Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Марийский государственный университет»

Институт цифровых технологий

Физико-математический факультет

Отчет по дисциплине «Технология разработки программного обеспечения»

Лабораторная работа №2

Выполнил: студент гр. И-32

Медведев Н.Е.

Проверил: преподаватель

Забродин Д.В.

Йошкар-Ола

2025

**Содержание**

[Введение 3](#_Toc209193336)

[Теоретическая часть 4](#_Toc209193337)

[Практическая часть 5](#_Toc209193338)

[Заключение 12](#_Toc209193339)

**Введение**

Цель работы – провести построение архитектуры информационной системы колледжа.

Задача – получить навыки построение архитектуры с помощью case-средства Draw.io.

**Теоретическая часть**

Схема «сущность-связь» (также ERD или ER-диаграмма) — это разновидность блок-схемы, где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы. ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ER-диаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей. Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи — глаголов.

Диаграмма классов – диаграмма языка UML, на которой представлена совокупность декларативных или статических элементов модели, таких как классы с атрибутами и операциями, а также связывающие их отношения.

Она предназначена для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования.

Диаграмма содержит интерфейсы, пакеты, отношения и даже отдельные экземпляры классификаторов, такие как объекты и связи.

**Практическая часть**

Создадим er-диаграмму для информационной системы колледжа с помощью case-средства Draw.io. Создадим сущности (Рисунок 1).

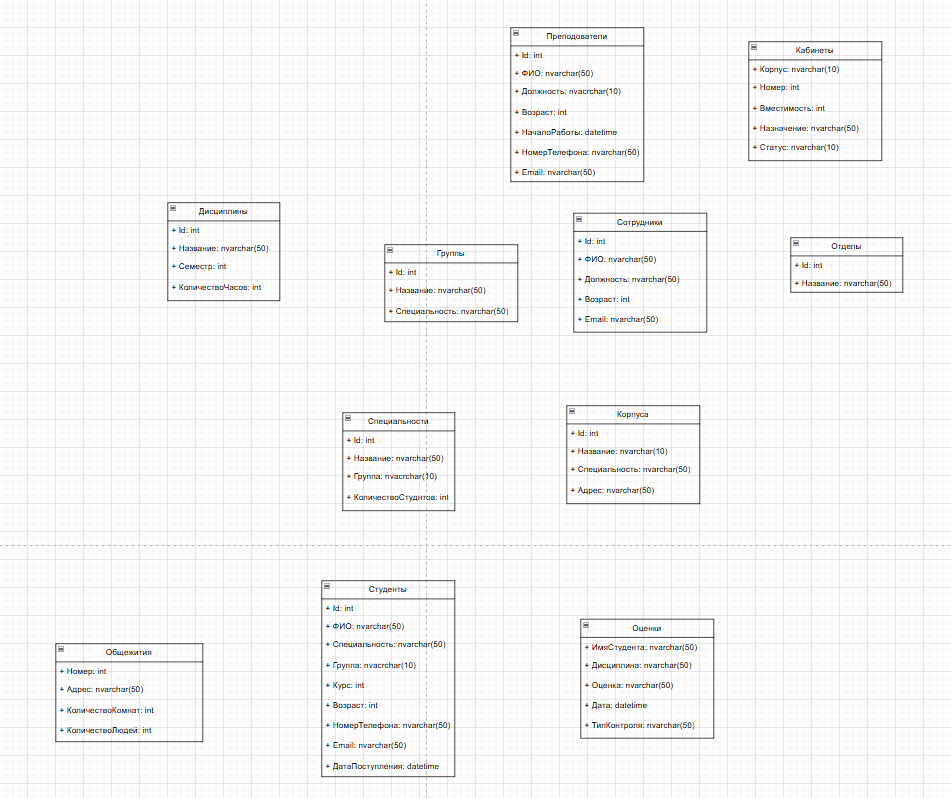


Рисунок 1 – Создание сущностей

Сущности:

1. Таблица «Специальности»:

Эта таблица включает информацию специальностей, которые есть в колледже. Она включает в себя такие поля, как, Id, название, группа, количество студентов.

