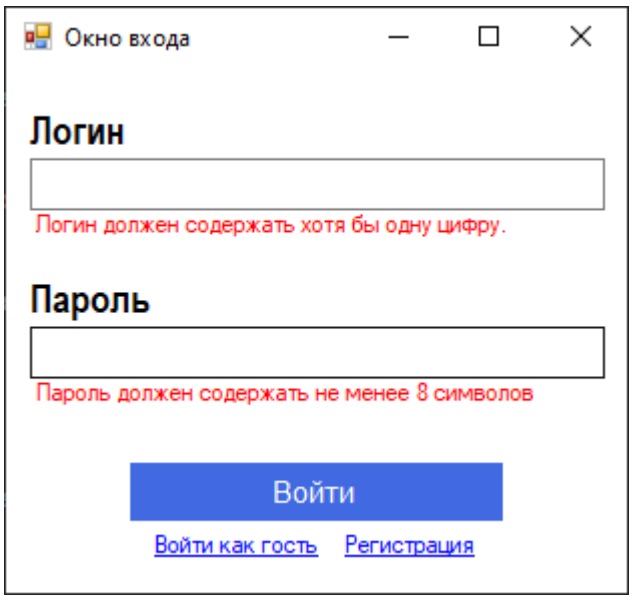
Проверка входных данных

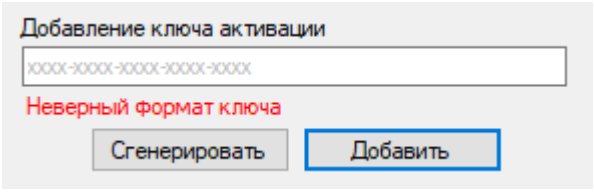
При проверке входных данных следует пропускать только корректные данные, а всё остальное – отбрасывать. Этому принципу мы следовали при разработке приложения. Каждое текстовое поле приложения (перед использованием его содержимого, например, для модификации записей таблицы базы данных) проверяется на корректность ввода (валидация):

- отсутствие пустоты или превышения максимальной длины;
- обязательное наличие определенных символов (для безопасности учетных записей);
- соответствие определенному формату (в случае с ключом активации);

- запрет на ввод символов, наличие которых в соответствующих текстовых полях невозможно (например, текстовое поле, которое соответствует году выпуска фильма, разрешает только циферную запись).

Дополнительно для каждого текстового поля предусмотрены подсказки, которые опишут проблему в случае некорректного ввода. Примеры таких подсказок:



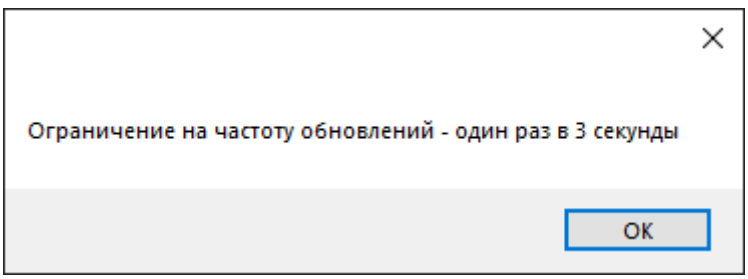


Отказ в обслуживании

В нашем приложении есть несколько кнопок, которые обновляют следующие таблицы: каталог фильмов, избранное или каталог ключей. А обновление осуществляется путем обращения к соответствующей таблице в базе данных.

