**АВТОНОМАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ**

**ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ**

**МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ**

**«АКАДЕМИЯ ТОП»**

**ДОКУМЕНТАЦИЯ ИНДИВИДУАЛЬНОГО ПРОЕКТА**

**На тему** “Система управления колледжем”

Выполнили:

студент гр. РПО 9/23/2

Сухинин А.В

Проверила:

Голякова Т.О.

г. Санкт-Петербург

2025

**Введение**

Данный документ содержит описание архитектуры консольного приложения на C# для управления данными колледжа. Система разработана для эффективного управления информацией о студентах, преподавателях, курсах, группах, расписании, успеваемости и аудиториях. В документе представлены UML-диаграммы, отражающие структуру и поведение системы, а также подробное описание её функциональности.

**Решаемая проблема**

Современные образовательные учреждения сталкиваются с необходимостью эффективного управления большими объемами данных, связанных с учебным процессом. Отсутствие централизованной системы управления информацией приводит к ряду проблем: дублирование данных, сложности в поиске и обновлении информации, трудности в формировании отчетов и анализе успеваемости. Разрабатываемая система призвана решить эти проблемы путем создания единой базы данных и удобного интерфейса для работы с ней.

Система управления колледжем позволяет автоматизировать рутинные процессы администрирования учебного заведения, такие как зачисление студентов, формирование групп, составление расписания, учет успеваемости. Это значительно сокращает время на выполнение административных задач и минимизирует вероятность ошибок, связанных с человеческим фактором.

**Ключевые функции системы**

Система управления колледжем предоставляет следующие ключевые функции:

Управление данными о студентах: добавление новых студентов, редактирование информации о существующих студентах, удаление студентов из системы, поиск студентов по различным критериям, зачисление студентов в группы и перевод между группами.

Управление данными о преподавателях: добавление новых преподавателей, редактирование информации о существующих преподавателях, удаление преподавателей из системы, поиск преподавателей по различным критериям, назначение преподавателей на курсы.

Управление курсами: создание новых курсов, редактирование информации о существующих курсах, удаление курсов, поиск курсов по различным критериям, связывание курсов с преподавателями и группами.

Управление группами: создание новых групп, редактирование информации о существующих группах, удаление групп, просмотр списка студентов в группе.

Управление расписанием: добавление занятий в расписание, редактирование существующих занятий, удаление занятий, просмотр расписания для групп и преподавателей.

Управление успеваемостью: выставление оценок студентам, редактирование и удаление оценок, просмотр успеваемости студентов по курсам.

Поиск и фильтрация данных: возможность поиска и фильтрации данных по различным критериям для всех сущностей системы.

Все эти функции реализованы через консольный интерфейс, который обеспечивает удобное взаимодействие пользователя с системой.

**Выбранные UML-диаграммы и их роль**

Для проектирования системы были выбраны три типа UML-диаграмм: диаграмма классов, диаграмма вариантов использования и диаграмма последовательностей. Каждая из этих диаграмм играет важную роль в понимании различных аспектов системы.

**Диаграмма классов**

Диаграмма классов является центральной в проектировании системы, так как она отображает статическую структуру системы, включая классы, их атрибуты, методы и отношения между ними. Эта диаграмма была выбрана, потому что она наиболее полно отражает структуру данных и взаимосвязи между различными сущностями системы.

На диаграмме классов представлены следующие ключевые классы:

Person: абстрактный базовый класс для студентов и преподавателей, содержащий общие атрибуты, такие как имя, email, телефон и адрес.

Student: класс, наследующийся от Person, представляющий студента колледжа.

Преподаватель: класс, наследующийся от Person, представляющий преподавателя колледжа.

Курс: класс, представляющий учебный курс.

Группа: класс, представляющий учебную группу.

ScheduleEntry: класс, представляющий запись в расписании.

Mark: класс, представляющий оценку студента по определенному курсу.

Отделение: класс, представляющий кафедру или отдел колледжа.

