

ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ПРОЕКТИРОВАНИЕ АСОИУ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
09.03.01 «Информатика и вычислительная техника»
дневной формы обучения*



Могилев 2018

УДК 004.7
ББК 32.973-02
П 79

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»
«4» сентября 2018 г., протокол № 2

Составители: ст. преподаватель Е. А. Зайченко;
доц. С. К. Крутолевич

Рецензент доц. А. П. Прудников

Методические рекомендации предназначены для студентов направления
подготовки 09.03.01 «Информатика и вычислительная техника» дневной фор-
мы обучения.

Учебно-методическое издание

ПРОЕКТИРОВАНИЕ АСОИУ

Ответственный за выпуск	А. И. Якимов
Технический редактор	С. Н. Красовская
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 31 экз. Заказ №

Издатель и полиграфическое исполнение:
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.
Пр. Мира, 43, 212000, Могилев.

© ГУ ВПО «Белорусско-Российский
университет», 2018



Содержание

Введение.....	4
Общие требования к отчету	5
1 Лабораторная работа № 1. Общая характеристика CASE-средства Enterprise Architect.....	5
2 Лабораторная работа № 2. Разработка диаграммы вариантов использования.....	9
3 Лабораторная работа № 3. Разработка диаграммы классов на уровне сущностей.....	16
4 Лабораторная работа № 4. Разработка атрибутов, операций объектов и отношений между ними на диаграмме классов.....	20
5 Лабораторная работа № 5. Разработка диаграммы кооперации и редактирование свойств ее элементов	26
6 Лабораторная работа № 6. Разработка диаграммы последовательности и редактирование свойств ее элементов	27
7 Лабораторная работа № 7. Разработка диаграммы состояний и редактирование свойств ее элементов	31
8 Лабораторная работа № 8. Разработка диаграммы деятельности и редактирование свойств ее элементов	33
9 Лабораторная работа № 9. Разработка диаграммы компонентов и редактирование свойств ее элементов	36
10 Лабораторная работа № 10. Разработка диаграммы размещения и редактирование свойств ее элементов	37
11 Лабораторная работа № 11. Определение трудоемкости разработки АСОИ.....	38
Список литературы	40



Введение

Целью дисциплины «Проектирование автоматизированных систем» является приобретение специальных знаний, умений и навыков, необходимых инженеру по информационным технологиям в процессе проектирования автоматизированных систем обработки информации и управления (АСОИУ).

В результате изучения дисциплины студент должен знать:

- современные концептуальные, теоретические и прикладные аспекты проектирования АСОИУ;
- современные системы автоматизированного проектирования АСОИУ;
- прикладное программное обеспечение в корпоративных системах обработки данных.

Данные методические рекомендации содержат описание 11 лабораторных работ, выполнение которых закрепляет материал, изложенный в курсе «Проектирование АСОИУ».

Для выполнения лабораторных работ необходимо согласовать с преподавателем тему АСОИУ для индивидуальной разработки.

Студенту необходимо, используя данный материал как основу, разработать диаграммы своей АСОИУ.



Общие требования к отчету

Отчет должен содержать стандартные составные части.

1 Титульный лист с указанием следующих реквизитов: название учреждения образования, название закрепленной за дисциплиной кафедры, номер и название лабораторной работы, название дисциплины, вариант, кто выполнил лабораторную работу (ФИО, группа), кто проверяет работу (должность, ФИО), место и дата составления отчета.

- 2 Цель работы.
- 3 Постановка задачи.
- 4 Выполненное задание.
- 5 Выводы по теме лабораторной работы.

Отчет оформляется шрифтом гарнитуры Times New Roman, кегль 14, междустрочный интервал – полуторный, абзацный отступ – 1,25 см.

Страницы должны быть пронумерованы вверху посередине. Титульный лист при нумерации считается, но не нумеруется.

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

Цель работы: изучение основных возможностей и режимов работы CASE-средства Enterprise Architect.

Теоретический материал

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 (рисунок 1.1).

Эта страница предоставляет следующие возможности:

Search выполняет 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 ис-



пользуется для открытия существующего проекта.



Рисунок 1.1 – Стартовая страница Enterprise Architect

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 (рисунок 1.2). Можно перетаскивать элементы из папки в папку или из браузера прямо на диаграмму.

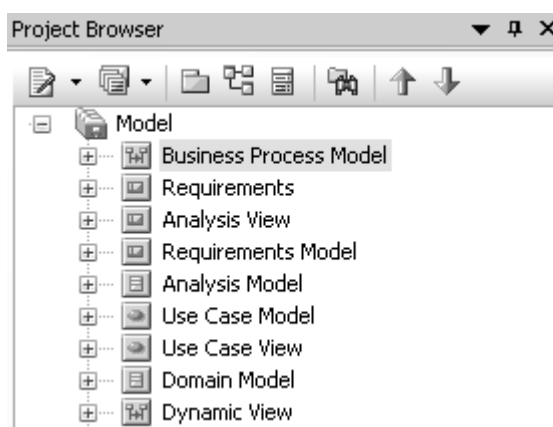


Рисунок 1.2 – Браузер проекта

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 (рисунок 1.3).

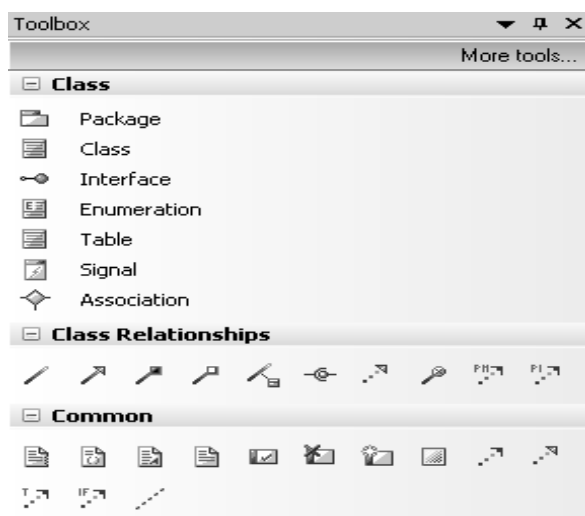


Рисунок 1.3 – Панель инструментов для диаграммы классов

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.

Главное меню Enterprise Architect (рисунок 1.4) обеспечивает управляемый мышью доступ ко многим функциям, связанным с жизненным циклом проекта наряду с функциями администрирования.



Рисунок 1.4 – Главное меню Enterprise Architect

Меню инструментов (Tools Menu) обеспечивает доступ к различным настройкам, общим для генерации кода, управления ЕАР (рисунок 1.5). файлами, настройки правописания, внешних ресурсов, настройки таких функций, как управление ярлыками.

В меню инструментов наибольший интерес представляет пункт Options. Здесь находятся опции настройки Enterprise Architect для отображения и работы с

моделями и элементами модели. Основные режимы представлены в таблице 1.1.

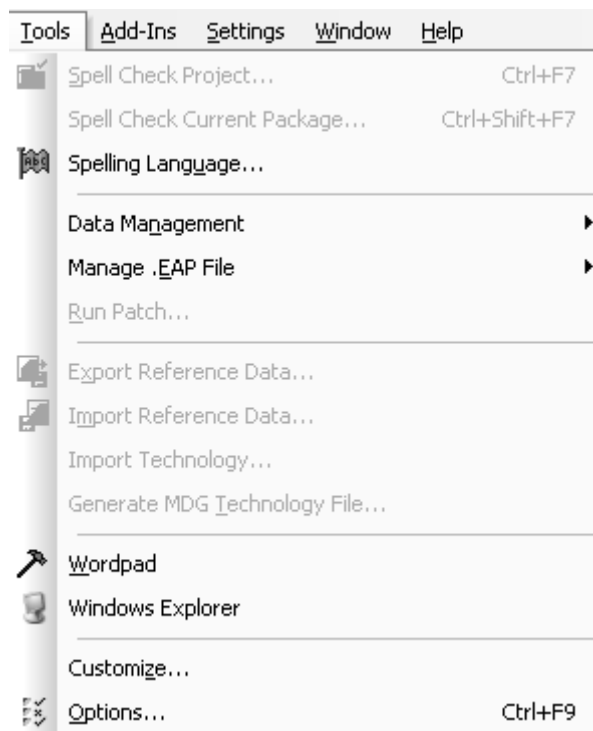


Рисунок 1.5 – Меню инструментов (Tools Menu)

Таблица 1.1 – Назначение вкладок пункта меню Options

Вкладка	Назначение вкладки
General	Общие настройки проекта, например, автор, адрес домашнего веб-сайта, общие настройки браузера проекта
Standard Colors	Позволяет установить цвет ряда объектов и их фон
New Diagram Defaults	Позволяет конфигурировать опции для новых диаграмм и общего поведения диаграммы
Diagram Appearance	Позволяет определить, как диаграммы и их содержимое отображается на экране
Diagram Behavior	Позволяет определить, как диаграмма реагирует на действия, производимые над ней
Diagram Sequence	Позволяет установить настройки шрифта и фокус контрольного индикатора для диаграммы последовательности
Objects	Позволяет установить, как элементы выглядят на диаграмме
Links	Настройки создания, поведения и нотации связей
XML Specifications	Позволяет установить настройки для работы с XML
Communication Colors	Позволяет установить цвета, используемые в диаграммах взаимодействия
Source Code Engineering	Описывает общие настройки, применяемые для всех языков при генерации кода из Enterprise Architect.
C++ (рисунок 1.6)	Установка опций для генерации кода на C++ включает: установка опций для текущей модели; установка опций для текущего пользователя

