

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

*Методические рекомендации к лабораторным работам
для студентов специальности 1-53 01 02
«Автоматизированные системы обработки информации»
очной и заочной форм обучения*



Могилев 2023

УДК 00.891
ББК 32.973.202
П79

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «28» марта 2023 г., протокол № 9

Составитель канд. техн. наук, доц. С. К. Крутолевич

Рецензент канд. техн. наук, доц. В. М. Ковальчук

Методические рекомендации к лабораторным работам предназначены для студентов специальности 1-53 01 02 «Автоматизированные системы обработки информации» очной и заочной форм обучения.

Учебное издание

ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Ответственный за выпуск	В. В. Кутузов
Корректор	Т. А. Рыжикова
Компьютерная верстка	М. М. Дударева

Подписано в печать 29.08.2023 . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. 1,40. Уч.-изд. л. 1,44. Тираж 21 экз. Заказ № 923.

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2023

Содержание

Введение.....	4
1 Лабораторная работа № 1. Общая характеристика CASE-средства Enterprise Architect	5
2 Лабораторная работа № 2. Разработка диаграммы вариантов использования	8
3 Лабораторная работа № 3. Разработка диаграммы классов на уровне сущностей	11
4 Лабораторная работа № 4. Разработка атрибутов, операций объектов и отношений между ними на диаграмме классов	12
5 Лабораторная работа № 5. Разработка диаграммы кооперации и редактирование свойств ее элементов	15
6 Лабораторная работа № 6. Разработка диаграммы последовательности и редактирование свойств ее элементов	16
7 Лабораторная работа № 7. Разработка диаграммы состояний и редактирование свойств ее элементов	18
8 Лабораторная работа № 8. Разработка диаграммы деятельности и редактирование свойств ее элементов	20
9 Лабораторная работа № 9. Разработка диаграммы компонентов и редактирование свойств ее элементов	21
10 Лабораторная работа № 10. Разработка диаграммы развертывания и редактирование свойств ее элементов	22
Список литературы	23

Введение

Методические рекомендации состоят из 10 лабораторных работ, выполнение которых закрепляет материал, изложенный в курсе «Проектирование автоматизированных систем».

В результате выполнения лабораторных работ студент должен продемонстрировать следующие умения:

- осуществлять проектирование автоматизированных систем с использованием CASE-средства Enterprise Architect;
- осуществлять документирование требований заказчика с помощью диаграмм: вариантов использования и бизнес-процессов;
- документировать поведение проектируемой системы с использованием диаграмм поведения;
- разрабатывать структуру проектируемой системы с использованием диаграммы классов.

Лабораторные работы выполняются с использованием программного обеспечения Enterprise Architect с использованием не ниже 12 версии. Каждая диаграмма распечатывается на отдельном листе. В отчете по лабораторным работам также присутствуют ответы на контрольные вопросы.

Методические рекомендации по лабораторным работам содержат теоретические сведения, порядок выполнения лабораторных работ, требования к содержанию отчетов, набор контрольных вопросов.

Для выполнения лабораторных работ необходимо получить у преподавателя индивидуальное задание.

Методические рекомендации предназначены для оказания помощи студентам в организации работы по выполнению лабораторных работ по дисциплине.

1 Лабораторная работа № 1. Общая характеристика CASE-средства Enterprise Architect

Цель работы

Изучение основных возможностей и режимов работы CASE-средства Enterprise Architect.

Теоретические сведения

Рабочее пространство приложения Enterprise Architect содержит набор окон, контекстных меню и панелей инструментов. Вместе эти элементы обеспечивают простую и гибкую работу с приложением. В этой концепции рабочее пространство Enterprise Architect похоже на Microsoft Outlook и Microsoft Visual Studio.

При запуске Enterprise Architect первое, что отображается, это стартовая страница.

Эта страница предоставляет следующие возможности.

Search – поиск объектов в Enterprise Architect. Введите имя объекта в текстовое поле и нажмите кнопку [...].

Getting Started открывает Tasks Pain. Отображает полезные темы и руководства для различных областей деятельности в Enterprise Architect.

Online Resources & Tutorials открывает страницу веб-сайта компании Sparx Systems, который предоставляет доступ к широкому кругу уроков по языку UML, демонстраций, примеров, надстроек и обсуждений.

Configure Options отображает диалоговое окно Options, которое позволяет определить, каким образом Enterprise Architect отображает и обрабатывает информацию.

Open a Project File отображает диалоговое окно Open Project, которое используется для открытия существующего проекта.

Create a New Project позволяет сохранить новый проект.

Copy a Base Project позволяет копировать базовый проект.

Connect to Server служит для выбора имени источника данных для подключения.

Браузер проекта (Project Browser) позволяет перемещаться по пространству проекта Enterprise Architect. В нем отображаются пакеты, диаграммы, элементы и их свойства. Можно перетаскивать элементы из папки в папку или из браузера прямо на диаграмму.

Браузер проекта служит для управления и представления всех элементов в модели.

Браузер проекта можно разделить на представления, каждое из которых содержит диаграммы, пакеты и другие элементы.

Представление вариантов использования включает модели вариантов использования и бизнес-процессов.

Динамическое представление включает диаграммы состояний, деятельности и последовательности. Логическое представление включает модель классов.

Представление компонентов – это представление системных компонентов (исполняемые, DDL и другие компоненты).

Представление размещения – физическая модель, которая показывает, какие использованы технические средства ЭВМ и какое программное обеспечение установлено.

Панель инструментов Enterprise Architect представляется как панель с иконками, которые используются для создания элементов и связей между ними. К тому же родственные элементы и связи организованы в страницы. Каждая страница содержит элементы и связи, используемые в конкретном типе диаграммы.

Анализ предметной области «Система регистрации для вуза».

Перед руководителем информационной службы университета ставится задача разработки новой клиент-серверной системы регистрации студентов. Система должна позволять студентам регистрироваться на курсы и просматривать свои таблицы успеваемости с персональных компьютеров, подключенных к локальной сети университета. Профессора должны иметь доступ к онлайн-системе, чтобы указать курсы, которые они будут читать, и проставить оценки за курсы.

Из-за недостатка средств университет не в состоянии заменить сразу всю существующую систему. Остается функционировать в прежнем виде база данных, содержащая всю информацию о курсах (каталог курсов). Эта база данных поддерживается реляционной СУБД. Новая система будет работать с существующей БД в режиме доступа, без обновления.

В начале каждого семестра студенты могут запросить каталог курсов, содержащий список курсов, предлагаемых в данном семестре. Информация о каждом курсе должна включать имя профессора, наименование кафедры и требования к предварительному уровню подготовки (прослушанным курсам).

В конце семестра студенты должны иметь доступ к системе для просмотра своих электронных таблиц успеваемости. Поскольку эта информация конфиденциальная, система должна обеспечивать ее защиту от несанкционированного доступа.

Профессора должны иметь доступ к онлайн-системе, чтобы указать курсы, которые они будут читать, и просмотреть список студентов, записавшихся на их курсы. Кроме этого, профессора должны иметь возможность проставить оценки за курсы.

Составление глоссария проекта.

Глоссарий предназначен для описания терминологии предметной области. Он может быть использован как неформальный словарь данных системы (таблица 1.1).