1. Таблица «Корпуса»

Она хранит информацию о корпусах колледжа. Включает Id, название, специальность, адрес.

1. Таблица «Дисциплины»

Содержит информацию о дисциплине. Включает Id, название, семестр, количество часов.

1. Таблица «Преподаватели»

Эта таблица включает список преподавателей, работающих в колледже. Она включает в себя такие поля, как Id, ФИО, должность, возраст, начало работы, номер телефона, email.

1. Таблица «Отделы»

Эта таблица хранит информацию об отделах колледжа. Включает Id, название.

1. Таблица «Группы»

Включает информацию о группах. Включает Id, название, специальности.

1. Таблица «Студенты»

Хранит информацию о студентах, которые учатся в колледже. Включает Id, ФИО, специальность, группа, курс, возраст, номер телефона, email, дата поступления.

1. Таблица «Оценки»

Включает информацию об оценках, который получил студент. Содержит такие поля, как имя студента, дисциплина, преподаватель, оценка, дата, тип контроля.

1. Таблицы «Сотрудники»

Содержит информацию о сотрудниках, которые работают в колледже. Включает Id, ФИО, должность, возраст, номер телефона, email.

1. Таблица «Кабинеты»

Включает список кабинетов, расположенных в корпусах колледжей. Она включает в себя такие поля, как корпус, номер, вместимость, назначение, статус (открыт, закрыт, свободно, занято).

1. Таблица «Общежития»

Хранит информацию об общежитии колледжа. Включает номер, адрес, количество комнат, количество людей.

Соединим их (Рисунок 2).

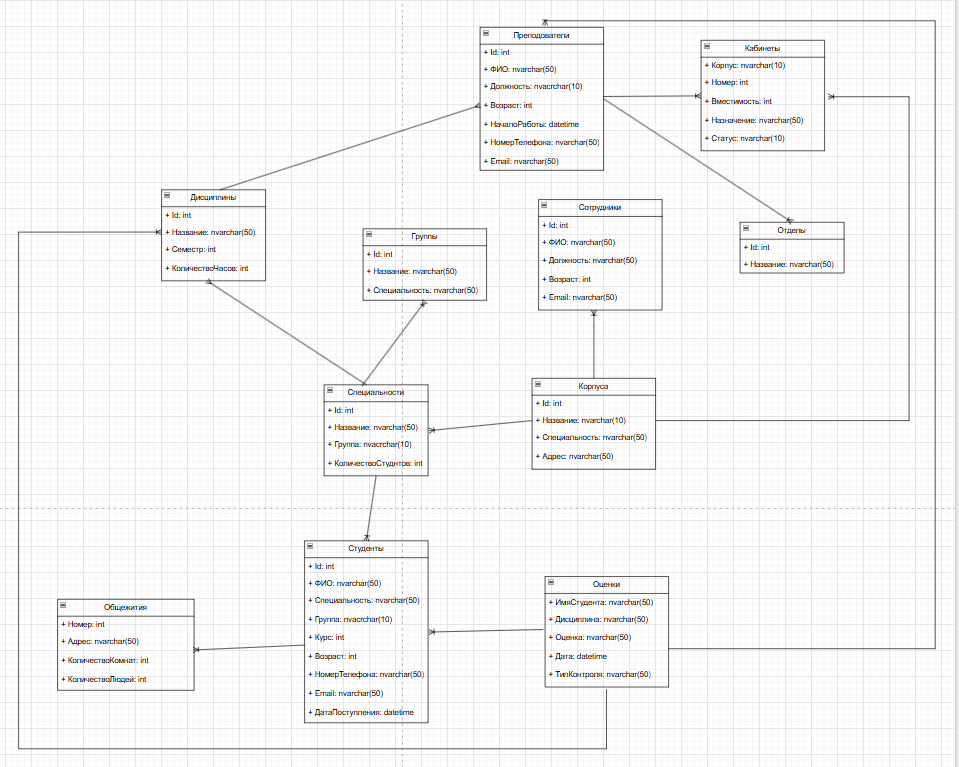


Рисунок 2 – Er - диаграмма информационной системы колледжа

Связи:

1. «Специальности – Студенты» (Один-ко-многим)

Позволяет определить, на какой специальности учится каждый студент.