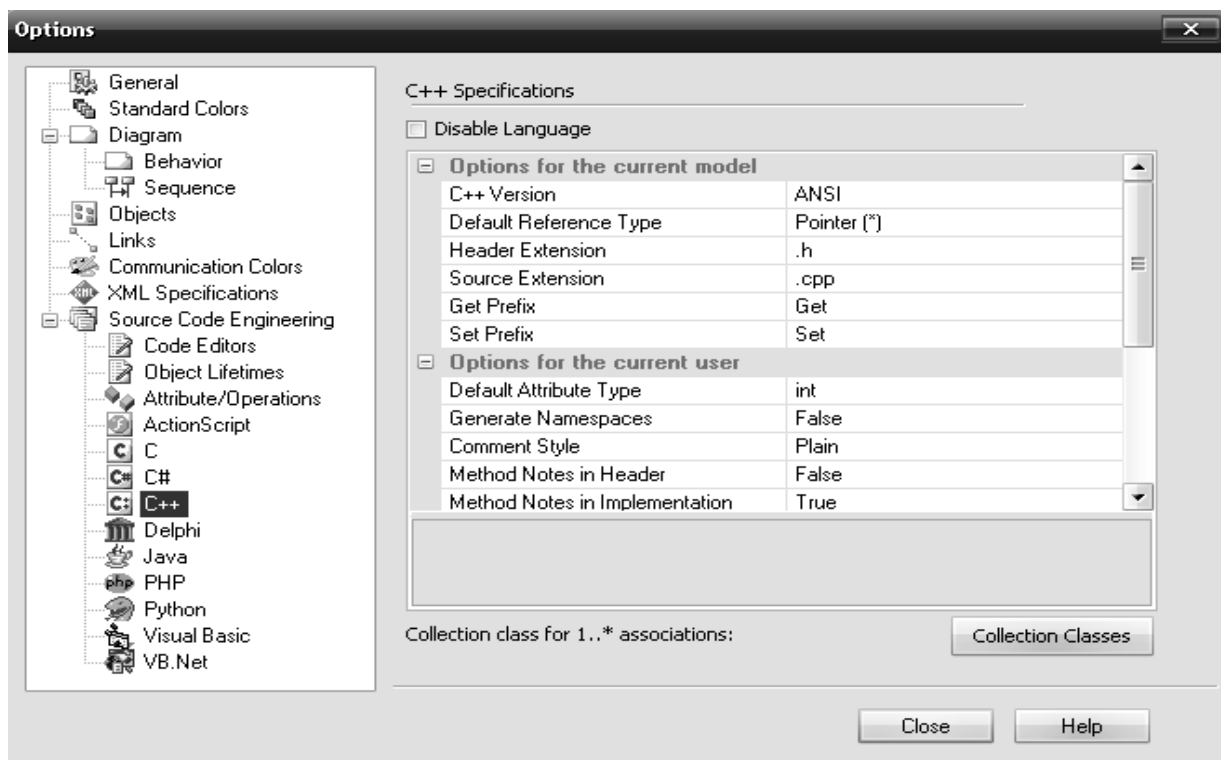


Рисунок 1.6 – Вкладка C++ Specifications

Практическое задание

Освоить приемы работы в среде Enterprise Architect.

Контрольные вопросы

- 1 Перечислить основные диаграммы для моделирования АСОИ.
- 2 Перечислить основные элементы экрана.
- 3 Дать характеристику Браузеру.
- 4 Дать характеристику Окну диаграмм.
- 5 Дать характеристику четырем представлениям модели.

2 Лабораторная работа № 2. Разработка диаграммы вариантов использования

Цель работы: изучение технологии формирования диаграммы вариантов использования.

Теоретический материал

Создание действующих лиц в среде Enterprise Architect

Чтобы поместить действующее лицо в браузер:

– щелкните левой кнопкой мыши на панели инструментов Toolbox по

кнопке Actor;

– щелкните по рабочей области диаграммы, в появившемся окне введите имя актера, выберите стереотип. В браузере появится новое действующее лицо. Слева от его имени вы увидите пиктограмму действующего лица UML. После создания действующих лиц сохраните модель под именем courserereg(analysis) с помощью пункта меню File > Save Project As.

Варианты использования.

Исходя из потребностей действующих лиц, выделяются следующие варианты использования:

- Login (Войти в систему);
- Register for Courses (Зарегистрироваться на курсы);
- View Report Card (Просмотреть табель 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).

Создание вариантов использования в среде Enterprise Architect.

Чтобы поместить вариант использования в браузер:

– щелкните левой кнопкой мыши на панели инструментов Toolbox по кнопке Use Case;

– щелкните по рабочей области диаграммы, в появившемся окне введите название варианта использования, выберите его стереотип.

Новый вариант использования появится в браузере. Слева от него будет видна пиктограмма варианта использования UML.

Диаграмма вариантов использования. Создайте диаграмму вариантов использования для системы регистрации. Требуемые для этого действия подробно перечислены далее. Готовая диаграмма вариантов использования должна выглядеть как на рисунке 2.1.

В среде Enterprise Architect диаграммы вариантов использования создаются в представлении вариантов использования. Главная диаграмма предлагается по умолчанию. Для моделирования системы можно затем разработать столько дополнительных диаграмм, сколько необходимо.

Чтобы получить доступ к главной диаграмме вариантов использования:

– рядом с представлением вариантов использования в браузере щелкните на значке « + », это приведет к открытию данного представления;

– дважды щелкните на главной диаграмме, чтобы открыть её.

Для создания новой диаграммы вариантов использования:

– щелкните правой кнопкой мыши на пакете представления вариантов использования в браузере;

– из всплывающего меню выберите пункт Add > Add Diagram;

– выделив новую диаграмму, введите ее имя;

– дважды щелкните на названии этой диаграммы в браузере, чтобы открыть ее.



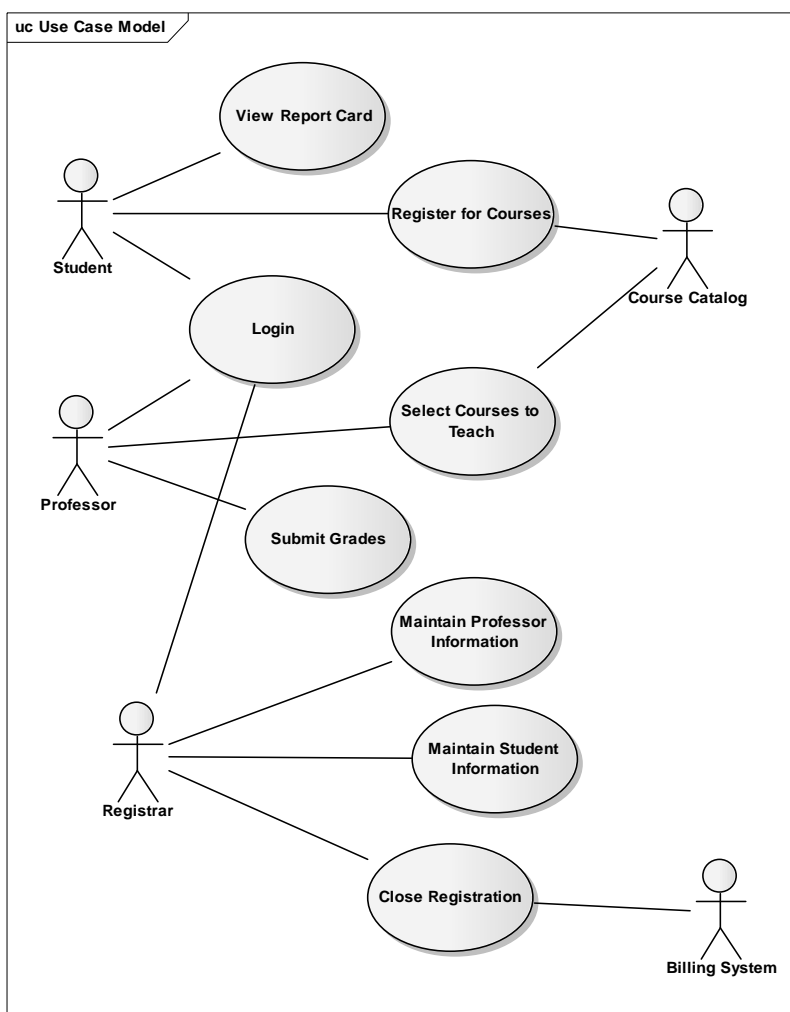


Рисунок 2.1 – Диаграмма вариантов использования

Построение диаграммы вариантов использования.

Откройте диаграмму вариантов использования. Чтобы поместить действующее лицо или вариант использования на диаграмму, перетащите его мышью из браузера на диаграмму вариантов использования.

Наличие общего варианта использования Login для трех действующих лиц позволяет обобщить их поведение и ввести новое действующее лицо Any User. Модифицированная диаграмма вариантов использования показана на рисунке 2.2.

Добавление описаний к вариантам использования.

Щелкните правой кнопкой мыши по варианту использования «Register for Courses».

Выберите Add > Note, введите следующее описание к этому варианту использования: «This use case allows a student to register for courses in the current semester» (Этот вариант использования дает студенту возможность зарегистрироваться на курсы в текущем семестре).

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 (Зарегистриро-

ваться на курсы) и Close Registration (Закрывать регистрацию).

Вариант использования Login.

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

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

Система запрашивает имя пользователя и пароль.

Пользователь вводит имя и пароль.

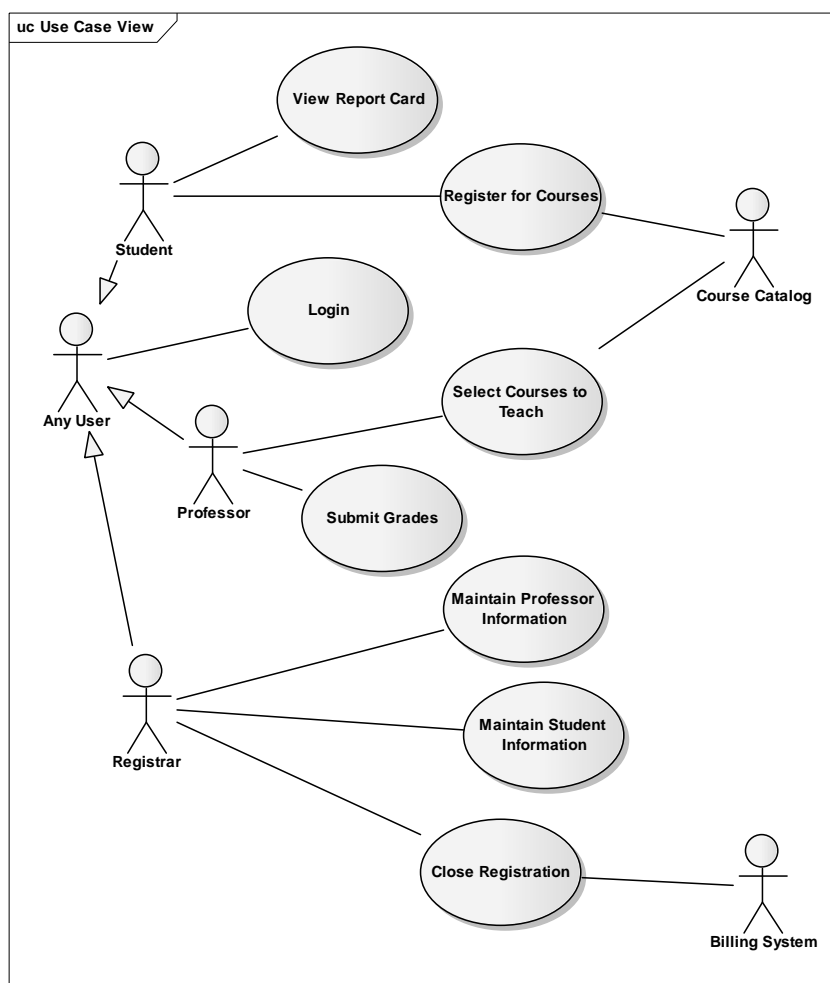


Рисунок 2.2 – Модифицированная диаграмма вариантов использования

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 начинается, 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).

