**Министерство образования и науки Нижегородской области**

**Государственное бюджетное профессиональное образовательное учреждение «Нижегородский Губернский колледж»**

Допустить к защите:

преподаватели

\_\_\_\_\_\_\_\_\_\_ Е.П. Голубева

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2025 г.

**ОТЧЁТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

**модуль: ПМ. 01. Разработка модулей программного обеспечения для компьютерных систем**

|  |  |
| --- | --- |
| Руководитель\_\_\_\_\_\_\_\_\_Голубева Е.П.  Циркова В.В.  Полетаева М.В. | 01.03.2025 г. |
| Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кузьмин И.В. | 01.03.2025 г. |

Специальность, группа: 09.02.07, 41П

Нижний Новгород

2025 г.

# 

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc192180368)

[1. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 1 (01.03.2025 г.) - СОЗДАНИЕ БИБЛИОТЕКИ DLL И ЕЕ ТЕСТИРОВАНИЕ ПОСРЕДСТВОМ ЮНИТ-ТЕСТОВ 5](#_Toc192180369)

[2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 2 (03.03.2025 г.) - СОЗДАНИЕ И ЗАПОЛНЕНИЕ ТАБЛИЦ С ДАННЫМИ, БАЗЫ ДАННЫХ И ER-ДИАГРАММЫ 7](#_Toc192180370)

[3. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 3 (04.03.2025 г.) - СОЗДАНИЕ DATA DICTIONARY, ДИАГРАММЫ ПРЕЦЕДЕНТОВ 14](#_Toc192180371)

[4. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 4 (05.03.2025 г.) - РАЗРАБОТКА ДЕСКТОПНОГО ПРИЛОЖЕНИЯ «СИСТЕМА УПРАВЛЕНИЯ КОНФЕРЕНЦИЯМИ» 20](#_Toc192180372)

[5. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 5 (06.03.2025 г.) - СОСТАВЛЕНИЕ ФАЙЛА О РУЧНОМ ТЕСТИРОВАНИИ 26](#_Toc192180373)

[6. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 6 (07.03.2025 г.) - СОСТАВЛЕНИЕ ИТОГОВЫХ ДОКУМЕНТОВ ПО БЛОКУ УЧЕБНОЙ ПРАКТИКИ, ПРОВЕРКА ЧУЖОЙ РАБОТЫ 31](#_Toc192180374)

[ЗАКЛЮЧЕНИЕ 32](#_Toc192180375)

# ВВЕДЕНИЕ

Учебная практика - составная часть образовательного процесса в системе образования, как на ступени СПО, так и на ступени высшего образования, направленная на комплексное освоение видов профессиональной деятельности, предусмотренной в рамках изучаемой студентом специальности, формирование общих и профессиональных компетенций, а также приобретение необходимых практических навыков, умений и опыта практической работы, в целях подготовки компетентных, квалифицированных, легко адаптирующихся и конкурентоспособных на рынке труда специалистов, в том числе, готовых переориентироваться, в случае необходимости, в рамках укрупненных групп направлений подготовки.

Основная цель учебной практики – закрепление и углубление знаний, приобретенных в процессе лекционных, лабораторных и практический занятий и самостоятельной работы студента, необходимых профессиональных качеств будущего специалиста в избранном будущем профессиональном направлении деятельности, а также получение фундаментальных компетенций.

В задачи учебной практики входят:

* реализация на практике теоретических познаний;
* формирование более детального представления о будущей профессии;
* приобретение практических знаний и опыта работы по выбранной специальности;
* выработка компетенций самостоятельного формирования первичных данных для отражения результатов прохождения практики;
* формирование навыков критического и творческого анализа вторичных данных и литературных источников по теме практики;
* отработка и закрепление методов, приемов и способов разработки программного обеспечения;
* овладевание принципами технологии структурного и объектноориентированного программирования;
* отработка практических навыков и способов оптимизации и приемов рефакторинга;
* освоение основных принципов отладки и тестирования программных продуктов.
* сбор и обработка материалов, необходимых для составления отчета по практике.

Этапы проходимой учебной практики по модулю ПМ 01. «Разработкамодулей программного обеспечения для компьютерных систем» включают в себя подготовительный этап, ознакомление с индивидуальными заданиями (программой) учебной практики и графиком ее прохождения, основной этап (выполнение заданий в соответстви с графиком прохождения практики) и заключительный этап – формирование итогового отчета и мультимедийной презентации для защиты результатов и итогов учебной практики.

# ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 1 (01.03.2025 г.) - СОЗДАНИЕ БИБЛИОТЕКИ DLL И ЕЕ ТЕСТИРОВАНИЕ ПОСРЕДСТВОМ ЮНИТ-ТЕСТОВ

**Цель занятия:** познакомиться и овладеть практическими навыками работы с написанием DLL-библиотек и создании юнит тестов для ее тестирования**.**

В начале практической работы, входящей в комплекс заданий, учебной практики по профессиональному модулю «ПМ. 01. Разработка модулей программного обеспечения для компьютерных систем» необходимо было написать библиотеку классов RegMarkLib.dll на языке C# которая, была бы предназначена для работы с регистрационными знаками транспортных средств Российской Федерации. В функциональную сферу данной библиотеки была бы предусмотрена возможность проверки корректности регистрационных номеров, генерации следующего номера в последовательности, определение следующего номера в заданном диапазоне, вычисление количества возможных комбинаций номеров.