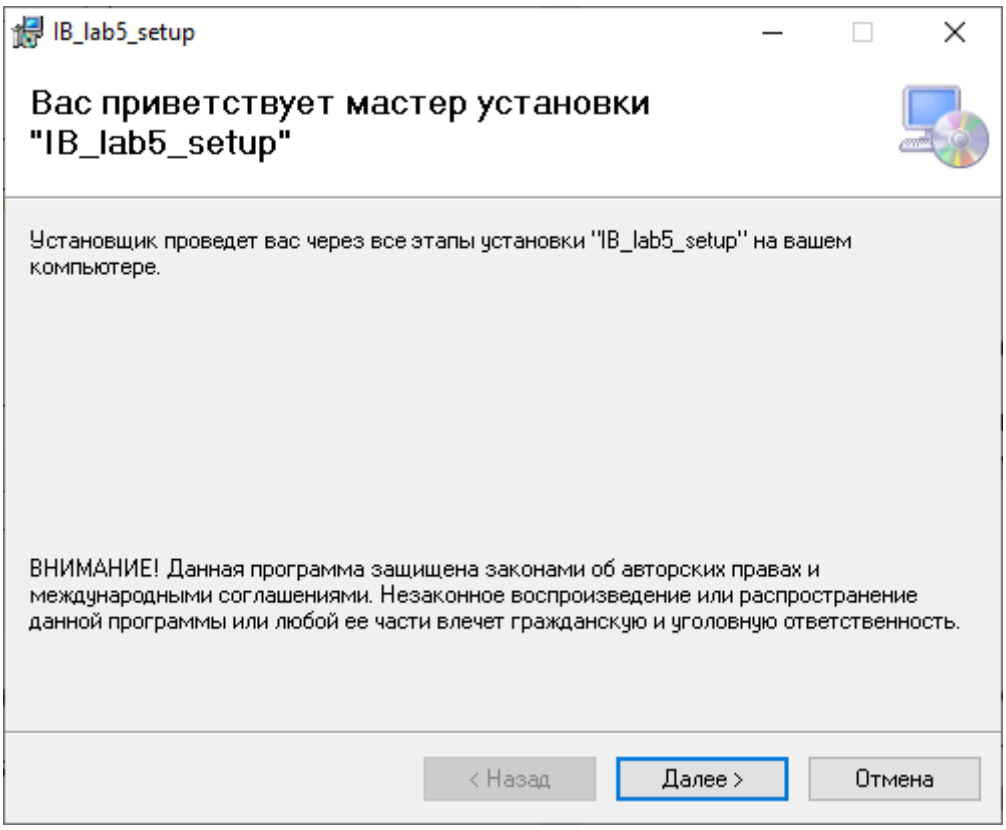
Если пользователь будет очень часто и быстро нажимать на любую из рассмотренных кнопок, это может привести к излишней нагрузке на базу данных и ее замедлению в лучшем случае, а в худшем - к ее отказу, если нагрузка слишком велика. Эта атака называется **DoS-атакой** или атакой отказа в обслуживании.

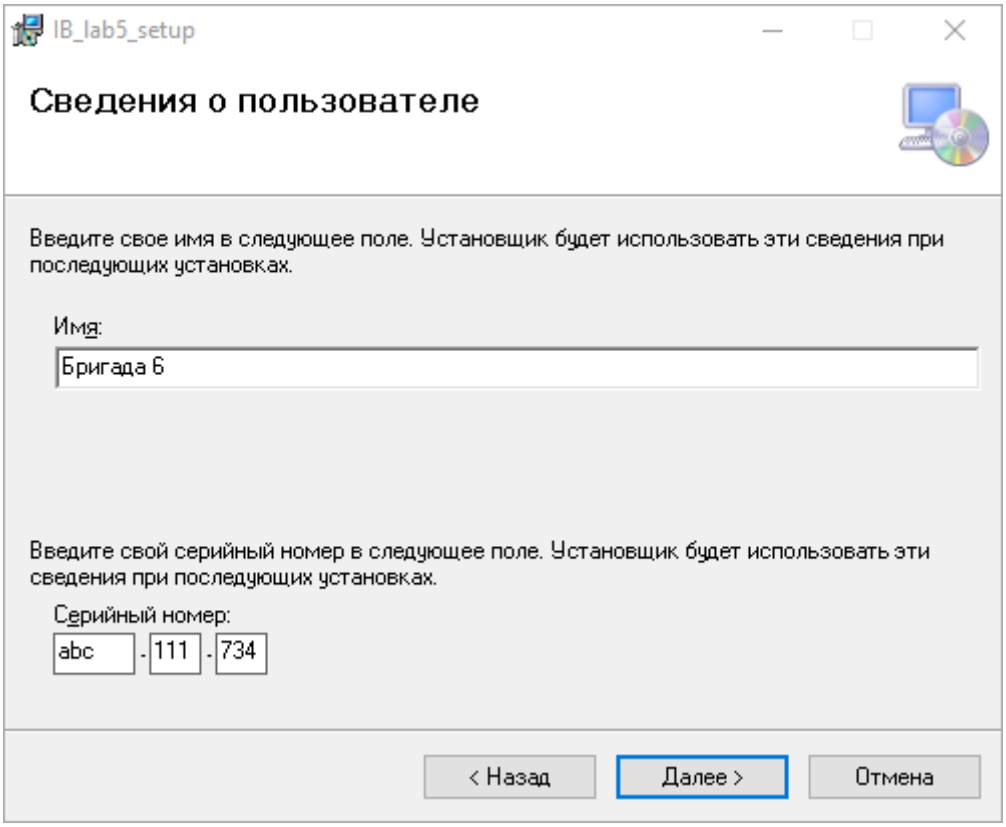
Чтобы избежать этой атаки, мы ввели ограничение на частоту обновлений, разрешив обновление только один раз в 3 секунды. В случае попытки частого обновления пользователю будет отказано в нем, и программа выдаст предупреждение:



5. Обзор приложения-инсталлятора

Был разработан установщик с проверкой серийного номера с помощью расширения Microsoft Visual Studio Installer Projects. Рассмотрим процесс установки:

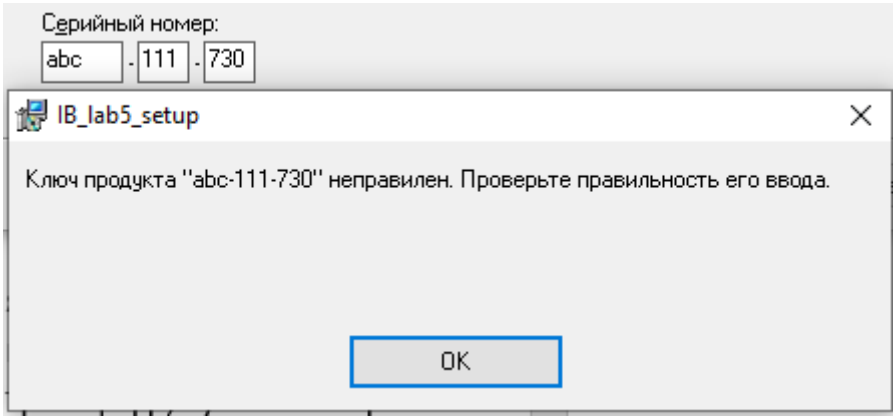




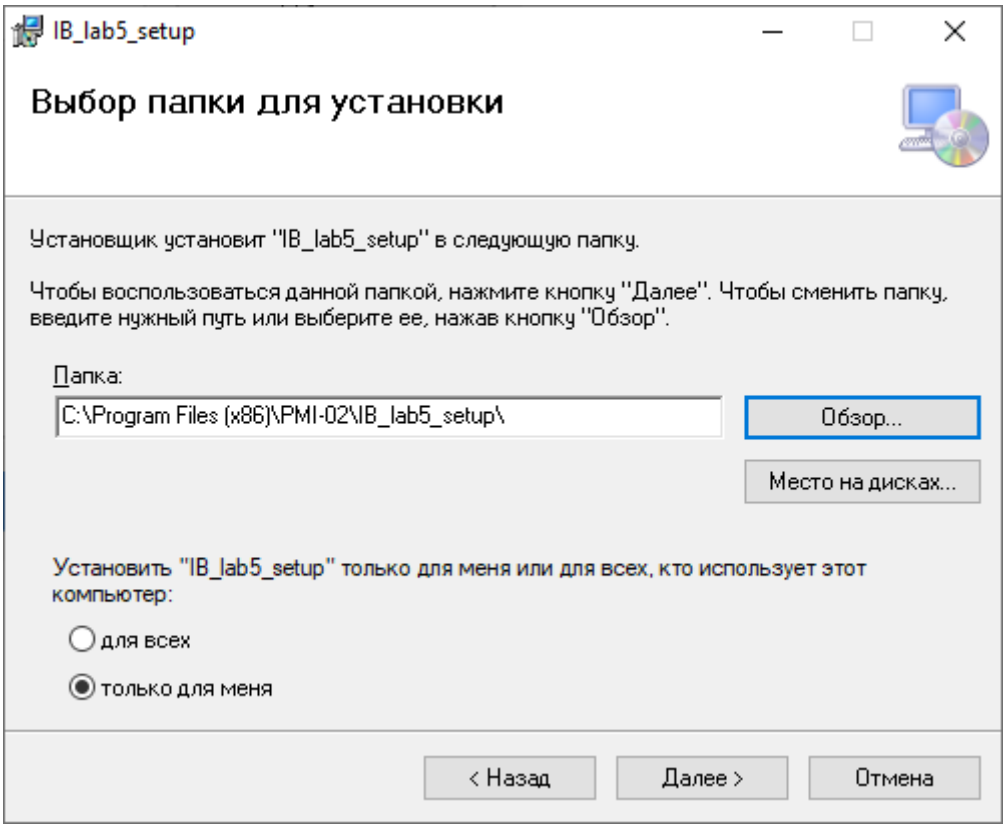
Серийный номер - это уникальный идентификатор, который используется производителями программного обеспечения для контроля над