Создать график:

- система выполняет поиск в каталоге курсов доступных конкретных курсов и выводит их список;
- студент выбирает из списка четыре основных курса 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 подтверждения 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.

Принять график.

Для каждого выбранного, но еще не «зафиксированного» конкретного курса в графике система проверяет выполнение студентом предварительных требований (прохождение определенных курсов), факт открытия конкретного курса и отсутствие конфликтов графика. Затем система добавляет студента в список выбранного конкретного курса. Курс фиксируется в графике и график сохраняется в системе.

Альтернативные потоки. Сохранить график.

В любой момент студент может вместо принятия графика сохранить его. В этом случае шаг «Принять график» заменяется на следующий.

«Незафиксированные» конкретные курсы помечаются в графике как «выбранные».



График сохраняется в системе.

Не выполнены предварительные требования, курс заполнен или имеют место конфликты графика. Если во время выполнения подчиненного потока «Принять график» система обнаружит, что студент не выполнил необходимые предварительные требования, или выбранный им конкретный курс заполнен, или имеют место конфликты графика, то выдается сообщение об ошибке. Студент может либо выбрать другой конкретный курс и продолжить выполнение варианта использования, либо сохранить график, либо отменить операцию, после чего основной поток начнется сначала.

График не найден. Если во время выполнения подчиненных потоков «Обновить график» или «Удалить график» система не может найти график студента, то выдается сообщение об ошибке. После того, как студент подтвердит это сообщение, основной поток начнется сначала.

Система каталога курсов недоступна. Если окажется, что невозможно установить связь с системой каталога курсов, то будет выдано сообщение об ошибке. После того, как студент подтвердит это сообщение, вариант использования завершится.

Регистрация на курсы закончена. Если в самом начале выполнения варианта использования окажется, что регистрация на текущий семестр закончена, будет выдано сообщение и вариант использования завершится.

Удаление отменено. Если во время выполнения подчиненного потока «Удалить график» студент решит не удалять его, удаление отменяется, и основной поток начнется сначала.

Предусловия. Перед началом выполнения данного варианта использования студент должен войти в систему.

Постусловия. Если вариант использования завершится успешно, график студента будет создан, обновлен или удален. В противном случае состояние системы не изменится.

Вариант использования Close Registration.

Краткое описание. Данный вариант использования позволяет регистратору закрывать процесс регистрации. Конкретные курсы, на которые не записалось достаточного количества студентов (менее трех), отменяются. В расчетную систему передается информация о каждом студенте по каждому конкретному курсу, чтобы студенты могли внести оплату за курсы.

Основной поток событий. Данный вариант использования начинает выполняться, когда регистратор запрашивает прекращение регистрации.

Система проверяет состояние процесса регистрации. Если регистрация еще выполняется, выдается сообщение и вариант использования завершается.

Для каждого конкретного курса система проверяет, ведет ли его какой-либо профессор, и записалось ли на него не менее трех студентов. Если эти условия выполняются, система фиксирует конкретный курс в каждом графике, который включает данный курс.

Для каждого студенческого графика проверяется наличие в нем максимального количества основных курсов; если их недостаточно, система пытается дополнить альтернативными курсами из списка данного графика. Выбирается



первый доступный альтернативный курс. Если таких курсов нет, то никакое дополнение не происходит.

Система закрывает все конкретные курсы. Если в каком-либо конкретном курсе оказывается менее трех студентов (с учетом добавлений, сделанных в п. 3), система отменяет его и исключает из каждого содержащего его графика.

Система рассчитывает плату за обучение для каждого студента в текущем семестре и направляет информацию в расчетную систему. Расчетная система посылает студентам счета для оплаты с копией их окончательных графиков.

Альтернативные потоки. Конкретный курс никто не ведет. Если во время выполнения основного потока обнаруживается, что некоторый конкретный не ведется никаким профессором, то этот курс отменяется. Система исключает данный курс из каждого содержащего его графика.

Расчетная система недоступна. Если невозможно установить связь с расчетной системой, через некоторое установленное время система вновь попытается связаться с ней. Попытки будут повторяться до тех пор, пока связь не установится.

Предусловия. Перед началом выполнения данного варианта использования регистратор должен войти в систему.

Постусловия. Если вариант использования завершится успешно, регистрация закрывается. В противном случае состояние системы не изменится.

Прикрепление файла к варианту использования.

Щелкните правой кнопкой мыши на варианте использования.

В открывшемся меню выберите пункт Properties.

Перейдите на вкладку Files.

Укажите путь к ранее созданному файлу (File Path) и нажмите на кнопку Save, чтобы прикрепить файл к варианту использования.

Нажмите кнопку Launch, чтобы открыть прикрепленный файл.

Удаление вариантов использования и действующих лиц. Существует два способа удалить элемент модели – из одной диаграммы или из всей модели. Чтобы удалить элемент модели из диаграммы:

- выделите элемент на диаграмме;
- нажмите на клавишу Delete или нажмите сочетание клавиш CTRL+ D;
- обратите внимание, что, хотя элемент и удален с диаграммы, он остался в

браузере и на других диаграммах системы.

Чтобы удалить элемент из модели:

- выделите элемент на диаграмме;
- выберите пункт меню Delete.

Практическое задание

Разработать диаграммы вариантов использования по индивидуальному заданию.



Контрольные вопросы

- 1 Цель диаграммы.
- 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 технологии формирования диаграммы классов.

Теоретический материал

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;
- диаграмму «View Of Participating Classes» (VOPC).

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 (рисунок 3.1), и диаграмм взаимодействия (диаграмм последовательности), отражающих взаимодействие объектов в процессе реализации варианта использования).

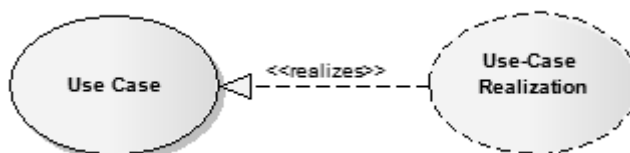


Рисунок 3.1 – Реализация варианта использования

Идентификация ключевых абстракций заключается в предварительном определении классов системы (классов анализа). Источники – знание предмет-

ной области, требования к системе, глоссарий. Классы анализа для системы регистрации показаны на рисунке 3.2.

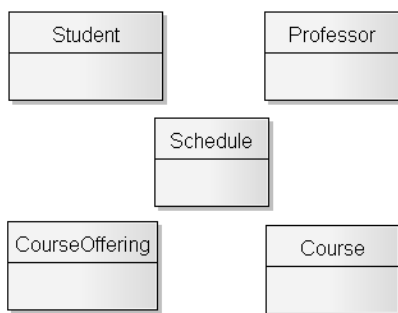


Рисунок 3.2 – Классы анализа системы регистрации

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

Структура логического представления браузера должна иметь следующий вид, показанный на рисунке 3.3.

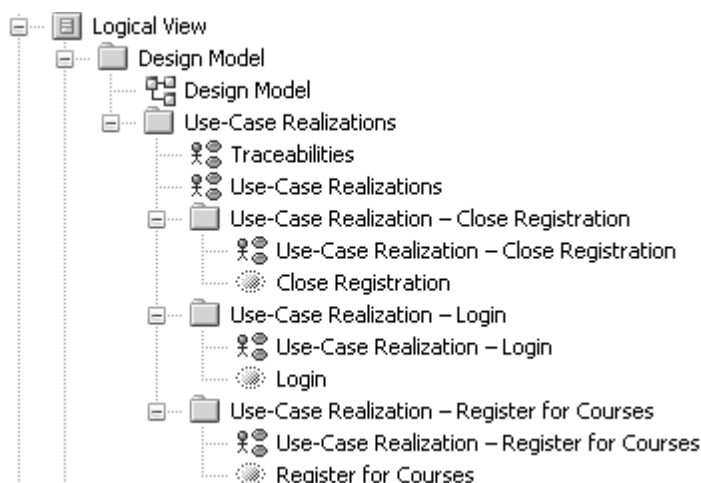


Рисунок 3.3 – Структура логического представления

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.

В каждом из пакетов типа Use-Case Realization создайте соответствующие кооперации Close Registration, Login и Register for Courses (каждая кооперация представляет собой вариант использования со стереотипом «use-case realization», который задается в спецификации варианта использования).

Создайте в пакете Use-Case Realizations новую диаграмму вариантов использования с названием Traceabilities и постройте ее в соответствии с рисунком 3.4.