Иными словами, в ее функционал должно входить:

1) проверка корректности регистрационного знака

2) генерация следующего регистрационного знака

3) генерация следующего регистрационного знака в заданном диапазоне

4) подсчет возможных комбинаций номеров в диапазоне

По результатам написания библиотеки необходимо было написать 15 юнит тестов для проверки ее работоспособности, при этом 10 из данных юнит тестов простые и 5 сложные.

В ходе выполнения данного этапа учебной практики, были реализованы юнит тесты для проверки функции корректного распознания валидного регистрационного знака, корректного отклонения номера с недопустимыми символами, тестирование генерации следующего регистрационного знака, проверка, что метод корректно обрабатывает ситуацию, когда весь диапазон регистрационных знаков исчерпан, проверка правильности вычисления количества возможных регистрационных знаков в заданном диапазоне и другие.

После чего необходимо было запустить данные тесты для проверки работоспособности библиотеки (рисунок 1).

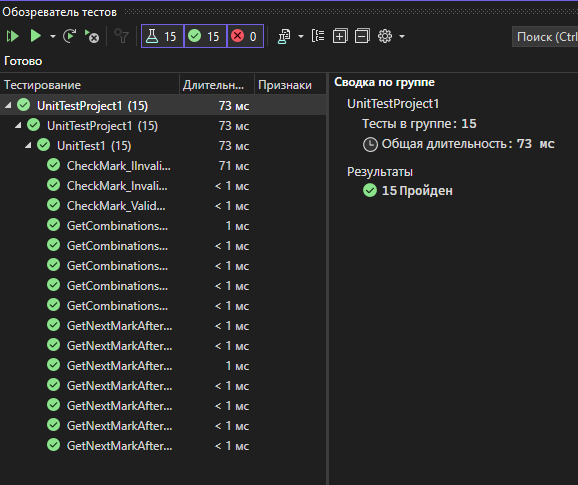


Рисунок 1 – Запуск юнит-тестов и результаты тестирования библиотеки DLL

**ВЫВОД**: В ходе выполнения данного блока заданий, было осуществлено создание библиотеки DLL и ее успешное тестирование юнит-тестами, играющими важную роль в тестировании библиотек DLL, в том числе, по причине того, что это неотъемлемая часть процесса разработки, обеспечивающая корректность, надежность, и качество предоставляемого функционала. Данные тесты помогают разработчикам быстро выявлять ошибки, предотвращать регрессии и повышать доверие к библиотеке со стороны пользователей.

# ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 2 (03.03.2025 г.) - СОЗДАНИЕ И ЗАПОЛНЕНИЕ ТАБЛИЦ С ДАННЫМИ, БАЗЫ ДАННЫХ И ER-ДИАГРАММЫ

**Цель занятия:** познакомиться и овладеть практическими навыками работы с созданием БД в DBeaver на языке PostgreSQL**.**

В продолжение практической работы, входящей в комплекс заданий, учебной практики по профессиональному модулю «ПМ. 01. Разработка модулей программного обеспечения для компьютерных систем» необходимо было реализовать БД на языке PostgreSQL в программе DBeaver.

В том числе, в ходе практического выполнения данного блока заданий, были созданы и заполнены данными из файлов Excel таблицы, содержащие в себе, первичные данные для создания базы данных (рисунки 2 - 13).

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2 - Таблица активности

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3 - Таблица «Города»

Изображение выглядит как текст, снимок экрана, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4 - Таблица «Участники»

Изображение выглядит как текст, снимок экрана, число, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 5 – Таблица «Страны»

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 6 – Таблица «Направления»

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 7 – Таблица «Пол»

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 8 – Таблица «Жюри»

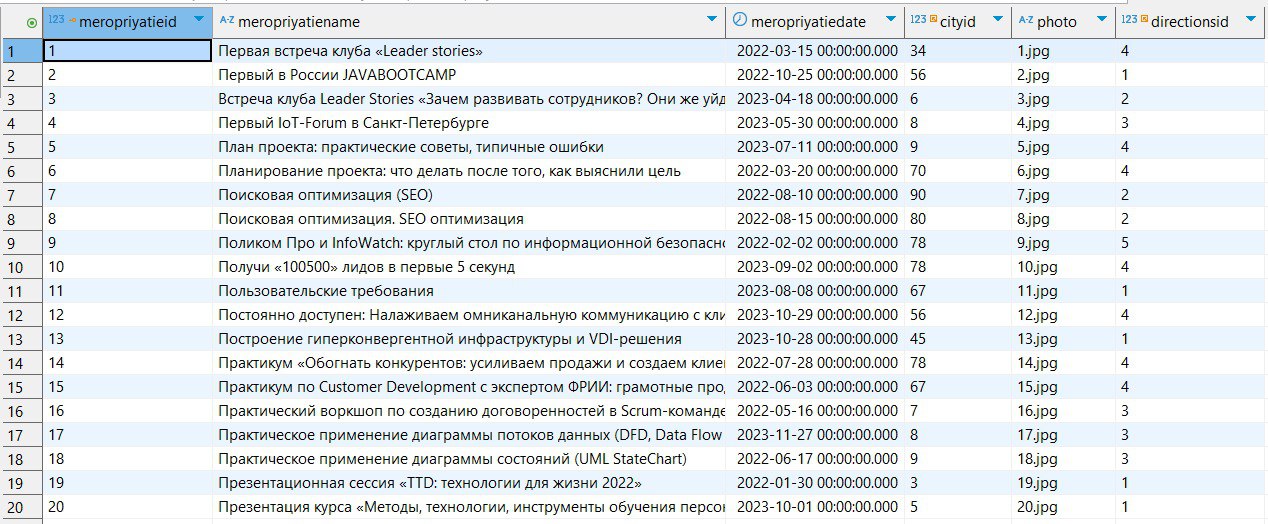


Рисунок 9 – Таблица «Мероприятия»