Таблица 1.1 – Глоссарий проекта

Термин предметной области	Описание
Курс	Учебный курс, предлагаемый университетом
Конкретный курс (Course Offering)	Конкретное чтение данного курса в конкретном семестре (один и тот же курс может вестись в нескольких параллельных сессиях). Включает точные дни недели и время
Каталог курсов	Полный каталог всех курсов, предлагаемых университетом
Расчетная система	Система обработки информации об оплате за курсы
Оценка	Оценка, полученная студентом за конкретный курс
Профессор	Преподаватель университета
Табель успеваемости (Report Card)	Все оценки за все курсы, полученные студентом в данном семестре
Список курса (Roster)	Список всех студентов, записавшихся на конкретный курс
Студент	Личность, проходящая обучение в университете
Учебный график (Schedule)	Курсы, выбранные студентом в текущем семестре

Описание дополнительных спецификаций.

Назначение дополнительных спецификаций – определить требования к системе регистрации курсов, которые не отражены в модели вариантов использования. Вместе они образуют полный набор требований к системе.

Дополнительные спецификации определяют нефункциональные требования к системе, такие как надежность, удобство использования, производительность, сопровождаемость, а также ряд функциональных требований, являющихся общими для нескольких вариантов использования.

Функциональные возможности: система должна обеспечивать многопользовательский режим работы. Если конкретный курс оказывается заполненным в то время, когда студент формирует свой учебный график, включающий данный курс, то система должна известить его об этом.

Надежность: система должна быть в работоспособном состоянии 24 ч в день 7 дней в неделю, время простоя – не более 10 %.

Производительность: система должна поддерживать до 2000 одновременно работающих с центральной базой данных пользователей и до 500 пользователей, одновременно работающих с локальными серверами.

Безопасность: система не должна позволять студентам изменять любые учебные графики, кроме собственных, а также не должна позволять профессорам модифицировать конкретные курсы, выбранные другими профессорами. Только профессора имеют право ставить студентам оценки. Только регистратор может изменять любую информацию о студентах.

Проектные ограничения: система должна быть интегрирована с существующей системой каталога курсов, функционирующей на основе реляционной СУБД.

Создание модели вариантов использования.

Действующие лица:

Student (Студент) – записывается на курсы;

Professor (Профессор) – выбирает курсы для преподавания;

Registrar (Регистратор) – формирует учебный план и каталог курсов, ведет все данные о курсах, профессорах и студентах;

Billing System (Расчетная система) – получает от данной системы информацию по оплате за курсы;

Course Catalog (Каталог курсов) – передает в систему информацию из каталога курсов, предлагаемых университетом.

Задание для самостоятельного выполнения

Освоить приемы работы в среде Enterprise Architect.

Кратко описать предметную область и сформировать глоссарий проекта.

Контрольные вопросы

1 Краткое описание основных режимов работы Enterprise Architect.

2 Перечислить основные диаграммы для моделирования АСОИ.

3 Перечислить основные элементы экрана.

4 Дать характеристику Браузеру.

5 Дать характеристику Окну диаграмм.

6 Дать характеристику четырем представлениям модели.

7 Цель создания глоссария проекта.

8 Кто выступает в качестве действующих лиц?

2 Лабораторная работа № 2. Разработка диаграммы вариантов использования

Цель работы

Изучение технологии формирования диаграммы вариантов использования.

Теоретические сведения

Создание действующих лиц и вариантов использования.

Чтобы поместить действующее лицо в браузер, щелкните левой кнопкой мыши на панели инструментов Toolbox по кнопке Actor.

Исходя из потребностей действующих лиц выделяются следующие варианты использования:

Login (Войти в систему);

Register for Courses (Зарегистрироваться на курсы);

View Report Card (Просмотреть таблицу успеваемости);
 Select Courses to Teach (Выбрать курсы для преподавания);
 Submit Grades (Проставить оценки);
 Maintain Professor Information (Вести информацию о профессорах);
 Maintain Student Information (Вести информацию о студентах);
 Close Registration (Закрыть регистрацию).

Диаграмма вариантов использования. Готовая диаграмма вариантов использования должна выглядеть, как на рисунке 2.1.

Описание основных и альтернативных потоков событий.

Создайте с помощью MS Word три текстовых файла с описаниями вариантов использования Login (Войти в систему), Register for Courses (Зарегистрироваться на курсы) и Close Registration (Закрыть регистрацию).

Вариант использования Login.

Краткое описание. Данный вариант использования описывает вход пользователя в систему регистрации курсов.

Основной поток событий. Данный вариант использования начинает выполняться, когда пользователь хочет войти в систему регистрации курсов.

Система запрашивает имя пользователя и пароль.

Пользователь вводит имя и пароль.

Система проверяет имя и пароль, после чего открывается доступ в систему.

Альтернативные потоки. Неправильное имя/пароль. Если во время выполнения Основного потока обнаружится, что пользователь ввел неправильное имя и/или пароль, система выводит сообщение об ошибке. Пользователь может вернуться к началу Основного потока или отказаться от входа в систему, при этом выполнение варианта использования завершается.

Предусловия. Отсутствуют.

Постусловия. Если вариант использования выполнен успешно, пользователь входит в систему. В противном случае состояние системы не изменяется.

Вариант использования Register for Courses.

Краткое описание. Данный вариант использования позволяет студенту зарегистрироваться на конкретные курсы в текущем семестре. Студент может изменить свой выбор (обновить или удалить курсы), если изменение выполняется в установленное время в начале семестра. Система каталога курсов предоставляет список всех конкретных курсов текущего семестра.

Основной поток событий. Данный вариант использования начинает выполняться, когда студент хочет зарегистрироваться на конкретные курсы или изменить свой график курсов.

Система запрашивает требуемое действие (создать график, обновить график, удалить график).

Когда студент указывает действие, выполняется один из подчиненных потоков (создать, обновить, удалить или принять график).