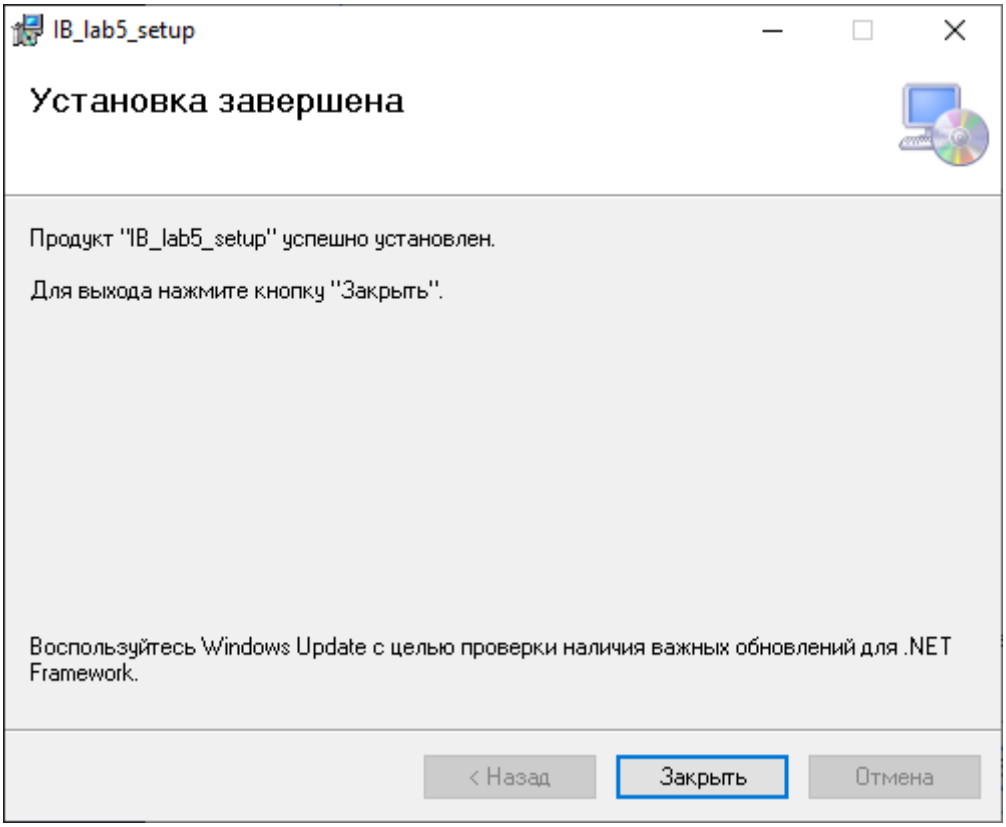
распространением своих продуктов. Он требуется при установке программного обеспечения для подтверждения, что пользователь приобрел лицензию на использование программы.

Мы использовали формат, требующий от пользователя ввода серийного номера, состоящего из трех букв (латинских строчных или прописных), трех цифр (простая проверка, что введены 3 цифры) и еще трех цифр, сумма которых делится на 7 без остатка (проверка с помощью алгоритма). Пример некорректного ввода серийного номера:



Дальнейшая установка:





Деинсталляция выполняется повторным запуском установщика и выбором соответствующего маркера.