Изображение выглядит как текст, снимок экрана, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 10 – Таблица «Мероприятия и активности»

Изображение выглядит как текст, снимок экрана, меню, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 11 – Таблица «Модераторы»

Изображение выглядит как текст, Шрифт, число, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 12 – Таблица «Организаторы»

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 13 – Таблица «События»

Структура созданной в ходе выполнения задания Базы Данных «informationsecuritydb» представлена на рисунке 14. В дальнейшем, в ходе выполнения данного блока задания , была создана ER-диаграмма(диаграмма сущностей и связей) по данной БД, представляющая собой графическое представление структуры БД, которая отображает основные сущности и связи между ними. Данная диаграмма помогает визуализировать и понять структуру базы данных, выявить возможные проблемы в проектировании и правильно определить взаимосвязи между различными частями системы (рисунок 15).

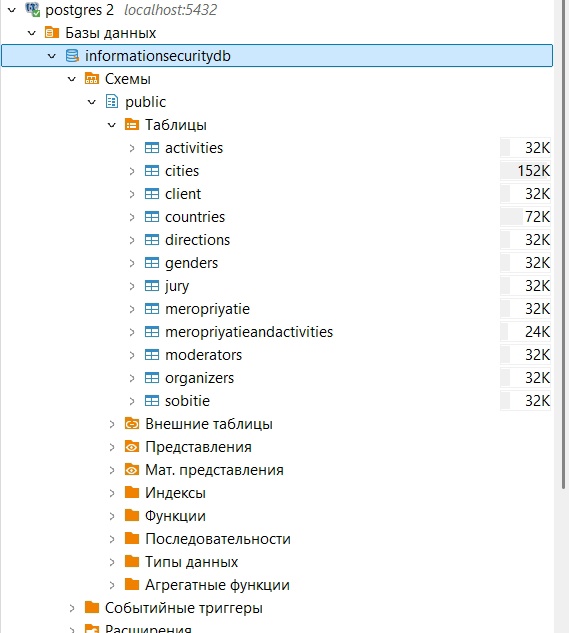


Рисунок 14 – Структура БД «informationsecuritydb»

**ВЫВОД**: В ходе выполнения данной части задания, как очередного этапа прохождения учебной практики, были созданы таблицы, содержащие массив исходных данных, необходимых для создания БД, сама база данных «informationsecuritydb» и ER-диаграмма, к созданной БД, визуализирующая ее структуру, в разрезе сущностей БД и связей между ними, используемая для планирования и проектирования базы данных и позволяет наглядно увидеть взаимосвязи между различными данными в системе.

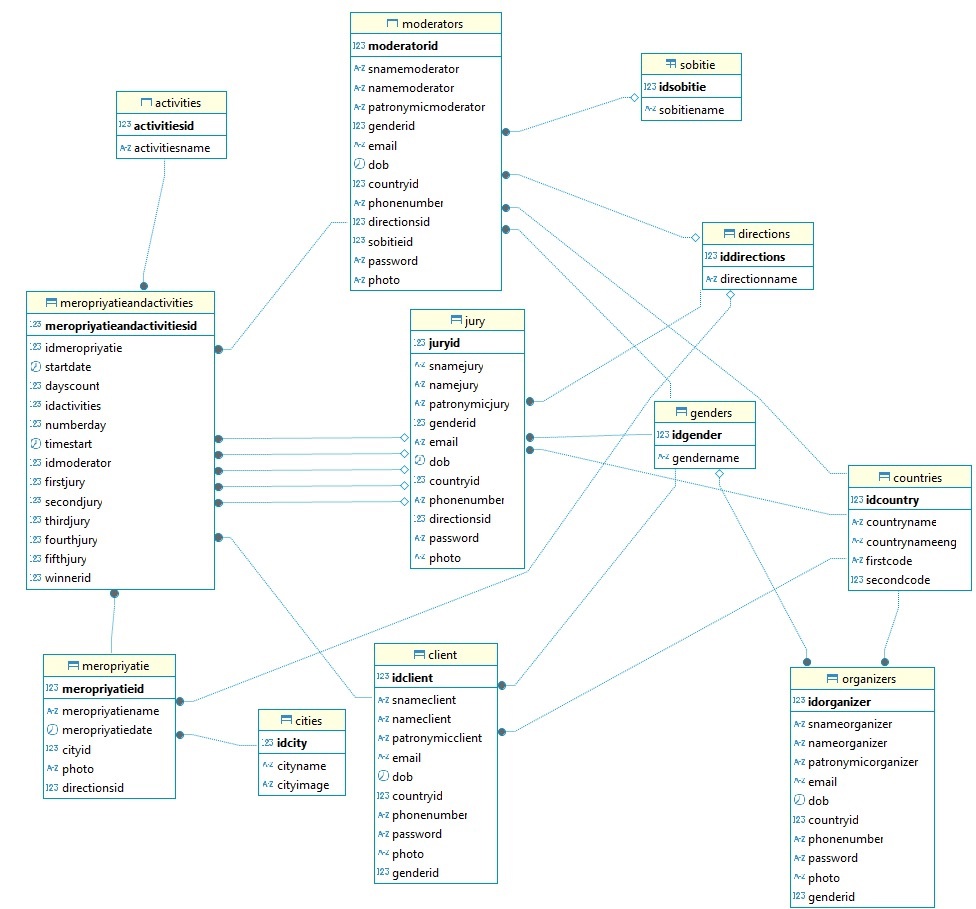


Рисунок 14 – ER-диаграмма БД

# ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 3 (04.03.2025 г.) - СОЗДАНИЕ DATA DICTIONARY, ДИАГРАММЫ ПРЕЦЕДЕНТОВ