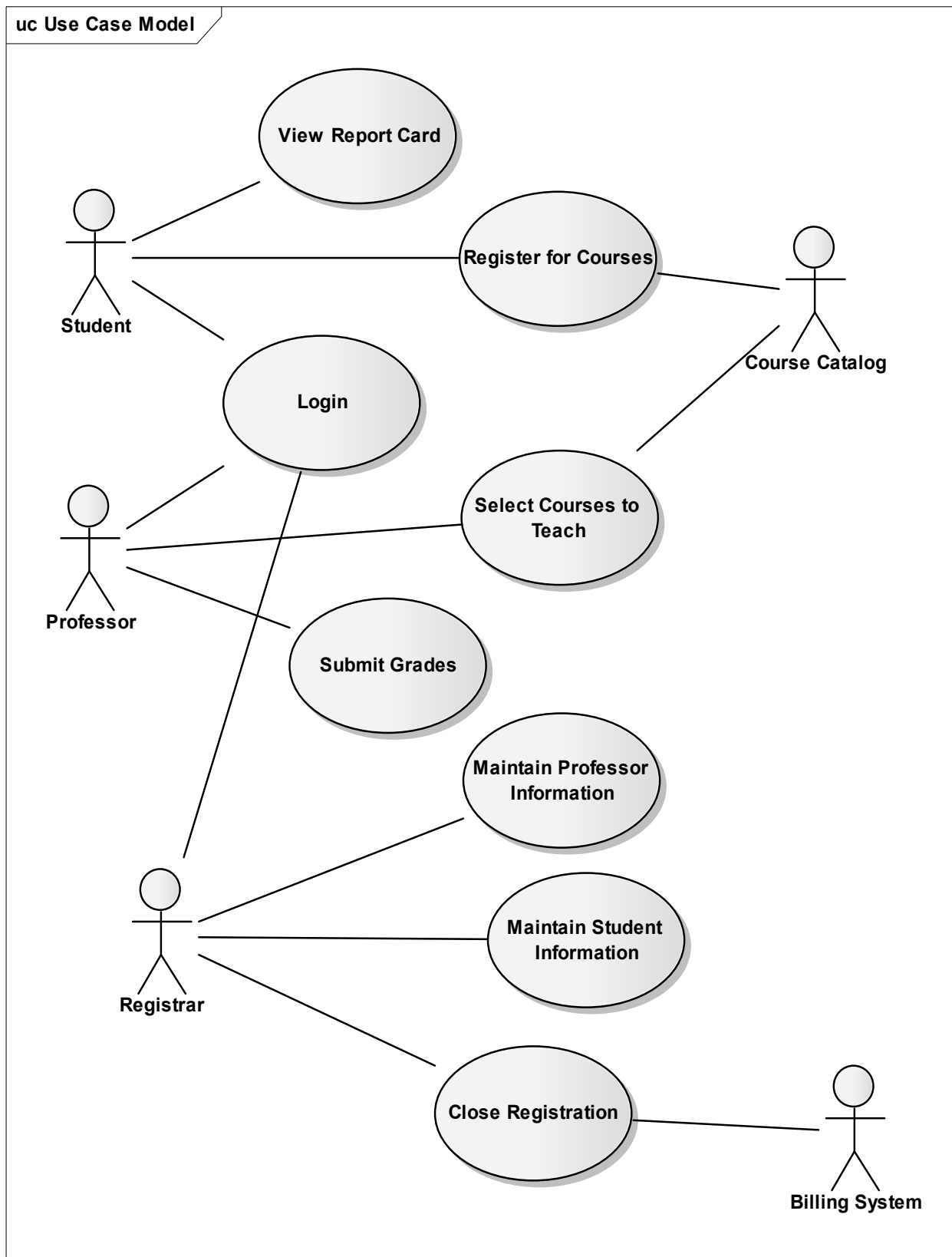


Рисунок 2.1 – Диаграмма вариантов использования для системы регистрации

Создать график.

Система выполняет поиск в каталоге курсов доступных конкретных курсов и выводит их список.

Студент выбирает из списка четыре основных и два альтернативных курса.

После выбора система создает график студента.

Выполняется подчиненный поток «Принять график».

Обновить график.

Система выводит текущий график студента.

Система выполняет поиск в каталоге курсов доступных конкретных курсов и выводит их список.

Студент может обновить свой выбор курсов, удаляя или добавляя конкретные курсы.

После выбора система обновляет график.

Выполняется подчиненный поток «Принять график».

Удалить график.

Система выводит текущий график студента.

Система запрашивает у студента подтверждение удаления графика.

Студент подтверждает удаление.

Система удаляет график. Если график включает конкретные курсы, на которые записался студент, он должен быть удален из списков этих курсов.

Задание для самостоятельного выполнения

Разработать диаграммы вариантов использования по индивидуальному заданию.

Контрольные вопросы

1 Цель диаграммы вариантов использования.

2 Дать определение терминам «вариант использования», «действующее лицо».

3 Виды связей между вариантами использования и действующими лицами.

3 Лабораторная работа № 3. Разработка диаграммы классов на уровне сущностей

Цель работы

Изучение последовательности разработки диаграммы классов.

Диаграммы последовательности позволяют сформировать свойства и операции (методы) классов. Вид классов после добавления свойств и операций представлен на рисунке 3.1.

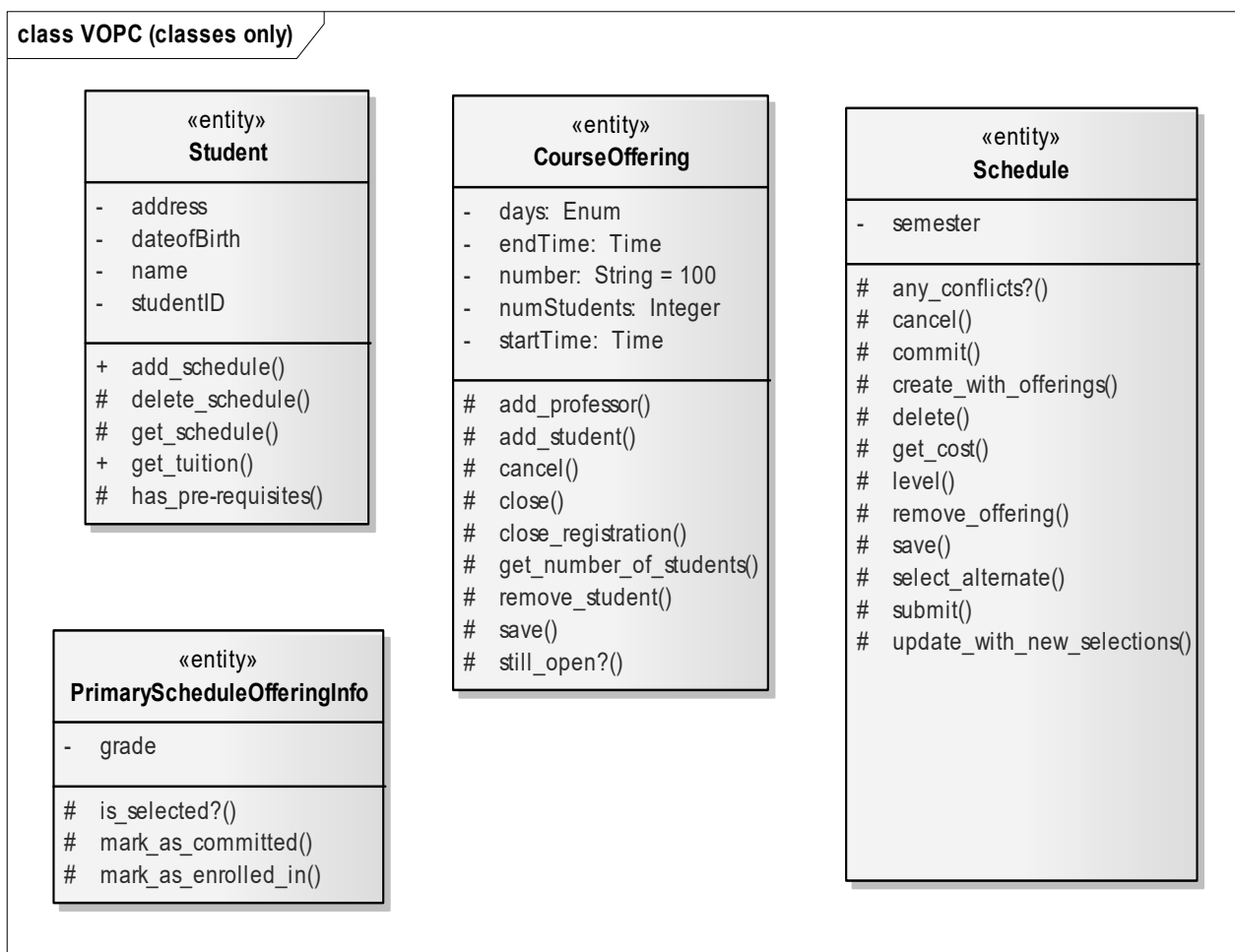


Рисунок 3.1 – Диаграмма классов с указанием свойств и операций

Задание для самостоятельного выполнения

Разработать диаграммы классов по индивидуальному заданию.