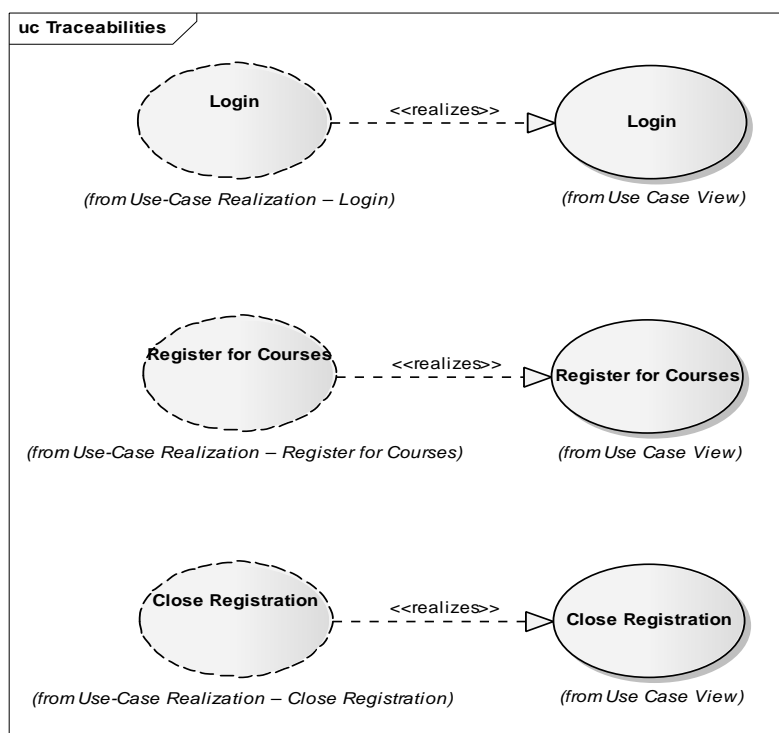


Рисунок 3.4 – Диаграмма Traceabilities

Анализ вариантов использования.

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

1) 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);

2) 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;

3) 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 3.5.

Создание классов, участвующих в реализации варианта использования Register for Courses, и диаграммы классов «View Of Participating Classes» (VOPC):
– щелкните правой кнопкой мыши на пакете Design Model;

- выберите в открывшемся меню пункт Add > Add Element, выберите тип Class;
- введите имя RegisterForCoursesForm;
- в открывшемся окне в поле стереотипа выберите Boundary и нажмите на кнопку ОК.

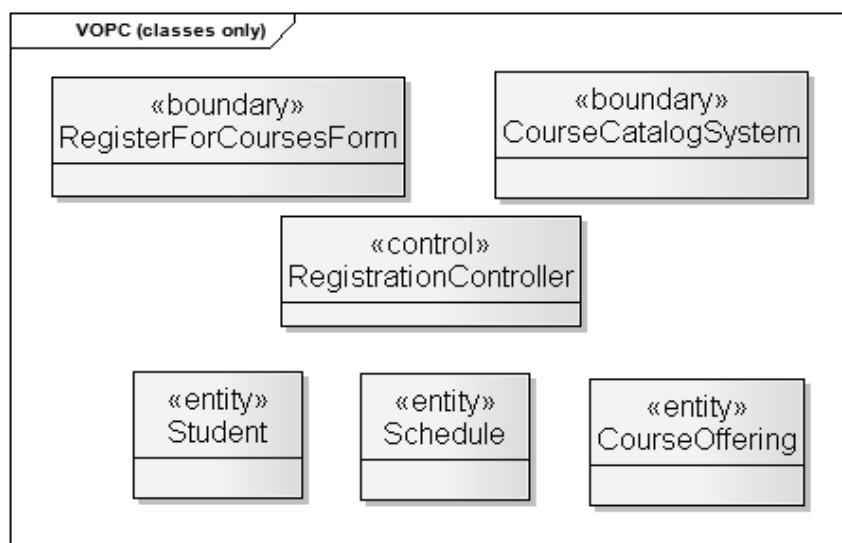


Рисунок 3.5 – Классы, участвующие в реализации варианта использования Register for Courses

Создайте аналогичным образом классы CourseCatalogSystem со стереотипом Boundary и RegistrationController со стереотипом Control.

Назначьте классам Schedule, CourseOffering и Student стереотип Entity.

Щелкните правой кнопкой мыши на кооперации Register for Courses в пакете Use-Case Realization – Register for Courses.

В открывшемся меню выберите пункт Add > Add Diagram > Class.

Назовите новую диаграмму классов VOPC (classes only).

Откройте ее и перетащите классы на открытую диаграмму в соответствии с рисунком 3.5.

Распределение поведения, реализуемого вариантом использования, между классами реализуется с помощью диаграмм взаимодействия (диаграмм последовательности и кооперативных диаграмм). В первую очередь строится диаграмма (одна или более), описывающая основной поток событий и его подчиненные потоки. Для каждого альтернативного потока событий строится отдельная диаграмма.

Примеры: обработка ошибок; контроль времени выполнения; обработка неправильных вводимых данных.

Нецелесообразно описывать тривиальные потоки событий (например, в потоке участвует только один объект).

Практическое задание

Разработать на диаграмме классов соответствующие классы по индивидуальному заданию.

Контрольные вопросы

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

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

Цель работы: изучение технологии формирования атрибутов, операций объектов и отношений между ними на диаграмме классов.

Теоретический материал

Диаграммы последовательности позволяют сформировать все операции (методы) классов. Вид классов после добавления операций представлен на рисунке 4.1.

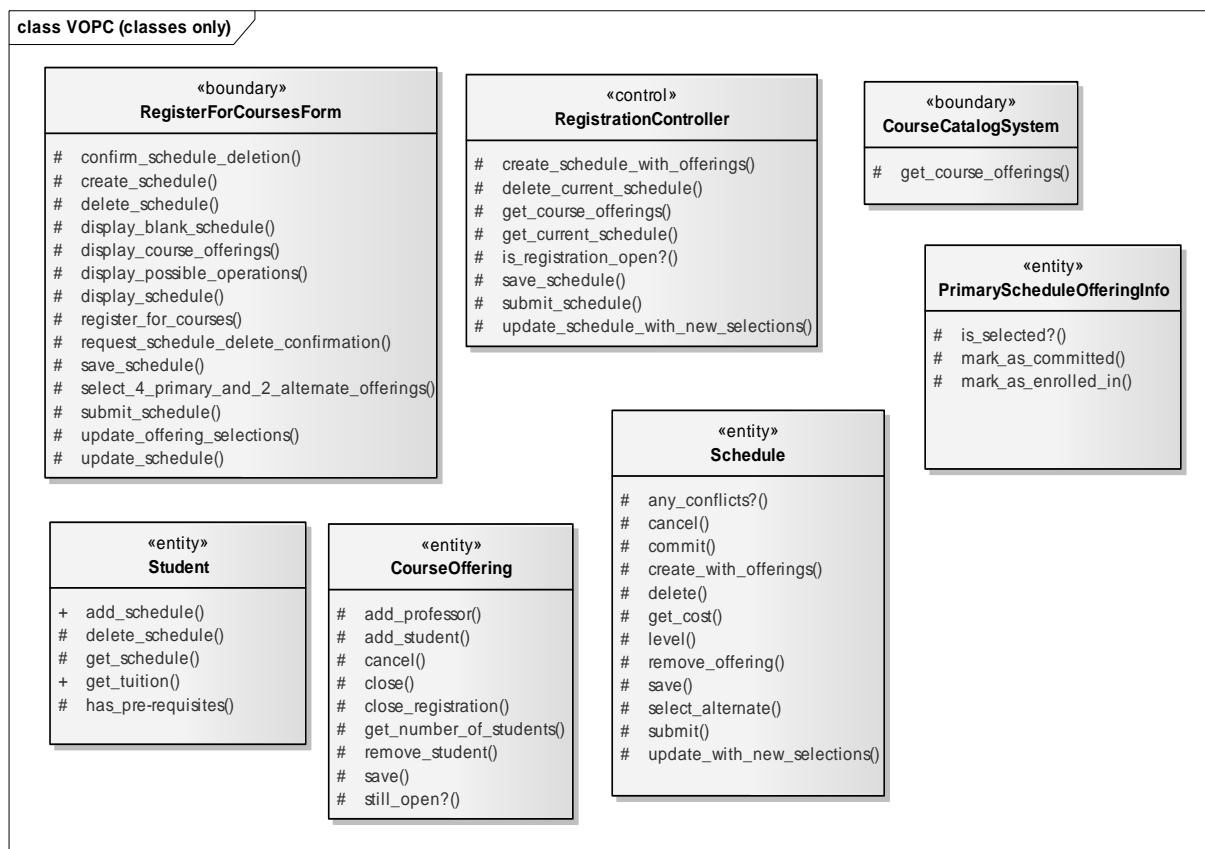


Рисунок 4.1 – Диаграмма классов с операциями

Проектирование баз данных.
 Разработка таблиц базы данных.
 Создание новой таблицы:

- 1) нажмите на панели инструментов кнопку Table;
- 2) щелкните мышью в том месте диаграммы, куда собираетесь поместить таблицу;
- 3) в открывшемся окне введите имя таблицы;
- 4) в окне Database выберите тип базы данных.

Для добавления колонок таблицы и редактирования их свойств:

- 1) щелкните правой кнопкой по таблице;
- 2) в открывшемся меню выберите пункт Attributes;
- 3) в открывшемся окне введите название колонки;
- 4) в окне Data Type выберите тип данных;
- 5) в окне Initial при необходимости введите начальное значение, которое может быть использовано как значение по умолчанию для колонки.

Если колонка представляет первичный ключ для таблицы, установите флажок Primary Key.

Enterprise Architect может генерировать простые DDL скрипты для создания таблиц в вашей модели.

Генерация DDL для таблицы:

- 1) щелкните правой кнопкой мыши по таблице, для которой генерируется DDL-скрипт;
- 2) в открывшемся меню выберите пункт Generate DDL;
- 3) открывшемся окне выберите путь, куда будет сгенерирован скрипт;
- 4) установите флажки, где необходимо;
- 5) для создания DDL-скрипта нажмите кнопку Generate;
- 6) после завершения генерации откройте созданный текстовый файл и просмотрите результаты.

Добавление атрибутов:

- 1) щелкните правой кнопкой мыши на классе Student;
- 2) в открывшемся меню выберите пункт Attributes;
- 3) введите новый атрибут address;
- 4) нажмите кнопку Save, для добавления нового атрибута нажмите кнопку New.
- 5) повторите шаги 1–4, добавив атрибуты name и studentID;
- 6) добавьте атрибуты к классам CourseOffering, Shedule и PrimaryScheduleOfferingInfo, как показано на рисунке 4.2.