**Цель занятия:** познакомиться и овладеть практическими навыками работы с создание словаря Data Dictionary, а также со спецификой построения диаграммы прецедентов**.**

На следующем этапе прохождения учебной практики необходимо было, в качестве последующего этапа создать Data Dictionary (словарь данных), представляющий собой документ, описывающий структуру базы данных, определяя ее таблицы, поля, типы данных, ограничения, связи и другие характеристики, имеющий важное значение в качестве инструмента для разработчиков, администраторов БД и аналитиков, помогающий управлять данными и обеспечивать их целостность. Иными словами, это ключевой инструмент для проектирования, администрирования и работы с базами данных, позволяющий стандартизировать структуру данных, улучшает качество работы с БД и облегчает понимание системы всеми участниками проекта (рисунки 15 – 26).

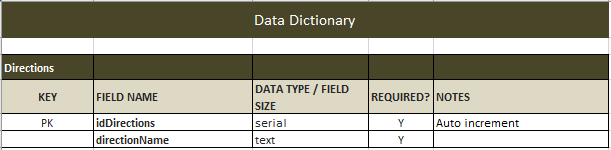


Рисунок 15 – Таблица «Directions»

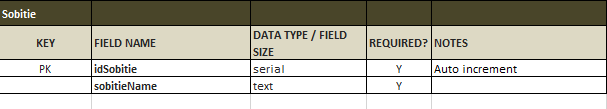


Рисунок 16 - Таблица «Sobitie»

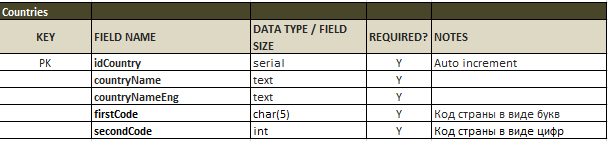


Рисунок 17 - Таблица «Countries»

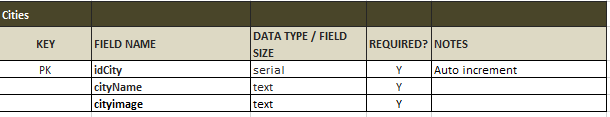


Рисунок 18 - Таблица «Cities»

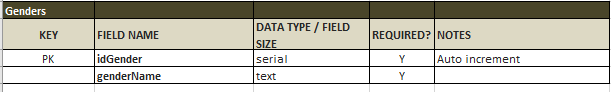


Рисунок 19 - Таблица «Genders»

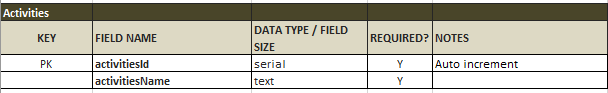


Рисунок 20 - Таблица «Activities»

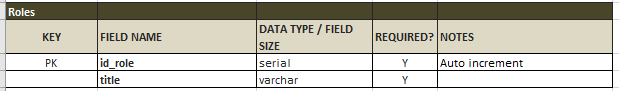


Рисунок 21 - Таблица «Roles»

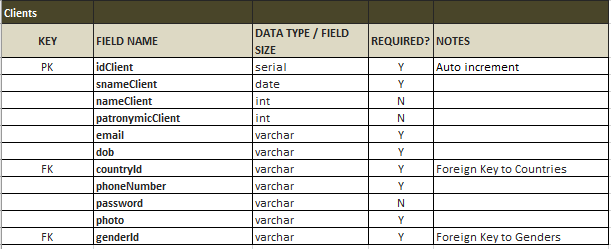


Рисунок 22 - Таблица «Clients»

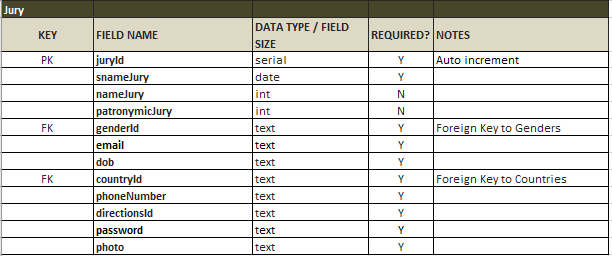
****

Рисунок 23 - Таблица «Jury»

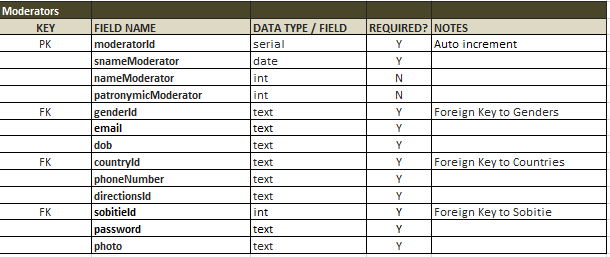


Рисунок 24 - Таблица «Moderators»

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 25 - Таблица «Meropriyatie»