Контрольные вопросы

- 1 Цель диаграммы классов.
- 2 Стереотипы на диаграмме классов.
- 3 Виды связей между классами.

4 Лабораторная работа № 4. Разработка атрибутов, операций объектов и отношений между ними на диаграмме классов

Цель работы

Изучение процесса разработки диаграммы классов.

Теоретические сведения

Принятие соглашений по моделированию включает используемые диаграммы и элементы модели, правила их применения, соглашения по именованию

элементов, организацию модели (пакеты).

Пример соглашений моделирования.

Имена вариантов использования должны быть короткими глагольными фразами. Для каждого варианта использования должен быть создан пакет Use-Case Realization, включающий:

– по крайней мере одну реализацию варианта использования;

Имена классов должны быть существительными, соответствующими, по возможности, понятиям предметной области (гlossарию проекта).

Имена классов должны начинаться с заглавной буквы.

Имена атрибутов и операций должны начинаться со строчной буквы.

Составные имена должны быть сплошными, без подчеркиваний, каждое отдельное слово должно начинаться с заглавной буквы.

Реализация варианта использования (Use-Case Realization) описывает реализацию конкретного варианта использования в терминах взаимодействующих объектов и представляется с помощью набора диаграмм (диаграмм классов, реализующих вариант использования, и диаграмм взаимодействия (диаграмм последовательности), отражающих взаимодействие объектов в процессе реализации варианта использования).

Идентификация ключевых абстракций заключается в предварительном определении классов системы (классов анализа). Источники – знание предметной области, требования к системе, гlossарий. Классы анализа для системы регистрации показаны на рисунке 4.1.

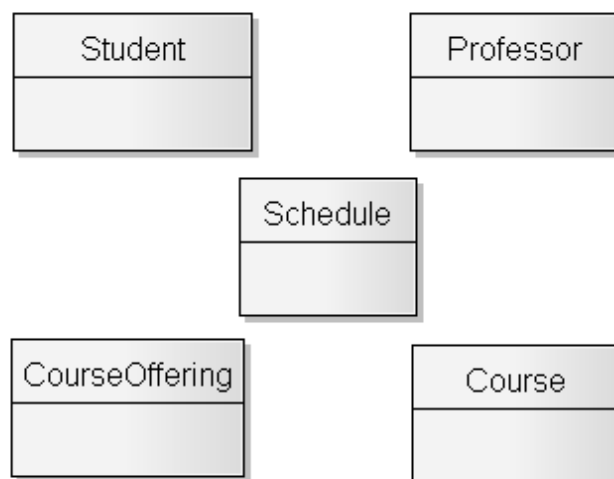


Рисунок 4.1 – Классы анализа системы регистрации

Создание структуры модели и классов анализа в соответствии с требованиями архитектурного анализа.

Создание пакетов и диаграммы Traceabilities.

Щелкните правой кнопкой мыши на логическом представлении браузера.

В открывшемся меню выберите пункт Add > Add Package.

Назовите новый пакет Design Model.

Создайте аналогичным образом пакеты Use-Case Realizations, Use-Case Realization – Close Registration, Use-Case Realization – Login и Use-Case Realization – Register for Courses.

Анализ вариантов использования.

Идентификация классов, участвующих в реализации потоков событий варианта использования. В потоках событий варианта использования выявляются классы трех типов.

Граничные классы (Boundary) служат посредниками при взаимодействии внешних объектов с системой. Как правило, для каждой пары «действующее лицо – вариант использования» определяется один граничный класс. Типы граничных классов: пользовательский интерфейс (обмен информацией с пользователем, без деталей интерфейса – кнопок, списков, окон), системный интерфейс и аппаратный интерфейс (используемые протоколы, без деталей их реализации).

Классы-сущности (Entity) представляют собой ключевые абстракции (понятия) разрабатываемой системы. Источники выявления классов-сущностей: ключевые абстракции, созданные в процессе архитектурного анализа, глоссарий, описание потоков событий вариантов использования.

Управляющие классы (Control) обеспечивают координацию поведения объектов в системе. Могут отсутствовать в некоторых вариантах использования, ограничивающихся простыми манипуляциями с хранимыми данными. Как правило, для каждого варианта использования определяется один управляющий класс. Примеры управляющих классов: менеджер транзакций, обработчик ошибок.

Пример набора классов, участвующих в реализации варианта использования Register for Courses, приведен на рисунке 4.2.

Задание для самостоятельного выполнения

Разработать на диаграмме классов соответствующие классы по индивидуальному заданию.

Контрольные вопросы

- 1 Цель диаграммы.
- 2 Стереотипы классов.
- 3 Связи между классами.

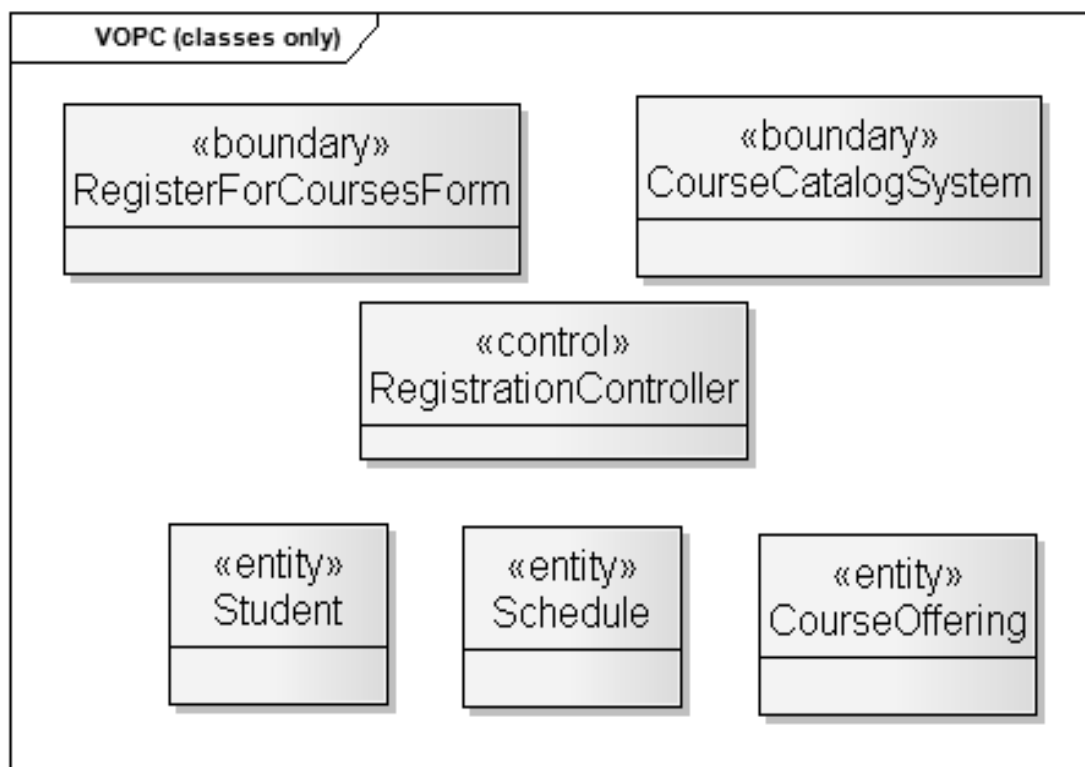


Рисунок 4.2 – Классы, участвующие в реализации варианта использования

5 Лабораторная работа № 5. Разработка диаграммы кооперации и редактирование свойств ее элементов

Цель работы

Изучение последовательности разработки кооперативной диаграммы.