Связи между классами (ассоциации) определяются на основе диаграмм взаимодействия. Если два объекта взаимодействуют (обмениваются сообщениями), между ними должна существовать связь (путь взаимодействия). Для ассоциаций задаются множественность и, возможно, направление навигации. Могут использоваться множественные ассоциации, агрегации и классы ассоциаций.

Добавление связей.

Добавим связи к классам, принимающим участие в варианте использования Register for Courses. Для отображения связей между классами построим три новых диаграмм классов в кооперации Register for Courses пакета Use-Case Realization – Register for Courses (рисунки 4.3–4.5).



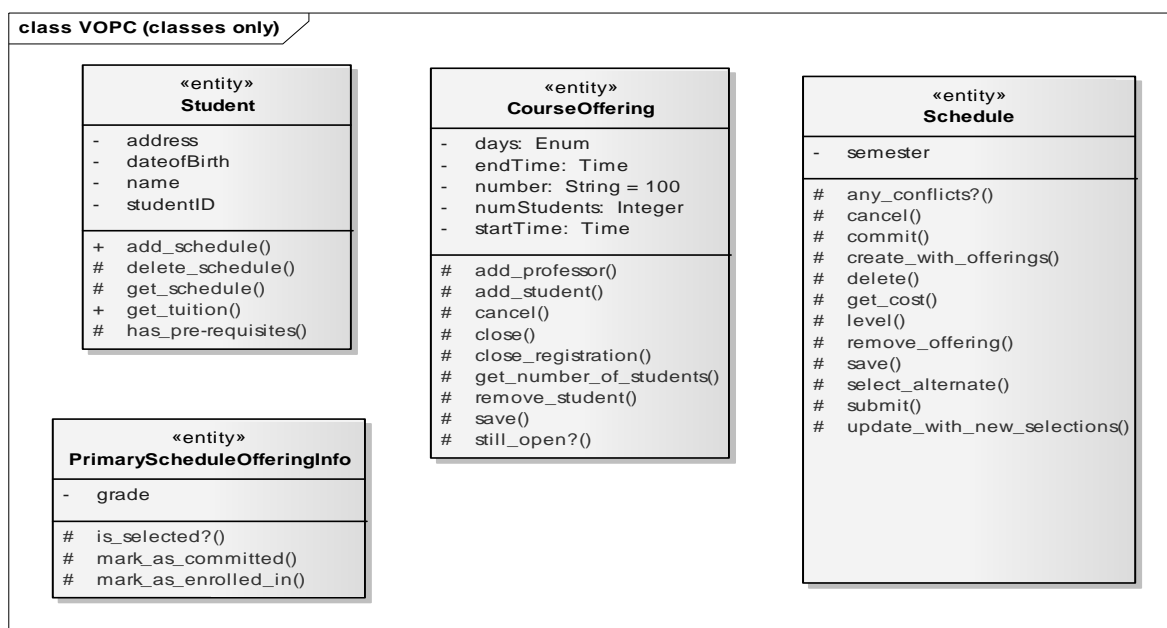


Рисунок 4.2 – Классы с операциями и атрибутами

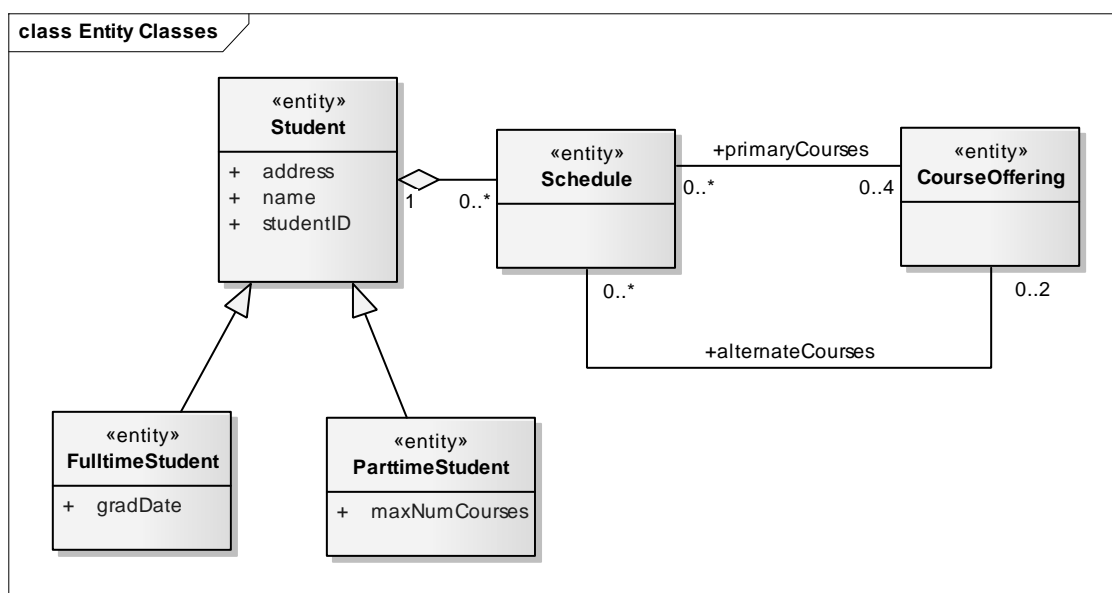


Рисунок 4.3 – Диаграмма Entity Classes (классы-сущности)

Добавлены два новых класса – подклассы FulltimeStudent (студент очного отделения) и ParttimeStudent (студент вечернего отделения).

На данной диаграмме показаны классы ассоциаций, описывающие связи между классами Schedule и CourseOffering и добавлен суперкласс ScheduleOfferingInfo. Данные и операции, содержащиеся в этом классе (status – курс включен в график или отменен), относятся как к основным, так и к альтернативным курсам, в то время как оценка (grade) и окончательное включение курса в график могут иметь место только для основных курсов.

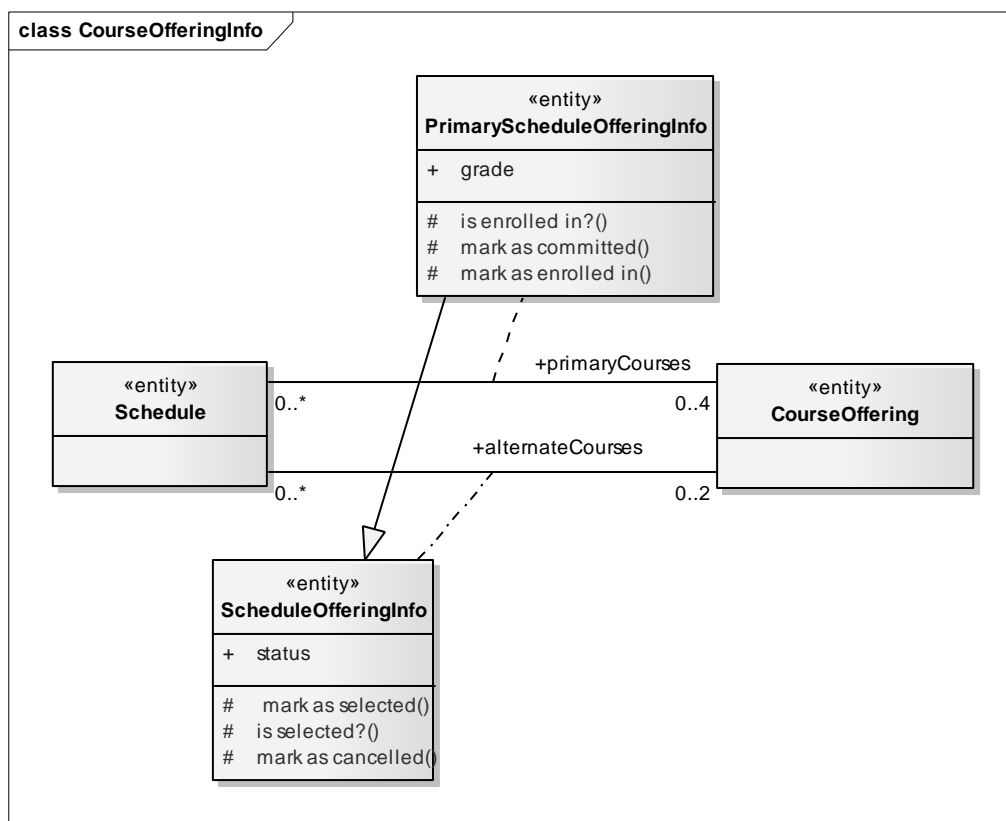


Рисунок 4.4 – Диаграмма CourseOfferingInfo

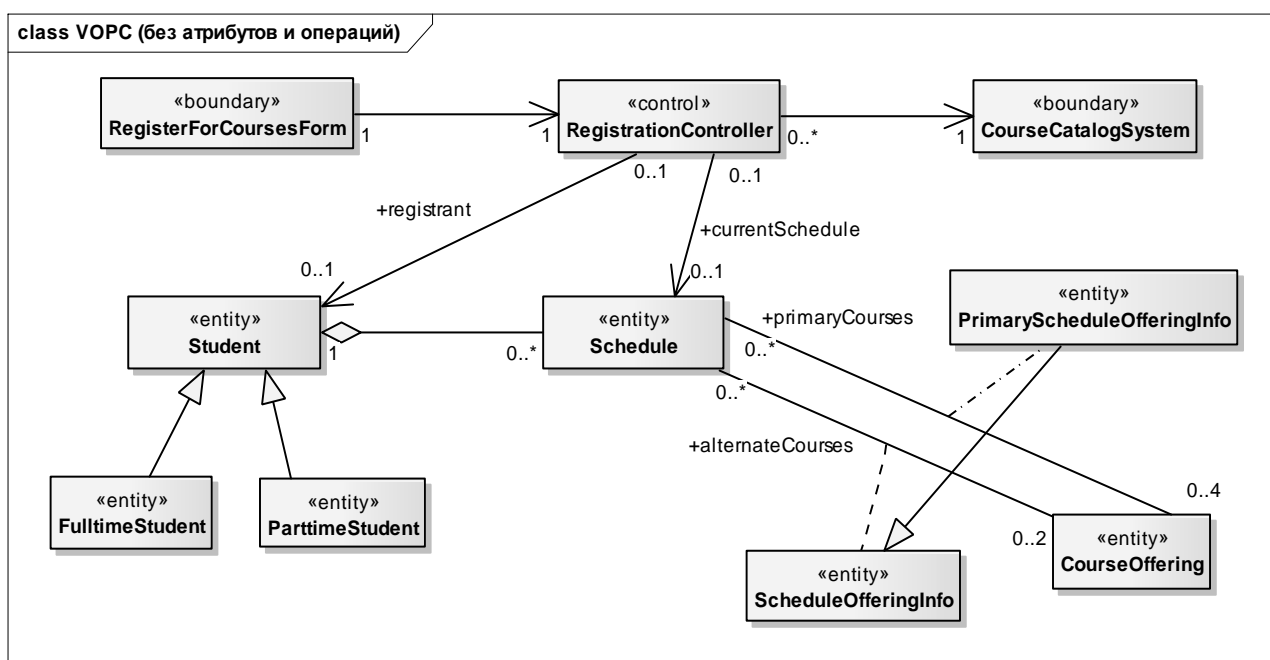


Рисунок 4.5 – Полная диаграмма классов VOPC (без атрибутов и операций)

Создание ассоциаций.