1. «Специальности – Дисциплины» (Многие-ко-многим)

Определяет учебный план для каждой специальности.

1. «Дисциплины – Преподаватели» (Многие-ко-многим)

Указывает, какой преподаватель ведёт дисциплину.

1. «Студенты – Общежития» (Один-ко-многим)

Определять, в каком общежитии проживает студент.

1. «Преподаватели – Кабинеты» (Один-ко-многим/Многие-ко-многим)

Показывает, какие кабинеты используются преподавателями для проведения занятий или как личные кабинеты.

1. **«**Специальности – Группы» (Один-ко-многим)

Каждая специальность может иметь несколько групп, но каждая группа принадлежит только одной специальности. Это позволит определить, какие группы относятся к какой специальности.

1. «Оценки – Студенты» (Один-ко-многим)

Каждый студент может иметь несколько оценок, но каждая оценка относится к одному студенту.

1. «Оценки – Дисциплины» (Один-ко-многим)

Каждая дисциплина может иметь несколько оценок, но каждая оценка относится к одной дисциплине.

1. «Оценки – Преподаватели» (Один-ко-многим)

Каждый преподаватель может выставить несколько оценок, но каждая оценка выставлена одним преподавателем.

1. «Преподаватели – Отделы» (Один-ко-многим)

Каждый отдел может содержать несколько преподавателей, но каждый преподаватель принадлежит только одному отделу.

1. «Корпуса – Специальности» (Один-ко-многим)

Каждый корпус может вмещать несколько специальностей, при этом каждая специальность относится к одному корпусу.

1. «Корпуса – Кабинеты» (Один-ко-многим)

Каждый корпус содержит несколько кабинетов. Каждый кабинет находится только в одном корпусе.

1. «Корпуса – Группы» (Один-ко-многим)

Можно учитывать, что у каждой группы есть основной корпус, где проходят занятия.

1. «Корпуса – Отделы» (Один-ко-многим)

В одном корпусе может располагаться несколько отделов, но каждый отдел находится в одном корпусе.

Создадим диаграмму классов для информационной системы колледжа с помощью case-средства Draw.io. Создадим классы, такие же как сущности в er-диаграмме (Рисунок 3).