Теоретические сведения

Создание кооперативной диаграммы.

На этой диаграмме представлена вся та информация, которая была и на диаграмме последовательности, но кооперативная диаграмма описывает поток событий в свете отношения между объектами.

На рисунке 5.1 представлена кооперативная диаграмма.

Задание для самостоятельного выполнения

Разработать кооперативную диаграмму по индивидуальному заданию.

Контрольные вопросы

- 1 Цель кооперативной диаграммы.
- 2 Сообщения и их виды.

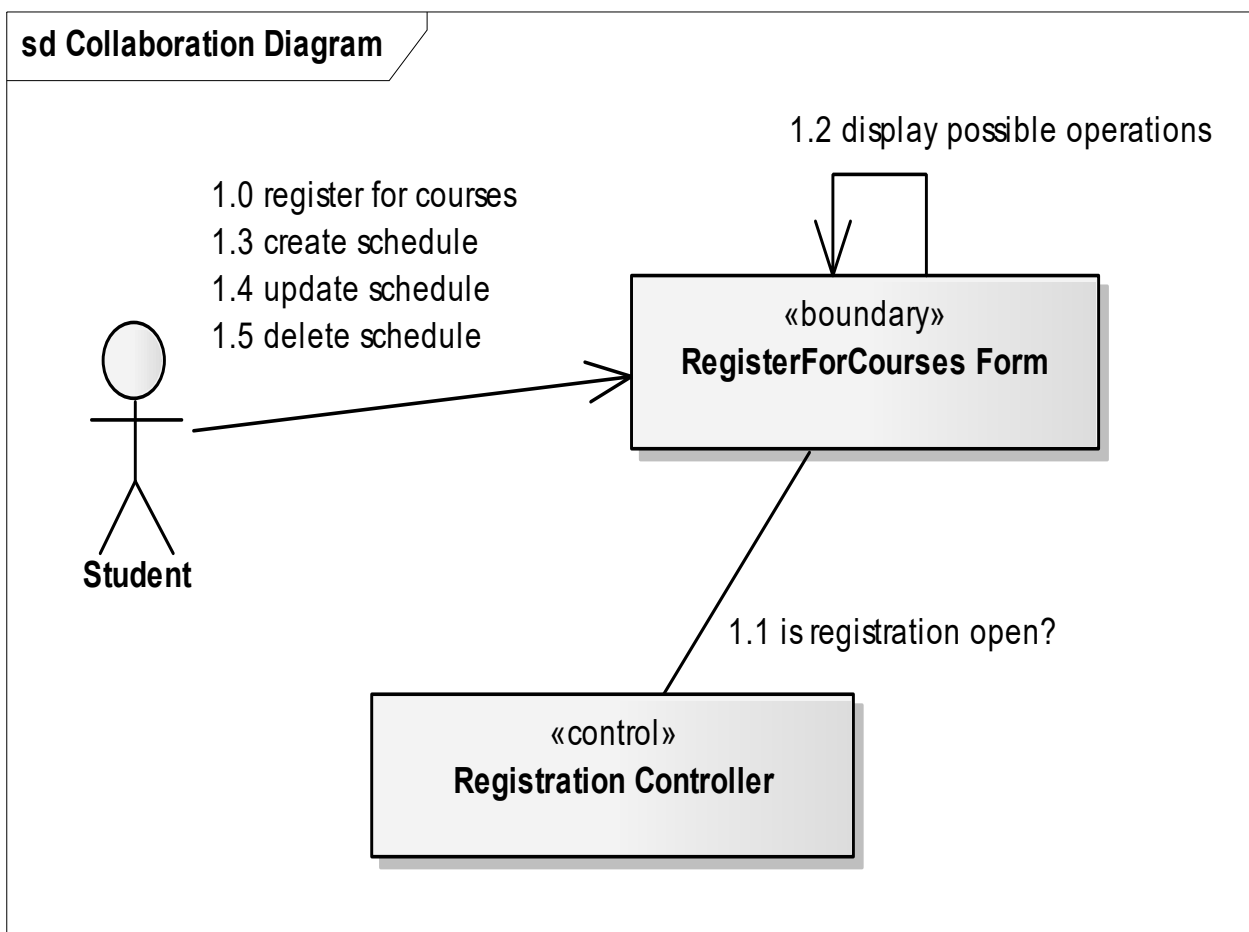


Рисунок 5.1 – Кооперативная диаграмма Register for Courses – Basic Flow

6 Лабораторная работа № 6. Разработка диаграммы последовательности и редактирование свойств ее элементов

Цель работы

Изучение последовательности разработки диаграммы последовательности.

Теоретические сведения

Создание диаграмм взаимодействия.

Создайте диаграммы последовательности и кооперативные диаграммы для основного потока событий варианта использования Register for Courses.

Готовые диаграммы последовательности должны иметь вид, как на рисунках 6.1 и 6.2.

Задание для самостоятельного выполнения

Разработать диаграммы последовательности по индивидуальному заданию.

Контрольные вопросы

- 1 Цель диаграммы последовательности.
- 2 Объекты, линии жизни объектов.
- 3 Сообщения и их виды.