Ассоциации создают непосредственно на диаграмме классов. Панель инструментов диаграммы классов содержит кнопки для создания как одно-, так и двунаправленных ассоциаций. Чтобы на диаграмме классов создать ассоциацию:

- нажмите на панели инструментов кнопку Association Class;
- проведите мышью линию ассоциации от одного класса к другому.

Создание агрегаций:

- нажмите кнопку Aggregate панели инструментов;
- проведите линию агрегации от класса-части к целому.

Создание обобщений. При создании обобщения может потребоваться перенести некоторые атрибуты или операции из одного класса в другой. Чтобы поместить обобщение на диаграмму классов:

- нажмите кнопку Generalize панели инструментов;
- проведите линию обобщения от подкласса к суперклассу.

Спецификации связей касаются имен ассоциаций, ролевых имен, множественности и классов ассоциаций. Чтобы задать множественность связи:

- щелкните правой кнопкой мыши на одном конце связи;
- в открывшемся меню выберите пункт Multiplicity;
- укажите нужную множественность;
- повторите то же самое для другого конца связи.

Чтобы задать имя связи:

- выделите нужную связь;
- введите ее имя.

Чтобы задать связи ролевое имя:

- щелкните правой кнопкой мыши на ассоциации с нужного конца;
- в открывшемся меню выберите пункт Target Role;
- введите ролевое имя.

Классы анализа преобразуются в проектные классы:

– проектирование граничных классов – зависит от возможностей среды разработки пользовательского интерфейса;

– проектирование классов-сущностей – с учетом соображений производительности (выделение в отдельные классы атрибутов с различной частотой использования);

– проектирование управляющих классов – удаление классов, реализующих простую передачу информации от граничных классов к сущностям;

– идентификация устойчивых (persistent) классов, содержащих хранимую информацию.

Обязанности классов, определенные в процессе анализа, преобразуются в операции. Каждой операции присваивается имя, характеризующее ее результат. Создается краткое описание операции, включая смысл всех ее параметров. Определяется видимость операции: public, private, protected.

Определяются (уточняются) атрибуты классов:

- кроме имени, задается тип и значение по умолчанию (необязательное);
- учитываются соглашения по именованию атрибутов, принятые в проекте и языке реализации;
- задается видимость атрибутов: public, private, protected;
- при необходимости определяются производные (вычисляемые) атрибуты.



Определение атрибутов и операций для класса Student.

Чтобы задать тип данных, значение по умолчанию и видимость атрибута:

- щелкните правой кнопкой мыши на классе в браузере;
- в открывшемся меню выберите пункт Attributes;
- укажите тип данных в раскрывающемся списке типов или введите собственный тип данных;
- в поле Initial (Первоначальное значение) введите значение атрибута по умолчанию;
- в поле Scope выберите видимость атрибута: Public, Protected, Private. По умолчанию видимость всех атрибутов соответствует Private (рисунок 4.6).

«entity» Student
- address: String - dateOfBirth: Date - name: String - studentID: Integer
+ addSchedule(Schedule) : void + deleteSchedule(Semester) : void + getAddress() : String + getName() : String + getSchedule(Semester) : Schedule + getStudentID() : Integer + getTuition() : Double + hasPrerequisites(CourseOffering) : Boolean # passed(CourseOffering) : Boolean

Рисунок 4.6 – Класс Student с полностью определенными операциями и атрибутами

Чтобы задать тип возвращаемого значения, стереотип и видимость операции:

- щелкните правой кнопкой мыши на операции в браузере;
- в открывшемся меню выберите пункт Operations;
- укажите тип возвращаемого значения (Return Type) в раскрывающемся списке или введите свой тип;
- укажите стереотип в соответствующем раскрывающемся списке или введите новый;
- в поле Scope укажите значение видимости операции: Public, Protected, Private. По умолчанию видимость всех операций установлена в Public.

Чтобы добавить к операции параметр:

- откройте окно спецификации операции;
- нажмите кнопку Edit Parameters;
- введите имя параметра;
- выберите тип данных аргумента;
- если надо, введите значение аргумента по умолчанию (Default);
- нажмите кнопку Save для сохранения параметра.

Практическое задание

Разработать диаграммы классов по индивидуальному заданию.

Контрольные вопросы

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

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

Цель работы: изучение технологии формирования диаграммы кооперации.

Теоретический материал

В UML есть четыре типа диаграмм взаимодействия:

- 1) диаграмма последовательности;
- 2) диаграмма кооперации;
- 3) диаграмма обзора взаимодействия;
- 4) диаграмма синхронизации.

Диаграмма кооперации – диаграмма, на которой изображаются взаимодействия между частями композитной структуры или ролями кооперации. В отличие от диаграммы последовательности, на диаграмме кооперации явно указываются отношения между объектами, а время как отдельное измерение не используется (применяются порядковые номера вызовов).

Диаграмма кооперации моделирует взаимодействия между объектами или частями в терминах упорядоченных сообщений. Коммуникационные диаграммы представляют комбинацию информации, взятой из диаграмм классов, последовательности и вариантов использования, описывая сразу и статическую структуру, и динамическое поведение системы.

Диаграммы имеют свободный формат упорядочивания объектов и связей как в диаграмме объектов. Чтобы поддерживать порядок сообщений при таком свободном формате, их хронологически нумеруют. Чтение диаграммы кооперации начинается с сообщения 1.0 и продолжается по направлению пересылки сообщений от объекта к объекту.

Диаграмма кооперации показывает во многом ту же информацию, что и диаграмма последовательности, но из-за другого способа представления информации какие-то вещи на одной диаграмме видеть проще, чем на другой. Диаграмма кооперации нагляднее показывает, с какими элементами взаимодействует каждый элемент, а диаграмма последовательности яснее показывает, в каком порядке происходят взаимодействия.

Создание диаграммы кооперации.

На рисунке 5.1 представлена кооперативная диаграмма.



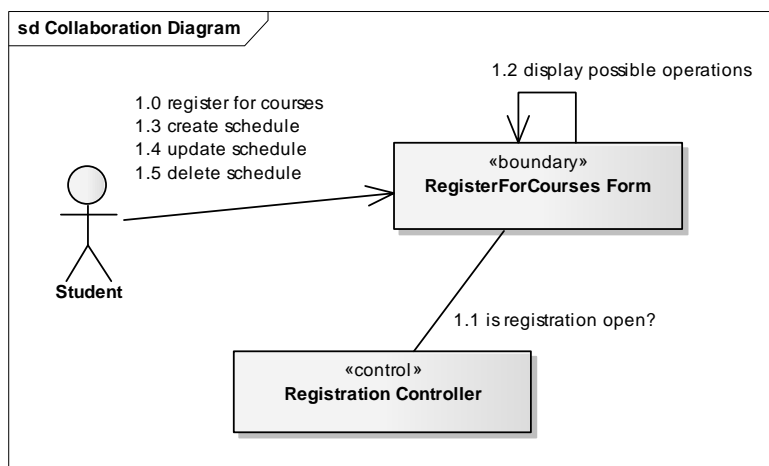


Рисунок 5.1 – Диаграмма кооперации Register for Courses – Basic Flow

Практическое задание

Разработать кооперативную диаграмму по индивидуальному заданию.

Контрольные вопросы

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

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

Цель работы: изучение технологии формирования диаграммы последовательности и редактирование свойств ее элементов.

Теоретический материал

Создание диаграмм последовательности

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

Готовые диаграммы последовательности должны иметь вид, как на рисунках 6.1–6.5.

Настройка:

- меню модели выберите пункт Tools > Options или нажмите сочетание клавиш CTRL + F9;
- перейдите на вкладку Diagram > Sequence;
- контрольный переключатель Show Sequence Numbering должен быть помечен;
- нажмите Close, чтобы выйти из окна параметров.