Диаграмма классов также отображает различные типы связей между классами, такие как наследование (Student и Teacher наследуются от Person), ассоциации (например, между Student и Group, Teacher и Course) и композиции. Эти связи помогают понять, как различные сущности взаимодействуют друг с другом в системе.

**Диаграмма вариантов использования**

Диаграмма вариантов использования была выбрана для отображения функциональных требований к системе с точки зрения пользователей. Она показывает, какие действия могут выполнять различные типы пользователей (акторы) в системе.

На диаграмме вариантов использования представлены следующие акторы:

Администратор: имеет полный доступ ко всем функциям системы.

Преподаватель: может просматривать свое расписание, выставлять оценки, просматривать списки студентов и искать информацию.

Студент: может просматривать свою успеваемость, расписание группы и искать информацию о курсах.

Для каждого актора определены соответствующие варианты использования, которые отражают действия, доступные этому актору в системе. Например, администратор может добавлять, редактировать и удалять пользователей, а преподаватель может выставлять оценки и просматривать списки студентов.

Диаграмма вариантов использования помогает понять, какие функции должна предоставлять система различным типам пользователей, и служит основой для разработки пользовательского интерфейса.

**Диаграмма последовательностей**

Диаграмма последовательностей была выбрана для иллюстрации взаимодействия объектов системы во времени при выполнении ключевых сценариев. Она показывает, как объекты обмениваются сообщениями для выполнения определенной задачи.

На диаграмме последовательностей представлены два ключевых сценария:

Добавление нового студента: показывает последовательность действий при добавлении нового студента в систему, включая создание объекта Person, создание объекта Student, выбор группы и добавление студента в группу.

Выставление оценки студенту: показывает последовательность действий при выставлении оценки студенту, включая поиск студента, выбор курса, создание объекта Mark и связывание его с объектами Student и Course.

Диаграмма последовательностей помогает понять динамическое поведение системы и взаимодействие объектов при выполнении конкретных задач. Она также служит руководством для разработчиков при реализации соответствующей функциональности.

**Как диаграммы помогают понять архитектуру приложения**

Выбранные UML-диаграммы в совокупности предоставляют полное представление об архитектуре приложения с разных точек зрения:

Диаграмма классов дает представление о статической структуре системы, показывая, какие классы существуют в системе, какие атрибуты и методы они имеют, и как они связаны друг с другом. Это помогает разработчикам понять, как организованы данные в системе и как различные компоненты взаимодействуют друг с другом.

Диаграмма вариантов использования предоставляет функциональный взгляд на систему, показывая, какие действия могут выполнять различные типы пользователей. Это помогает разработчикам понять требования к системе с точки зрения пользователей и обеспечить реализацию всех необходимых функций.

Диаграмма последовательностей иллюстрирует динамическое поведение системы, показывая, как объекты взаимодействуют друг с другом во времени при выполнении конкретных задач. Это помогает разработчикам понять, как должны быть реализованы различные функции системы и как должны взаимодействовать объекты для выполнения этих функций.

Вместе эти диаграммы обеспечивают комплексное понимание архитектуры приложения, охватывая как статические аспекты (структура данных), так и динамические аспекты (поведение системы). Они служат руководством для разработчиков при реализации системы и помогают обеспечить соответствие реализации проектным требованиям.

**Заключение**

Разработанная архитектура системы управления колледжем обеспечивает эффективное решение задачи управления данными образовательного учреждения. Система предоставляет широкий спектр функций для работы с информацией о студентах, преподавателях, курсах, группах, расписании и успеваемости.

UML-диаграммы, представленные в этом документе, дают полное представление о структуре и поведении системы, что облегчает понимание её архитектуры и служит основой для дальнейшей разработки. Диаграмма классов отражает статическую структуру системы, диаграмма вариантов использования показывает функциональные требования с точки зрения пользователей, а диаграмма последовательностей иллюстрирует динамическое поведение системы при выполнении ключевых сценариев.

Такой подход к проектированию обеспечивает создание гибкой, расширяемой и удобной в использовании системы, которая эффективно решает задачи управления данными колледжа.