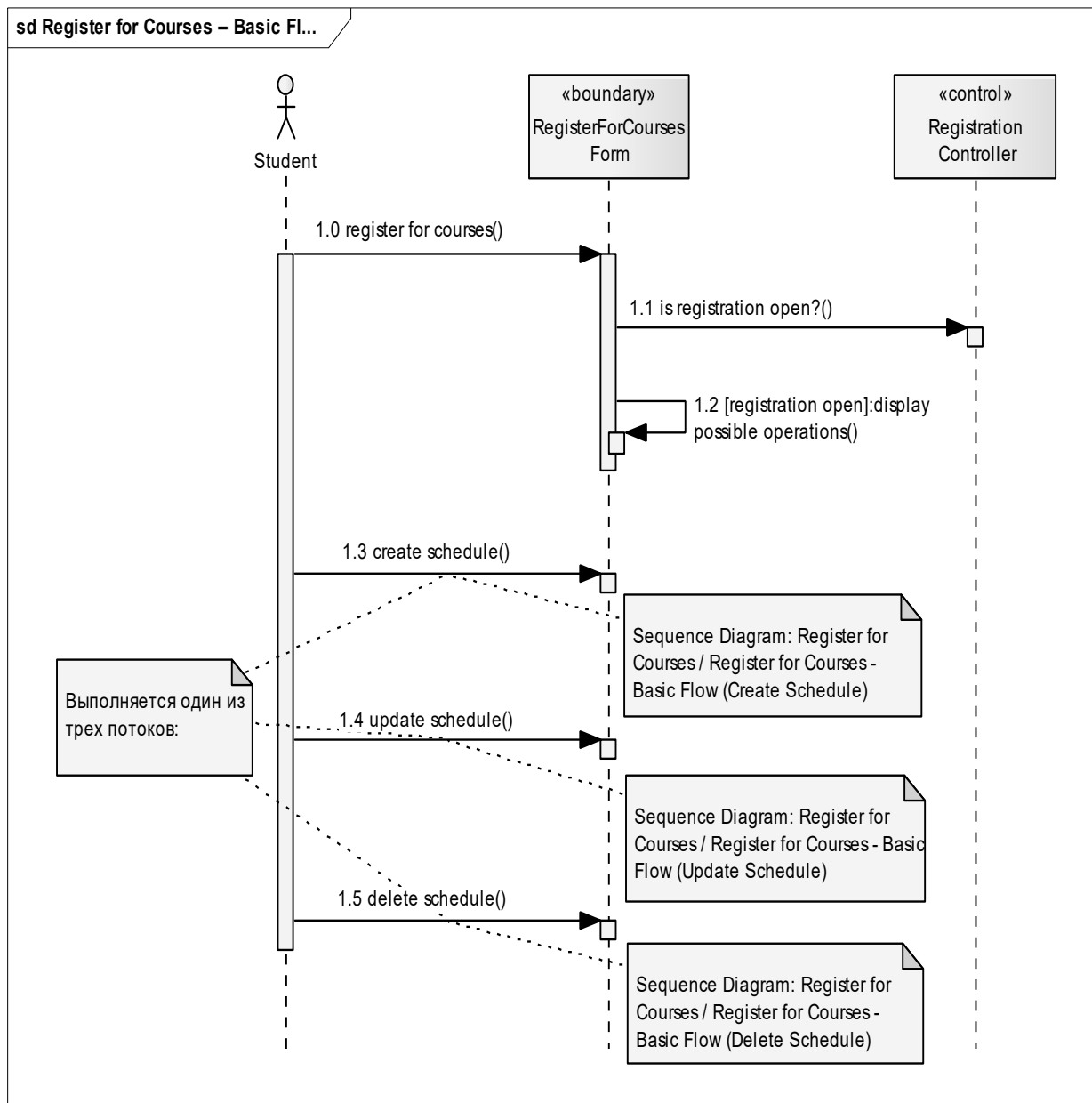


Рисунок 6.1 – Диаграмма последовательности Register for Courses – Basic Flow

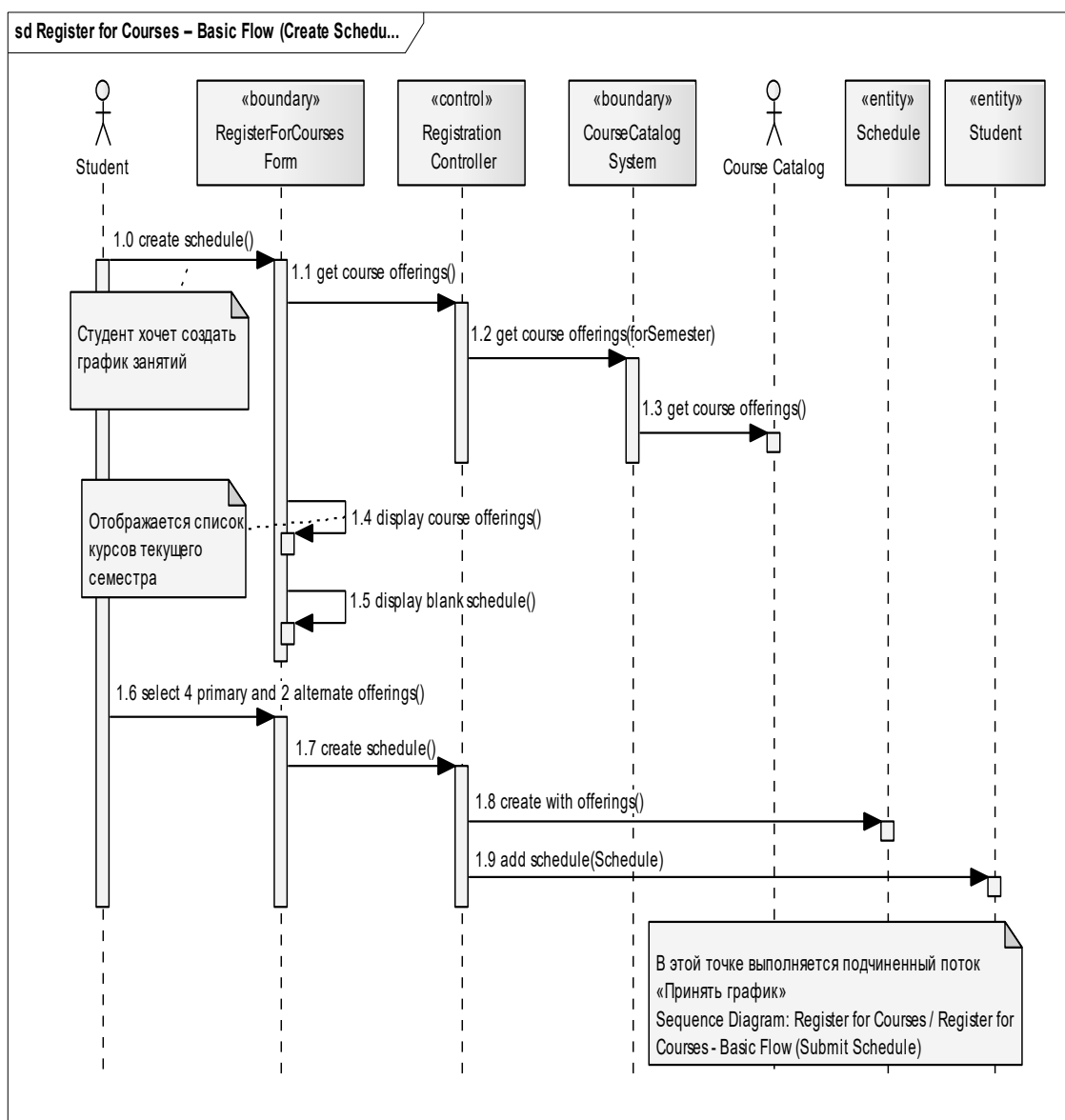


Рисунок 6.2 – Диаграмма последовательности Register for Courses

7 Лабораторная работа № 7. Разработка диаграммы состояний и редактирование свойств ее элементов

Цель работы

Изучение технологии создания диаграммы состояний.

Теоретические сведения

Определение состояний для классов моделируется с помощью диаграмм состояний.

Диаграммы состояний создаются для описания объектов с высоким уровнем динамического поведения.

В качестве примера рассмотрим поведение объекта класса CourseOffering.

Он может находиться в открытом состоянии (возможно добавление нового студента) или в закрытом состоянии (максимальное количество студентов уже записалось на курс). Таким образом, конкретное состояние зависит от количества студентов, связанных с объектом CourseOffering. Рассматривая каждый вариант использования, можно выделить еще два состояния: инициализация (до начала регистрации студентов на курс) и отмена (курс исключается из расписания) (рисунок 7.1).

Задание для самостоятельного выполнения

Разработать диаграмму состояний АСОИУ.

Контрольные вопросы

- 1 Цель диаграммы состояний.
- 2 Понятие состояний.
- 3 Деятельность. Входное и выходное действие.
- 4 Изображение состояний и событий на диаграмме.
- 5 Ограничивающие условия.