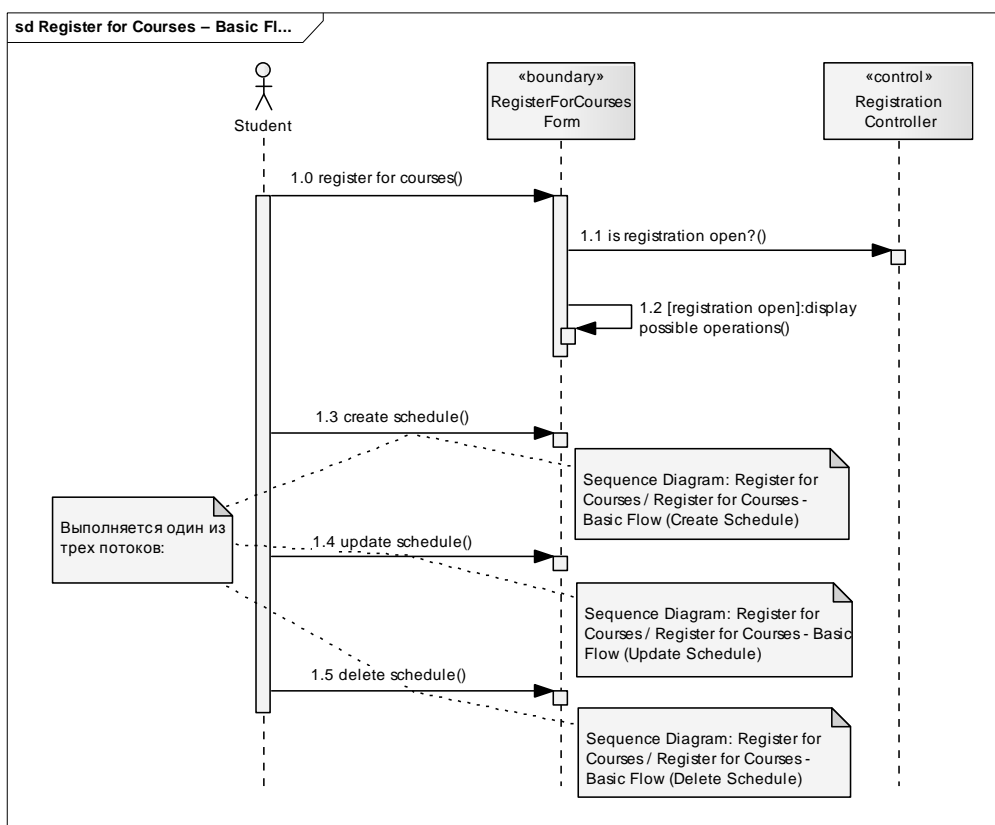


Рисунок 6.1 – Диаграмма последовательности Register for Courses – Basic Flow

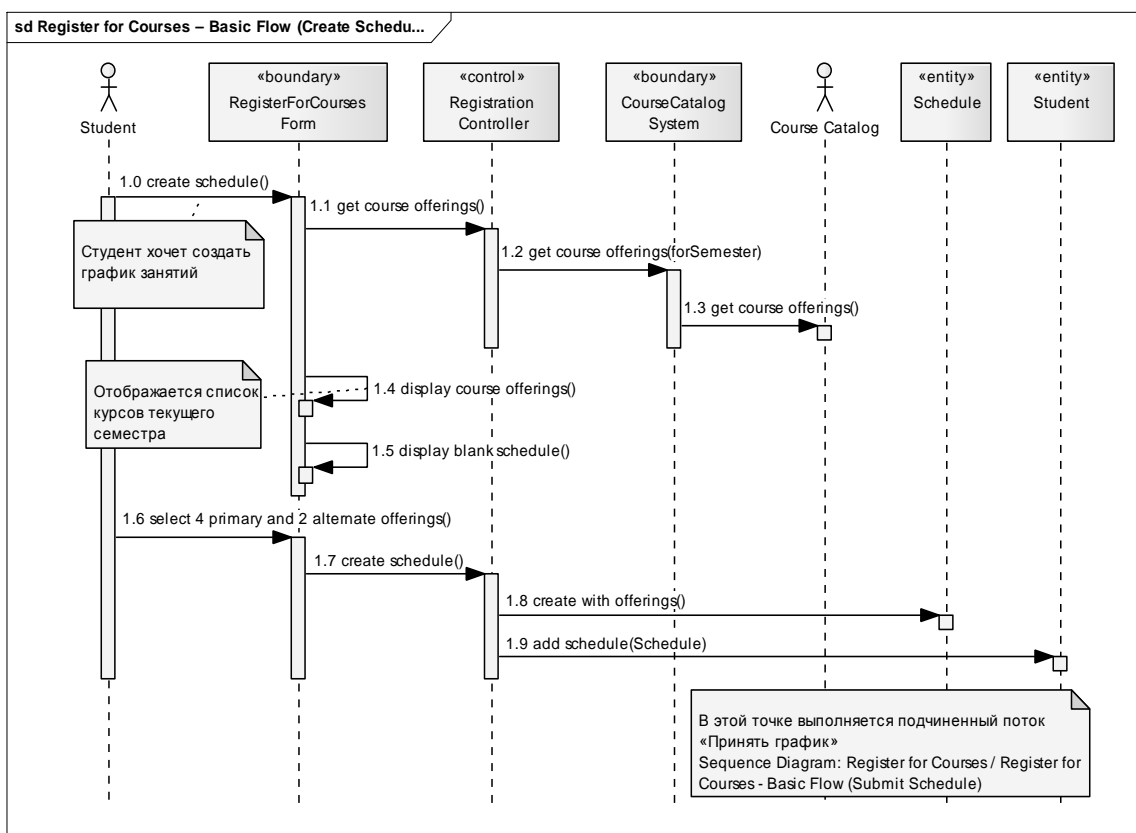


Рисунок 6.2 – Диаграмма последовательности Register for Courses – Basic Flow (Create Schedule)

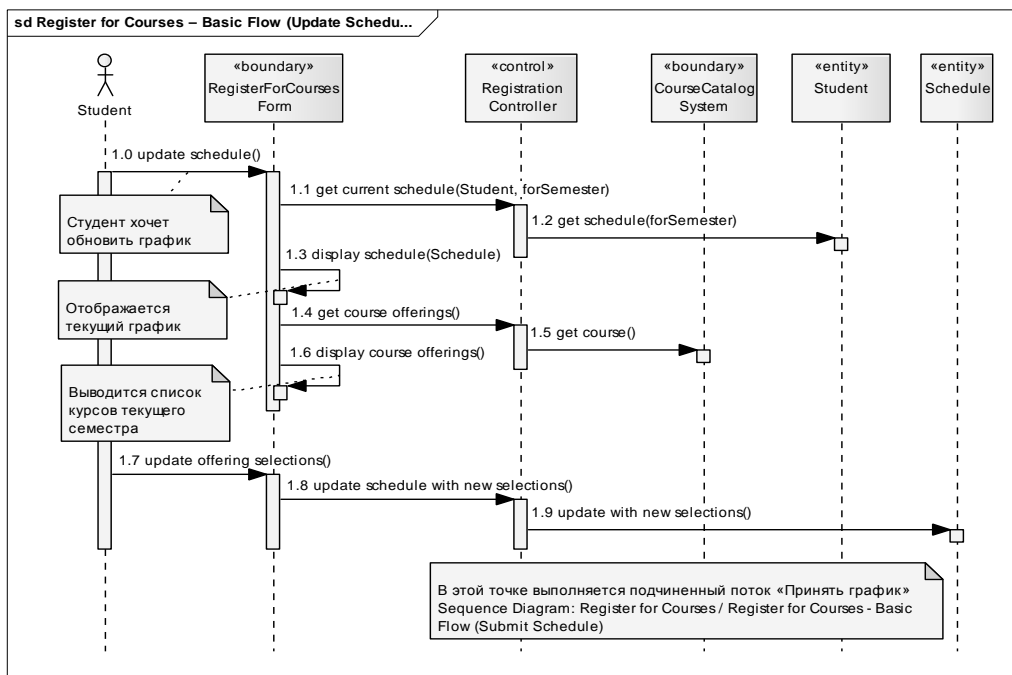


Рисунок 6.3 – Диаграмма последовательности Register for Courses – Basic Flow (Update Schedule)

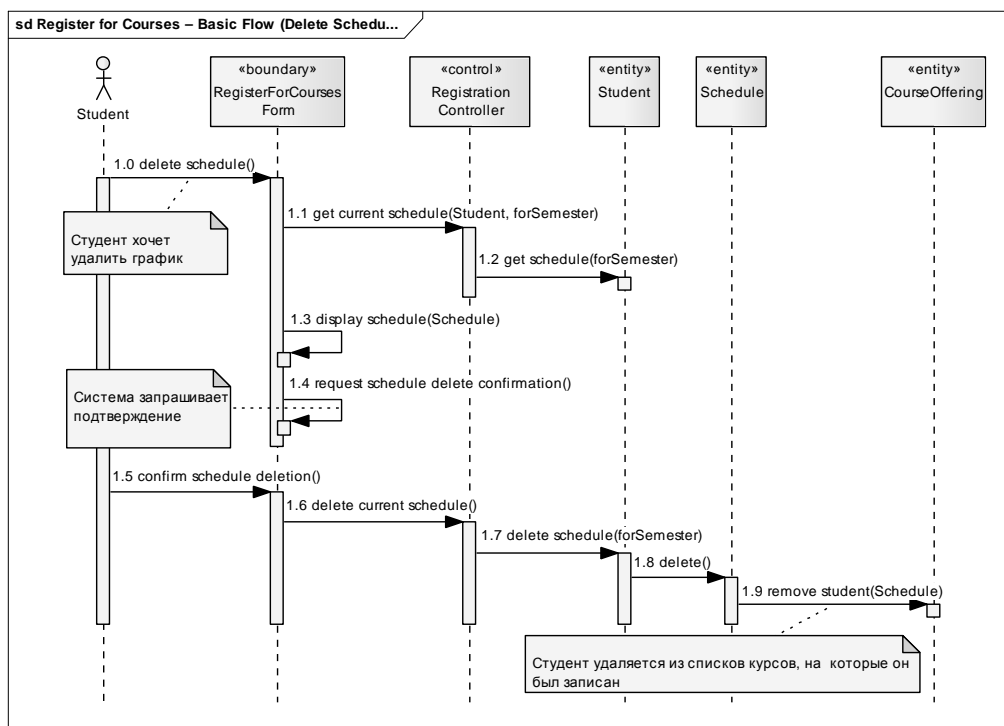


Рисунок 6.4 – Диаграмма последовательности Register for Courses – Basic Flow (Delete Schedule)