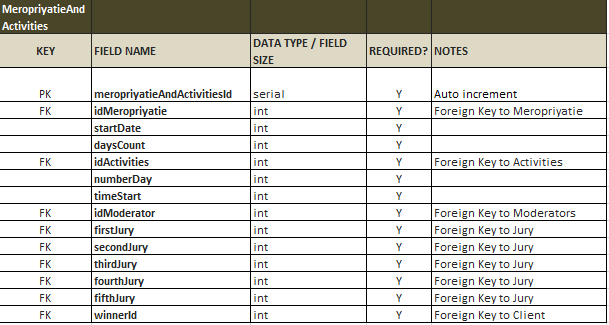


Рисунок 26 - Таблица «MeropriyatieAndActivities»

В ходе дальнейшего выполнения данного блока заданий по учебной практике, была составлена диаграмма прецедентов – один из видов диаграмм UML, который используется для моделирования функциональности системы с точки зрения взаимодействия пользователей (актеров) с системой.

В данном случае, эта диаграмма показывает, какие функции (прецеденты) доступны пользователям и какие роли (актеры) их выполняют. Диаграмма прецедентов позволяет четко очертить, что входит в функциональность системы, а что находится за её пределами, а также помогает избежать разработки ненужных функций, визуализируя взаимодействие пользователей с системой, наглядно демонстрируя, какие роли могут выполнять пользователи и помогая проектировщикам и разработчикам понять, какие сценарии нужно реализовать, исключая недопонимания на этапе проектирования. Именно диаграмма прецедентов визуализирует основные функции системы и показывает, кто и как с ними взаимодействует, она проста для понимания и полезна на ранних этапах проектирования, помогая определить ключевые сценарии работы приложения.

**ВЫВОД**: В ходе выполнения данного блока заданий по учебной практике, были составлены Data Dictionary и диаграмма прецедентов, имеющие стратегически важное значение. Data Dictionary обеспечивает структурированное описание всех данных в базе, помогая разработчикам и администраторам БД избежать ошибок при проектировании, упрощая сопровождение и масштабирование системы, в том числе, посредством гарантии единообразия данных и их корректное использование в системе. Диаграмма прецедентов позволяет определить, какие функции должна выполнять система и кто с ними взаимодействует, помогая выявить ключевые сценарии работы, которые требуют хранения данных, способствуя грамотному проектированию структуры БД, учитывая потребности пользователей. Использование Data Dictionary и диаграммы прецедентов вместе позволяет создать логичную, согласованную и удобную базу данных, которая полностью соответствует потребностям пользователей и бизнеса.

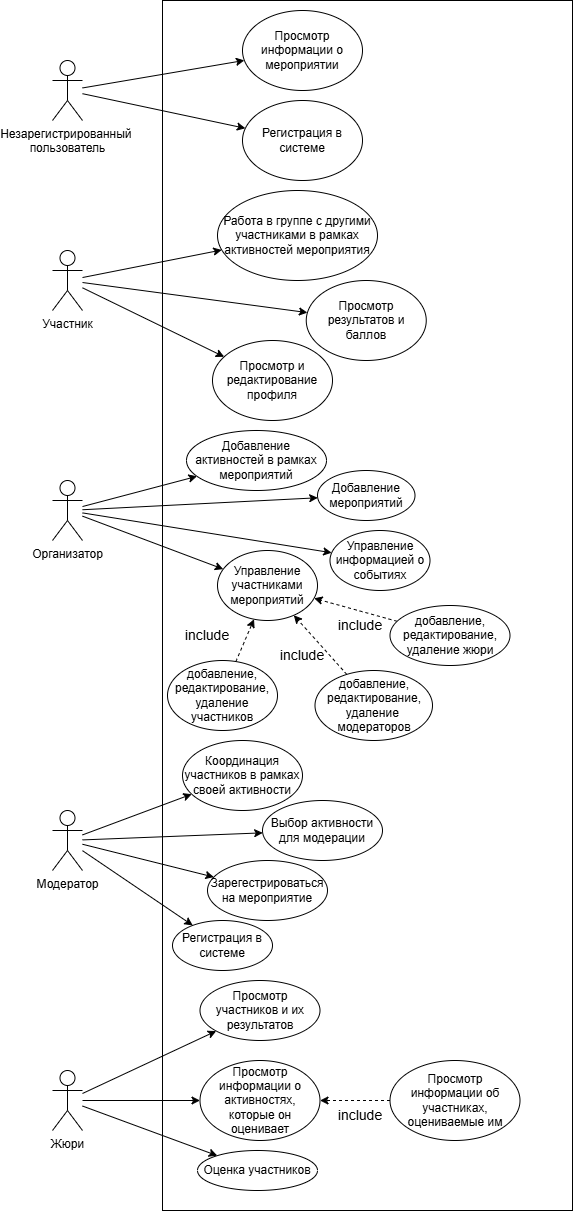


Рисунок 27 – Диаграмма прецедентов

# ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 4 (05.03.2025 г.) - РАЗРАБОТКА ДЕСКТОПНОГО ПРИЛОЖЕНИЯ «СИСТЕМА УПРАВЛЕНИЯ КОНФЕРЕНЦИЯМИ»

**Цель занятия:** Овладеть практическими навыками и умениями в разработке десктопных приложений с подключением к ним баз данных

На следующем этапе прохождения учебной практики необходимо было, в качестве последующего этапа создать десктопное приложение «Система управления конференциями» на Avalonia, созданное для организаторов конференций по информационной безопасности, представляющее собой программу, устанавливаемую и выполняемую на рабочей станции под управлением операционной системы, которое, в отличии от веб-приложений может работать автономно, без подключения к интернету, обеспечивая доступ к аппаратным ресурсам (например, файловой системе, принтерам, видеокартам) и могут работать с большими объемами данных.