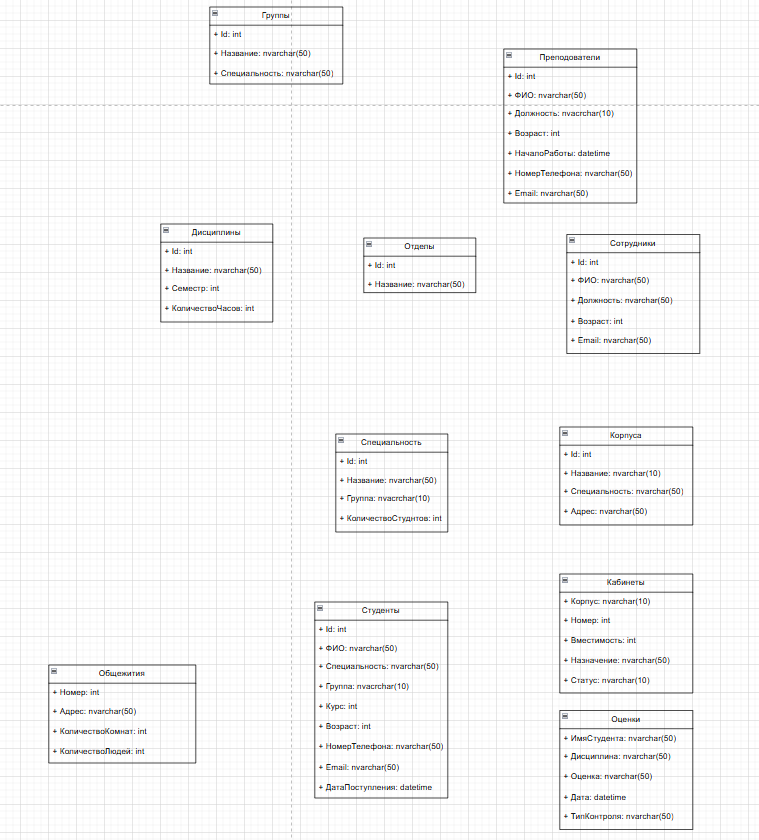


Рисунок 3 – Создание классов

Соединим эти классы (Рисунок 4).

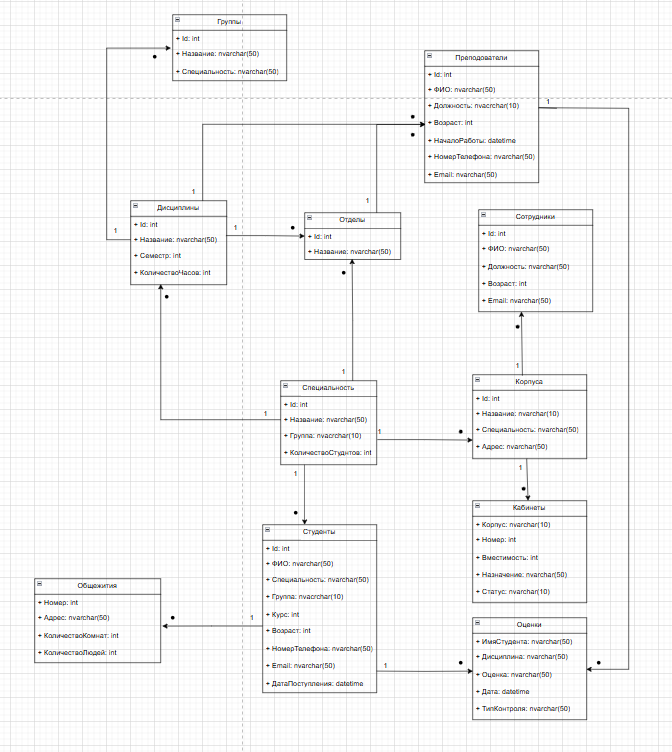


Рисунок 4 – Диаграмма классов информационной системы колледжа

Связи:

1. Группы (1) – Студенты (\*)

Одна группа может содержать много студентов.

1. Специальность (1) – Группы (\*)

Одна специальность может иметь много групп.

1. Специальность (1) – Студенты (\*)

Одна специальность может иметь много студентов.

1. Преподаватели (\*) – Дисциплины (\*)

Много преподавателей могут преподавать много дисциплин, и много дисциплин могут быть преподаваемы многими преподавателями.

1. Отделы (1) – Преподаватели (\*)

Один отдел может иметь много преподавателей.

1. Отделы (1) – Сотрудники (\*)

Один отдел может иметь много сотрудников.

1. Корпуса (1) – Кабинеты (\*)

Один корпус может иметь много кабинетов.

1. Общежития (1) – Студенты (\*)

Одно общежитие может размещать много студентов.

1. Студенты (\*) – Оценки (\*) – Дисциплины (\*)

Студенты получают оценки по дисциплинам. Это связь "многие ко многим" между студентами и дисциплинами, реализованная через таблицу "Оценки".

**Заключение**

В ходе выполнения лабораторной работы, удалось провести построение архитектуры информационной системы колледжа.