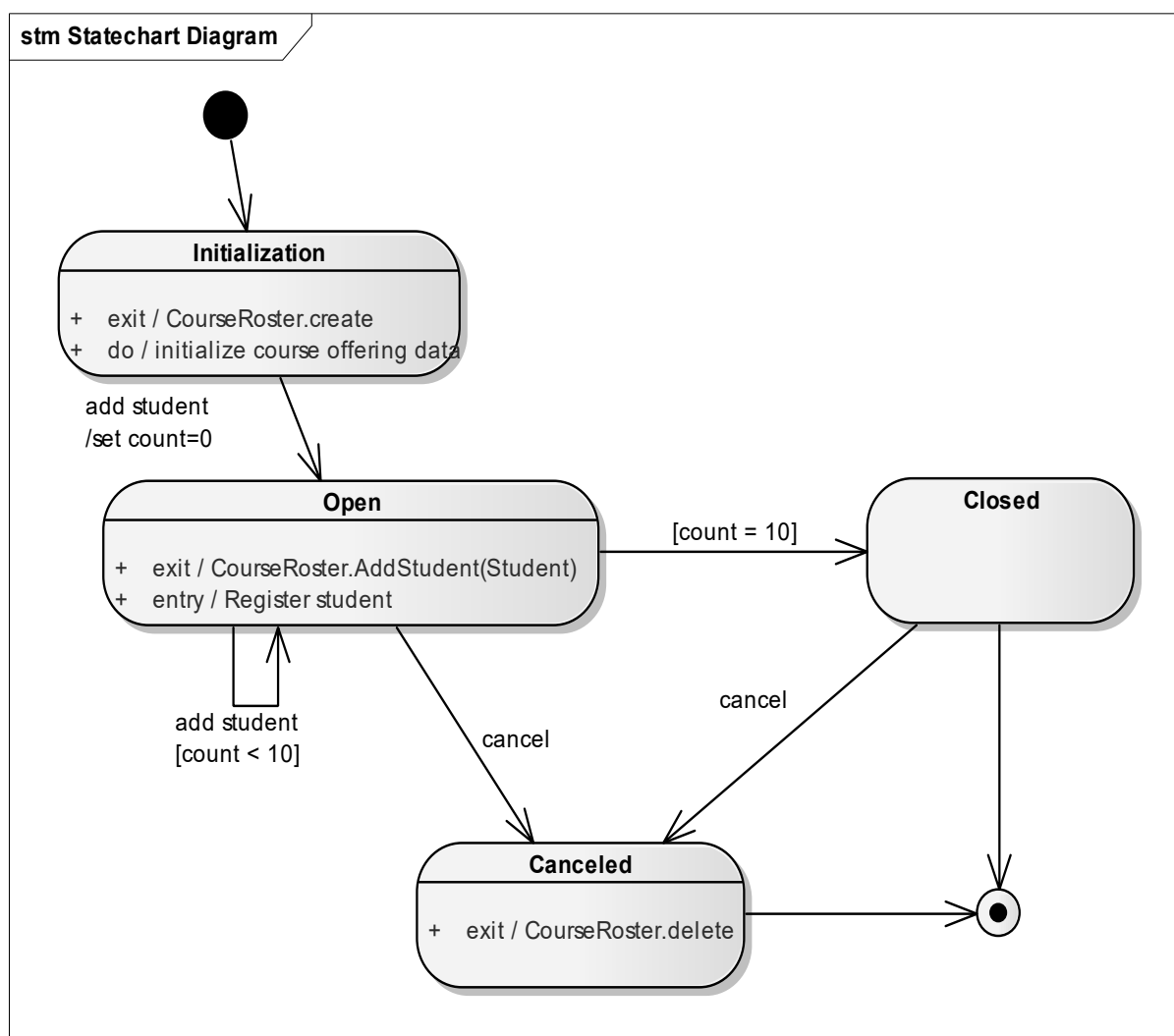


Рисунок 7.1 – Диаграмма состояний для класса CourseOffering

8 Лабораторная работа № 8. Разработка диаграммы деятельности и редактирование свойств ее элементов

Цель работы

Изучение технологии разработки диаграммы деятельности.

Теоретические сведения

При моделировании поведения проектируемой или анализируемой системы возникает необходимость не только представить процесс изменения ее состояний, но и детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Традиционно для этой цели использовались блок-схемы или структурные схемы алгоритмов. Каждая такая схема акцентирует внимание на последовательности выполнения определенных действий или элементарных операций, которые в совокупности приводят к получению желаемого результата.

Состояние действия (action state) является специальным случаем состояния с некоторым входным действием и, по крайней мере одним выходящим из состояния переходом. Этот переход неявно предполагает, что входное действие уже завершилось. Состояние действия не может иметь внутренних переходов, поскольку оно является элементарным. Обычное использование состояния действия заключается в моделировании одного шага выполнения алгоритма (процедуры) или потока управления.

Графически состояние действия изображается фигурой, напоминающей прямоугольник, боковые стороны которого заменены выпуклыми дугами. Внутри этой фигуры записывается выражение действия (action-expression), которое должно быть уникальным в пределах одной диаграммы деятельности.

Действие может быть записано на естественном языке, некотором псевдокоде или языке программирования. Никаких дополнительных или неявных ограничений при записи действий не накладывается. Рекомендуется в качестве имени простого действия использовать глагол с пояснительными словами. Если же действие может быть представлено в некотором формальном виде, то целесообразно записать его на том языке программирования, на котором предполагается реализовывать конкретный проект.

Фрагмент диаграммы деятельности для алгоритма нахождения корней квадратного уравнения представлен на рисунке 8.1.

Задание для самостоятельного выполнения

Разработать диаграмму деятельности для вычислительных классов.

Контрольные вопросы

- 1 Цель диаграммы деятельности.
- 2 Понятие деятельности.
- 3 Деятельность. Входное и выходное действие.
- 4 Изображение деятельности на диаграмме.
- 5 Как отображается разделение и слияние параллельных потоков деятельности.

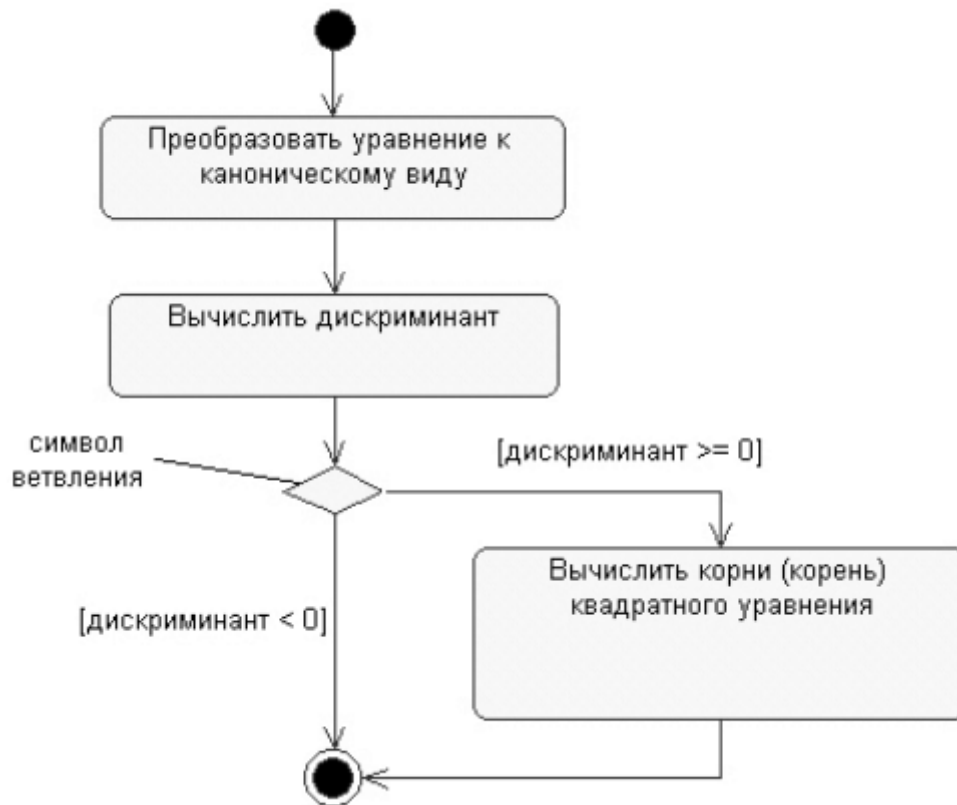


Рисунок 8.1 – Фрагмент диаграммы деятельности

9 Лабораторная работа № 9. Разработка диаграммы компонентов и редактирование свойств ее элементов

Цель работы

Изучение технологии разработки диаграммы компонентов.