В ходе создания десктопного приложения, к нему была подключена создаваемая в рамках прохождения учебной практики база данных (рис. 28).

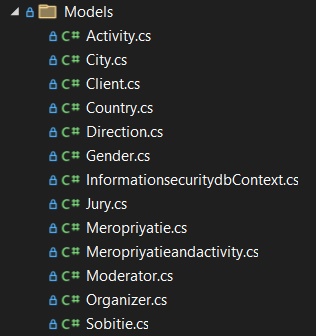


Рисунок 28 – Подключение базы данных к десктопному приложению

Разрабатываемое десктопное приложение функционирует таким образом, что при запуске приложения открывается главная страница на которой можно увидеть список мероприятий с их датами и направлениями, а также можно отсортировать мероприятия по направлению и дате (рисунки 29, 30).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 29 – Интерфейс главной страницы

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 30 – Пример фильтрации по дате и направлению мероприятий

С главной страницы разрабатываемого приложения можно перейти на страницу авторизации, на данной странице пользователь может ввести свой id и пароль для входа в систему.

При этом, при вводе некорректных данных или неверно введенной капчи система выдаст соответствующие ошибки авторизации под кнопкой «Войти» (рисунки 31-35).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 31 – Интерфейс страницы авторизации в приложении

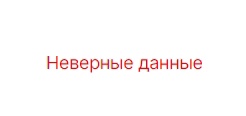


Рисунок 32 – Текст ошибки при вводе несуществующего пароля

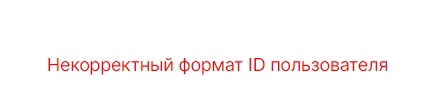


Рисунок 33 – Текст ошибки при вводе несуществующего ID пользователя

Изображение выглядит как текст, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 34 – Текст ошибки при некорректном вводе капчи

Изображение выглядит как текст, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 35 – Текст ошибки при некорректном вводе капчи или данных 3 раза

По результатам авторизации приложение перенаправляет пользователя на страницы соответствующие их ролям.

Кроме того, на ниже представленных страницах отображаются надписи «Доброе(ый) Утро/День/Вечер» и при обращении указывается «Mr» или «Ms» соответственно полу пользователя (рисунки 36 – 39).

Изображение выглядит как текст, снимок экрана, фиолетовый, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 36 – Интерфейс окна организатора

Изображение выглядит как текст, снимок экрана, фиолетовый, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 37 – Интерфейс окна участника

Изображение выглядит как текст, снимок экрана, диаграмма, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 38 – Интерфейс окна жюри

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 39 – Интерфейс окна модератора

Кроме того, в соответствии с функциональными возможностями создаваемого приложения, со страницы организатора можно перейти на страницу регистрации жюри и модератора, а уже в дальнейшем, после создания пользователя, приложение перенаправляет на главную страницу (рисунок 40).

Изображение выглядит как текст, собака, снимок экрана, млекопитающее

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 40 – Страница регистрации жюри и модератора

**ВЫВОД**: В ходе выполнения данного блока учебной практики, было создано десктопное приложение с подключением базы данных, играющее важную роль в работе с базами данных, особенно если они включают страницы регистрации и авторизации по следующим критериям:

* десктопное приложение может предоставлять удобный интерфейс для администраторов базы данных;
* позволяет выполнять CRUD-операции (создание, чтение, обновление, удаление записей) без использования сложных команд SQL;
* поддерживает офлайн-режим с возможностью синхронизации данных при подключении к интернету;
* десктопное приложение может содержать страницы регистрации и входа, где пользователи создают учетные записи;
* реализует проверку введенных данных перед их записью в базу данных в PostgreSQL;
* при необходимости можно настроить локальную базу данных (например, SQLite) и затем синхронизировать данные с сервером;
* в отличие от веб-приложений, десктопное ПО не зависит от браузера и может использовать мощность компьютера;
* обеспечивает лучший контроль над данными, так как можно ограничить доступ к файлам базы данных.

# ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 5 (06.03.2025 г.) - СОСТАВЛЕНИЕ ФАЙЛА О РУЧНОМ ТЕСТИРОВАНИИ

**Цель занятия:** Освоить практические навыки и умения в ручном тестировании десктопных приложений с подключением к ним баз данных

В ходе выполнения очередного этапа прохождения учебной практики необходимо было, в качестве последующего этапа реализовать 5 тестовых сценариев по функции регистрации жюри/модератора в десктопном приложении «Система управления конференциями» на Avalonia, созданное для организаторов конференций по информационной безопасности.

Тестовые сценарии представляющие собой описания последовательности действий и ожидаемых результатов, используемые для проверки работоспособности приложения, позволяют достичь необходимого уровня стабильности приложения, дают возможность убедиться, что оно ведет себя корректно, а также позволяют протестировать правильность валидации полей, систему аутентификации и т.п. функционал десктопного приложения.

При этом, были составлены 5 тестовых сценариев, представленных ниже (таблицы 1 – 5).

Таблица 1 – Тест «Регистрация без подтверждения пароля»

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Регистрация без подтверждения пароля |
| **Краткое изложение теста** | Проверка реакции системы при вводе пароля, но без повторного ввода. |
| **Этапы теста** | 1) Войти в систему под учетной записью организатора.  2) Открыть страницу регистрации.  3) Заполнить все обязательные поля, но оставить поле «Повтор пароля» пустым.  4) Нажать кнопку «ок».  5) Проверить, выдает ли система ошибку. |
| **Тестовые данные** | Фамилия: Иванов  Имя: Иван  Отчество: Иванович  Пол: мужской  Роль: Жюри  Email: ivanovivan@gmail.com  Телефон: 7(996)013-00-79  Направление: Дизайн  Видимое мероприятие  Мероприятие: Постоянно доступен: Налаживаем омниканальную коммуникацию с клиентами  Фото: «Выбрано»  Пароль: Password12!  Повтор пароля: \*пусто\* |
| **Ожидаемый результат** | Регистрация не происходит, система выдает сообщение: «Введите подтверждение пароля.» |
| **Фактический результат** | Регистрация не происходит, система выдает сообщение: «Все обязательные поля должны быть заполнены.» |
| **Статус** | Пройден |
| **Предварительное условие** | Вход выполнен под организатором. |
| **Постусловие** | Пользователь не зарегистрирован. |
| **Примечания/комментарии** | Нет |

Таблица 2 – Тест «Регистрация жюри/модератора с корректными данными»

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Регистрация жюри/модератора с корректными данными |
| **Краткое изложение теста** | Проверка корректной регистрации нового жюри/модератора при вводе всех требуемых данных. |
| **Этапы теста** | 1) Войти в систему под учетной записью организатора.  2) Открыть страницу регистрации жюри/модератора.  3) Заполнить все обязательные поля: ФИО, пол, роль, email, телефон, направление, мероприятие, пароль.  4) Указать корректный пароль (не менее 6 символов, буквы в разном регистре, цифра, спецсимвол).  5) Нажать кнопку «ок».  6) Проверить, что регистрация прошла успешно и пользователь добавлен в базу. |
| **Тестовые данные** | Фамилия: Иванов  Имя: Иван  Отчество: Иванович  Пол: мужской  Роль: Жюри  Email: ivanovivan@gmail.com  Телефон: 7(996)013-00-79  Направление: Дизайн  Видимое мероприятие  Мероприятие: Постоянно доступен: Налаживаем омниканальную коммуникацию с клиентами  Фото: «Выбрано»  Пароль: Password12!  Повтор пароля: Password12! |
| **Ожидаемый результат** | Новый пользователь успешно зарегистрирован. |
| **Фактический результат** | Новый пользователь успешно зарегистрирован, система переносит пользователя на главный экран. |
| **Статус** | Пройден |
| **Предварительное условие** | Вход выполнен под организатором. |
| **Постусловие** | Новый пользователь добавлен в систему. |
| **Примечания/комментарии** | Нет |

Таблица 3 – Тест «Регистрация с некорректным паролем»

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_3 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Регистрация с некорректным паролем |
| **Краткое изложение теста** | Проверка регистрации жюри при вводе пароля, не соответствующего требованиям. |
| **Этапы теста** | 1) Войти в систему под учетной записью организатора.  2) Открыть страницу регистрации.  3) Заполнить все обязательные поля кроме пароля и выбрать роль «Жюри».  4) Ввести некорректный пароль (например, без спецсимволов или короче 6 символов).  5) Нажать «ок».  6) Проверить, что система выдает сообщение об ошибке. |
| **Тестовые данные** | Пароль: pass12 |
| **Ожидаемый результат** | Система выдает ошибку о несоответствии пароля требованиям. |
| **Фактический результат** | Система выдает ошибку «Пароль слишком простой». |
| **Статус** | Пройден. |
| **Предварительное условие** | Вход выполнен под организатором. |
| **Постусловие** | Новый член жюри не зарегистрирован. |
| **Примечания/комментарии** | Нет |

Таблица 4 – Тест «Регистрация без заполнения обязательных полей»

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_4 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Регистрация без заполнения обязательных полей |
| **Краткое изложение теста** | Проверка реакции системы, если оставить пустыми обязательные поля. |
| **Этапы теста** | 1) Войти в систему под учетной записью организатора.  2) Открыть страницу регистрации.  3) Оставить одно или несколько обязательных полей пустыми.  4) Нажать «ок».  5) Проверить, что система выдает сообщение об ошибке. |
| **Тестовые данные** | Пустые поля: Email, Фамилия, Фото |
| **Ожидаемый результат** | Регистрация не выполняется, система показывает сообщение о незаполненных полях. |
| **Фактический результат** | Регистрация не выполняется, система показывает сообщение: «Все обязательные поля должны быть заполнены.» |
| **Статус** | Пройден |
| **Предварительное условие** | Вход выполнен под организатором. |
| **Постусловие** | Пользователь не зарегистрирован, система указывает на ошибку. |
| **Примечания/комментарии** | Нет |

Таблица 5 – Тест «Регистрация с некорректным email»

|  |  |
| --- | --- |
| **Test Case #** | TC\_UI\_5 |
| **Приоритет тестирования** | Средний. |
| **Заголовок/название теста** | Регистрация с некорректным email |
| **Краткое изложение теста** | Проверка работы валидации email при вводе некорректного адреса. |
| **Этапы теста** | 1) Войти в систему под учетной записью организатора.  2) Открыть страницу регистрации.  3) Ввести email без «@» или без доменного имени (например, userexample.com).  4) Нажать «ок».  5) Проверить, что система отклоняет регистрацию и выводит сообщение об ошибке. |
| **Тестовые данные** | Почта: userexample.com |
| **Ожидаемый результат** | Регистрация не происходит, система выдает сообщение: «Некорректный формат email.» |
| **Фактический результат** | Регистрация не происходит, система выдает сообщение: «У почты неверный шаблон» |
| **Статус** | Пройден. |
| **Предварительное условие** | Вход выполнен под организатором. |
| **Постусловие** | Пользователь не зарегистрирован. |
| **Примечания/комментарии** | Нет |