Теоретические сведения

В Enterprise Architect диаграммы компонентов создаются в представлении компонентов системы. Отдельные компоненты можно создавать непосредственно на диаграмме или перетаскивать их туда из браузера.

Создание диаграммы компонентов.

Диаграмма компонентов показывает части программного обеспечения,

встроенных контроллеров, их организацию и зависимость. Диаграмма компонентов имеет более высокий уровень абстракции, чем диаграмма класса, обычно компонент осуществляется одним или более классами (или объектами) во время выполнения. Они являются структурными элементами, построенными так, что в конечном счете один из компонентов может охватить значительную часть системы.

Компонент является модульной частью системы, чье поведение определяется его provided и required интерфейсами; внутреннее функционирование компонента должно быть невидимым, а его использование независимым от окружающей среды.

Создание компонента.

Дважды щелкните мышью на главной диаграмме компонентов в представлении компонентов.

Нажмите на панели инструментов кнопку Component.

Щелкните мышью в том месте диаграммы, куда собираетесь поместить компонент.

В открывшемся окне введите название компонента, выберите его стереотип, язык программирования.

Задание для самостоятельного выполнения

Разработать диаграмму компонентов АСОИУ.

Контрольные вопросы

- 1 Цель диаграммы.
- 2 Типы компонентов.
- 3 Связи между компонентами.

10 Лабораторная работа № 10. Разработка диаграммы развертывания и редактирование свойств ее элементов

Цель работы

Изучение технологии разработки диаграммы размещения.

Теоретические сведения

Распределенная конфигурация системы моделируется с помощью диаграммы размещения. Ее основные элементы:

- узел (node) – вычислительный ресурс (процессор или другое устройство (дисковая память, контроллеры различных устройств и т. д.). Для узла можно задать выполняющиеся на нем процессы;
- соединение (connection) – канал взаимодействия узлов (сеть). Пример: сетевая конфигурация системы регистрации (рисунок 10.1).

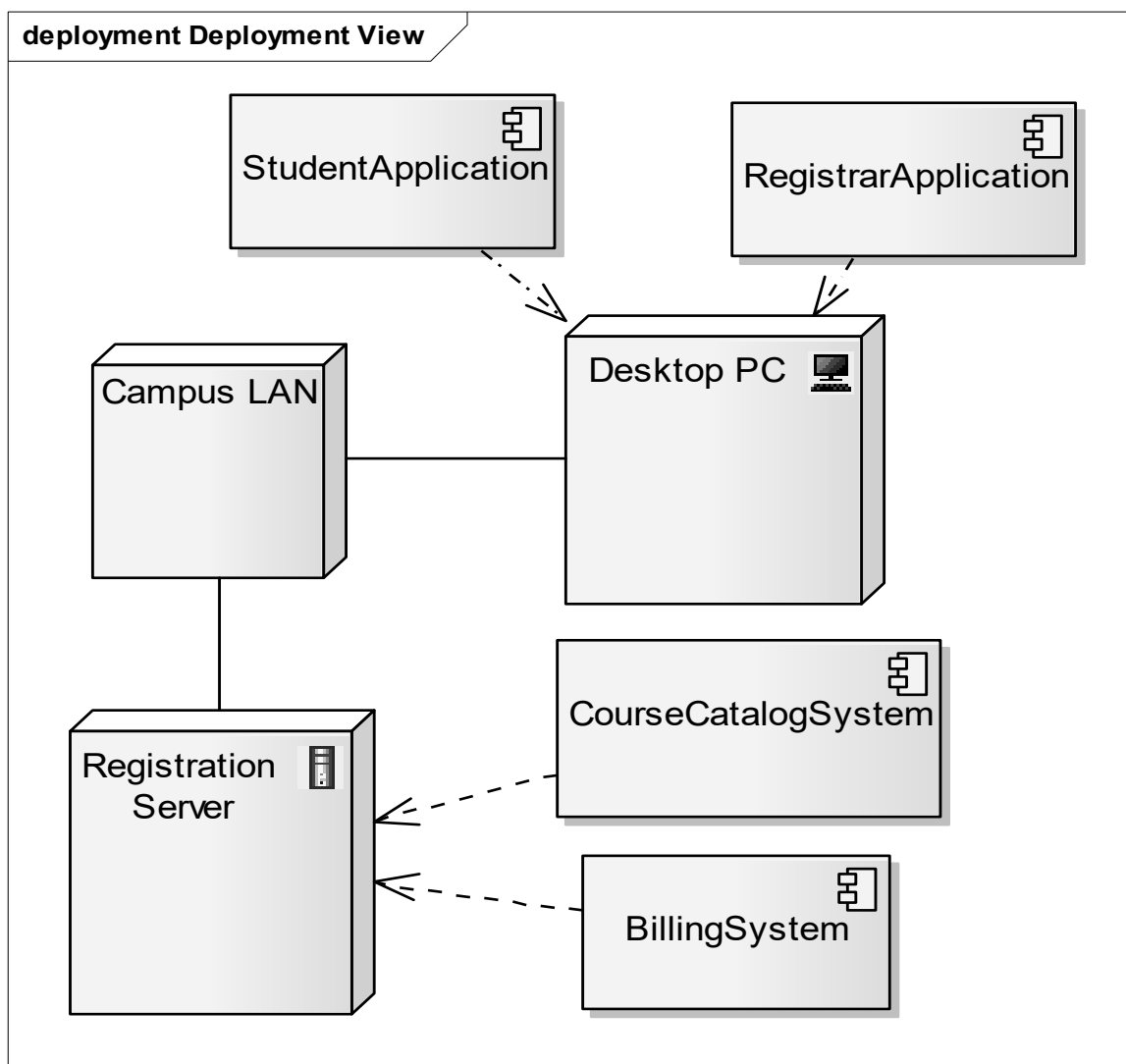


Рисунок 10.1 – Сетевая конфигурация системы регистрации

Задание для самостоятельного выполнения

Разработать диаграмму развертывания.

Контрольные вопросы

- 1 Цель диаграммы размещения.
- 2 Изображение основных элементов диаграммы размещения.

Список литературы

- 1 Using Enterprise Architect - UML Modeling Tool [Электронный ресурс]. – Режим доступа: <https://sparxsystems.com>. – Дата доступа: 05.05.2023.
- 2 Хугагуров, Я. А. Проектирование автоматизированных систем обработки информации и управления (АСОИУ)/ Я. А. Хачатуров. – Москва: Лаборатория знаний, 2020. – 243 с.