По результатам осуществленного тестирования десктопного приложения по вышеописанным тестам, можно резюмировать, что по всем направлениям тестирования были получены положительные результаты, иными словами, создаваемое десктопное приложение отвечает всем предустановленным его условиям, оно создано на должном уровне качества и соответствующей функциональной составляющей, предъявляемой его содержимому.

**ВЫВОД:** Использование тестовых сценариев в десктопных приложениях на Avalonia играет ключевую роль в обеспечении их стабильности, надежности и релевантности, позволяющую выявлять потенциальные ошибки, предотвращать критические сбои, а также значительно повышать безопасность пользовательских данных. Кроме того, тестирование помогает гарантировать корректное функционирование приложения на различных операционных системах, обеспечивая кроссплатформенную совместимость. Благодаря использованию тестов разработчики могут оперативно обнаруживать и устранять проблемы, что не только повышает качество продукта, но и существенно ускоряет процесс разработки, делая его более эффективным и предсказуемым.

# ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 6 (07.03.2025 г.) - СОСТАВЛЕНИЕ ИТОГОВЫХ ДОКУМЕНТОВ ПО БЛОКУ УЧЕБНОЙ ПРАКТИКИ, ПРОВЕРКА ЧУЖОЙ РАБОТЫ

На заключительном этапе первого блока прохождения учебной практики, осуществляемой под руководством преподавателя Е.П. Голубевой, были осуществлены следующие действия.

Изначально, была осуществлена проверка чужой работы, в том числе, по критериям, представленным в таблице 6.

Таблица 6 – Критерии проверки чужого варианта работы

|  |  |
| --- | --- |
| Кодировка критерия | Наименование критерия |
| А1 | Диаграмма прецедентов |
| А2 | Дизайн базы данных (ERD) - проверка по итогам двух сессий |
| А3 | Словарь данных |
| А7 | Оценка структуры проекта - проверка по итогам всех сессий |
| В1 | Разработка ПО - Главное окно системы |
| В2 | Разработка ПО – Авторизация |
| В3 | Разработка ПО – Окно организатора |
| В4 | Разработка ПО-Регистрация жюри и модераторов |
| В5 | Обработка ошибок/исключений - по итогам всех сессий |
| В6 | Импорт данных |
| С1 | Согласованный внешний вид приложения в соответствии с требованиями к макету - проверка по итогам всех сессий |
| С2 | Исходный код - проверка по итогам всех сессий |
| С3 | Предоставление результатов работы (git) - проверка по итогам всех сессий |
| С4 | Сообщения обратной связи с пользователем |
| С5 | Комментарии по коду - проверка по итогам всех сессий |
| D1 | Руководство пользователя |

# ЗАКЛЮЧЕНИЕ

Учебная практика по модулю ПМ. 01. Разработка модулей программного обеспечения для компьютерных систем проходила на базе Государственного бюджетного профессионального образовательного учреждения «Нижегородский Губернский колледж» в период с 01.03.2025 г. по 07.03.2025 г.

В качестве основной цели учебной практики выступала необходимость закрепления и углубления теоретических знаний, приобретенных в процессе лекционных, лабораторных и практический занятий и самостоятельной работы, необходимых профессиональных качеств будущего специалиста в избранном будущем профессиональном направлении деятельности, а также получение фундаментальных компетенций.

В целях достижения данной цели, в ходе прохождения учебной практики, был решен следующий комплекс задач:

* были реализованы на практике теоретические знания, освоены и получены практические навыки в их реализации;
* были сформированы более детального представления о будущей профессии;
* приобретены практические знания и опыт работы по выбранной специальности;
* получен субъективный опыт выработки компетенций самостоятельного формирования первичных данных для отражения результатов прохождения практики;
* сформированы навыки критического и творческого анализа вторичных данных и литературных источников по теме практики;
* отработаны и закреплены методы, приемы и способы разработки программного обеспечения;
* освоены принципы технологии структурного и объектно-ориентированного программирования, а также навыки и способы оптимизации и приемов рефакторинга;
* освоение основные принципы отладки и тестирования программных продуктов;
* были собраны материалы, необходимых для составления отчета по практике.

Таким образом, в процессе прохождения учебной практики была осуществлена разнообразная деятельность в соответствии с профилем подготовки и выдаваемыми преподавателем заданиями, по результатам которых, можно сделать вывод о том, что прохождение учебной практики является важным элементом учебного процесса, поскольку именно в результате прохождения практики были получены и освоены практический опыт и навыки работы по разработке модулей программного обеспечения для компьютерных систем. Решенные в ходе прохождения учебной практики практические проблемы, несомненно, позволили более глубоко усвоить и понять теоретический материал, полученный ранее в ходе лекционных занятий, изучить реальный механизм действий в той или иной ситуации.

Таким образом, можно считать, что ставленные в ходе прохождения учебной практики цели и задачи, были решены в полном объеме и в соответствии с установленным графиком ее прохождения.