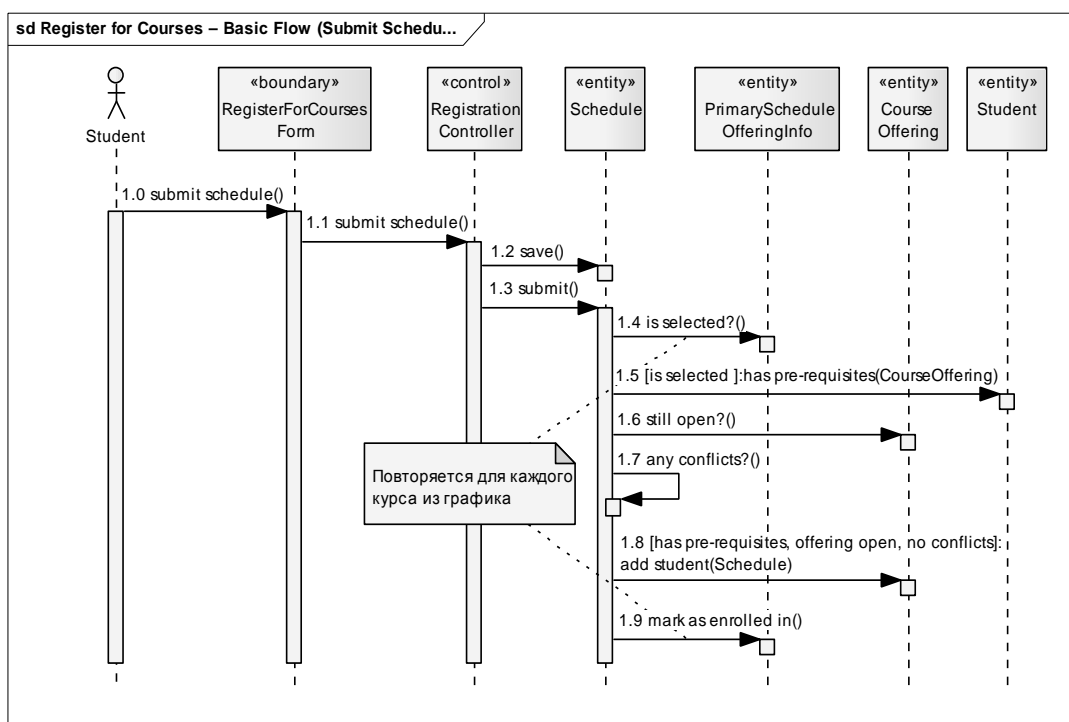


Рисунок 6.5 – Диаграмма последовательности Register for Courses – Basic Flow (Submit Schedule)

Создание диаграммы последовательности:

- щелкните правой кнопкой мыши на кооперации Register for Courses в пакете Use-Case Realization – Register for Courses;
- в открывшемся меню выберите пункт Add > Add Diagram > Sequence;
- назовите новую диаграмму Register for Courses – Basic Flow;
- дважды щелкните на ней, чтобы открыть ее.

Добавление на диаграмму действующего лица, объектов и сообщений:

- перетащите действующее лицо Student из браузера на диаграмму;
- перетащите классы RegisterForCoursesForm и RegistrationController из браузера на диаграмму;
- на панели инструментов нажмите кнопку Message (Сообщение);
- проведите мышью от линии жизни действующего лица Student к линии жизни объекта RegisterForCoursesForm;
- выделив сообщение, введите его имя: register for courses;
- повторите, описанные выше действия, чтобы поместить на диаграмму остальные сообщения, как показано на рисунке 6.1 (для рефлексивного сообщения 1.2 используется кнопка Self-Message).

Выполните аналогичные действия для создания диаграмм последовательности, показанных на рисунках 6.2–6.5. Обратите внимание, что на диаграмме (см. рисунок 6.5) появился объект нового класса PrimaryScheduleOffering-Info (класса ассоциаций, описывающего связь между классами Schedule и OfferingInfo), который нужно предварительно создать.

Создание примечаний.

Чтобы поместить на диаграмму примечание:

- нажмите на панели инструментов кнопку Note;
- щелкните мышью в том месте диаграммы, куда собираетесь поместить примечание;
- выделив новое примечание, введите туда текст;
- чтобы прикрепить примечание к элементу диаграммы, щелкните по примечанию правой кнопкой мыши, в появившемся окне выберите Advanced > Set Attached Links;
- установите флажок напротив того (тех) элемента (элементов) диаграммы, с которым связываете примечание. Между примечанием и элементом возникнет штриховая линия.

Кроме примечаний, на диаграмму можно поместить также и текстовую область. С ее помощью можно, например, добавить к диаграмме заголовок.

Чтобы поместить на диаграмму текстовую область:

- на панели управления нажмите кнопку New Text Element;
- щелкните мышью внутри диаграммы, чтобы поместить туда текстовую область. Введите в неё текст.

Практическое задание

Разработать диаграммы последовательности по индивидуальному заданию.

Контрольные вопросы

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

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

Цель работы: изучение технологии создания диаграммы состояний.

Теоретический материал

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 (рисунок 7.1). 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 вари-



ант использования, можно выделить еще два состояния: инициализация (до начала регистрации студентов на курс) и отмена (курс исключается из расписания).

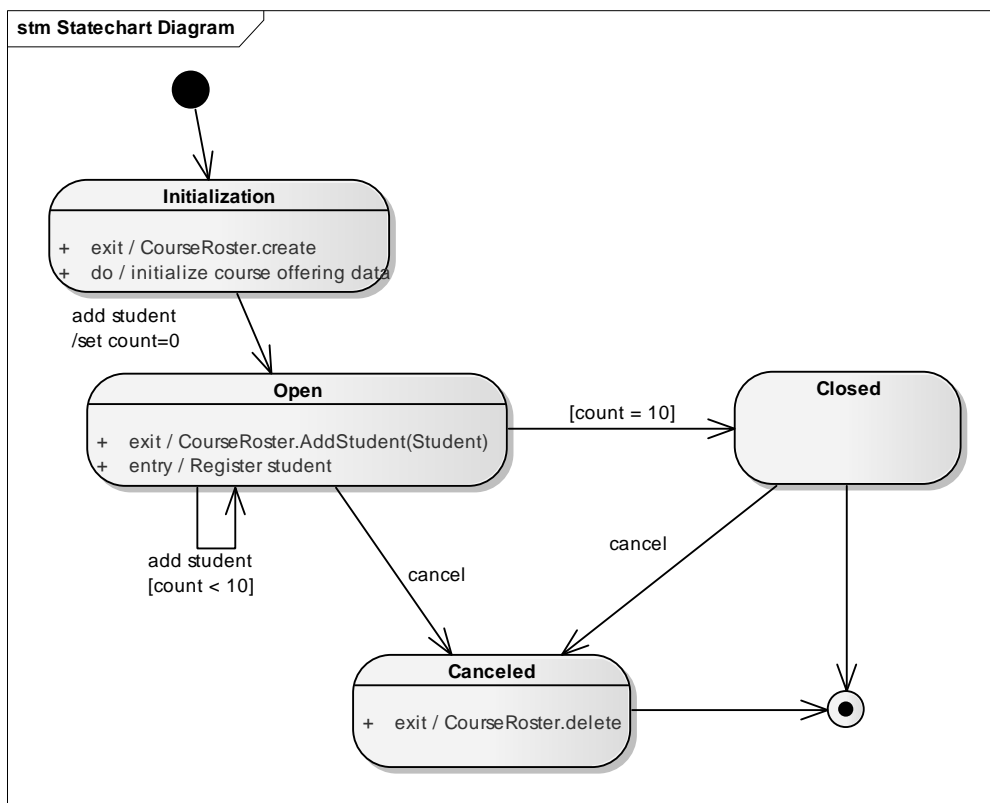


Рисунок 7.1 – Диаграмма состояний для класса CourseOffering

Создание диаграммы состояний для класса Course-Offering.

Для создания диаграммы состояний:

- щелкните правой кнопкой мыши в браузере на нужном классе;
- в открывшемся меню выберите пункт Add > Add Diagram > State Machine.

Чтобы добавить состояние:

- на панели инструментов нажмите кнопку State;
- щелкните мышью на диаграмме состояний в том месте, куда хотите его поместить.

Чтобы добавить деятельность:

- щелкните правой кнопкой мыши по требуемому состоянию;
- в открывшемся меню выберите пункт Operations;
- в открывшемся окне в поле Name введите название деятельности;
- в окне Action укажите do, чтобы сделать новое действие деятельностью.

Чтобы добавить входное действие, в окне Action укажите entry.

Чтобы добавить выходное действие, в окне Action укажите exit.

Чтобы добавить переход:

- нажмите кнопку Transition панели инструментов;
- нажмите левую клавишу мыши на состоянии, откуда осуществляется переход;
- не отпуская клавишу мыши, проведите линию перехода до того состоя-

ния, где он завершается.

Чтобы добавить рефлексивный переход:

- нажмите кнопку Transition панели инструментов;
- щелкните на том состоянии, где осуществляется рефлексивный переход.

Чтобы добавить событие, ограждающее условие и действие:

- дважды щелкните на переходе, чтобы открыть окно его спецификации;
- на вкладке General введите событие в поле Name;
- перейдите на вкладку Constraints;
- введите ограждающее условие в поле Guard;
- введите действие в поле Effect.

Для указания начального или конечного состояния:

- на панели инструментов нажмите кнопку Initial или Final;

– щелкните мышью на диаграмме состояний в том месте, куда хотите поместить состояние.

Практическое задание

Разработать диаграмму состояний АСОИУ.

Контрольные вопросы

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

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




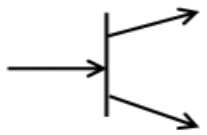
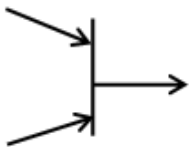


Цель работы: изучение технологии формирования диаграммы деятельности.

Теоретический материал

Диаграмма деятельности (activity diagram) – еще один способ описания поведения, который визуально напоминает блок-схему алгоритма. На ней показаны действия, состояния которых описаны на диаграмме состояний. Под деятельностью (англ. «activity») понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов – вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого. Управляющие узлы, используемые на диаграмме деятельности, приведены в таблице 8.1.



Таблица 8.1. – Управляющие узлы, используемые на диаграмме деятельности

Обозначение	Наименование на диаграмме деятельности
	Начальный узел (initial node)
	Завершение деятельности (activity final)
	Завершение потока (flow final)
	Ветвление (fork node)
	Соединение (join node)
	Слияние / решение (merge / decision node)
	Действие (activity)

Диаграммы деятельности состоят из ограниченного количества фигур, соединённых стрелками. Стрелки идут от начала к концу процесса и показывают потоки управления или потоки объектов (данных).

Пример простой диаграммы деятельности выполнения заказа приведен на рисунке 8.1.

Старт деятельности происходит с начального узла (*initial node*), а затем выполняется операция *Receive Order* (Принять заказ). Затем идет ветвление (*fork*), которое имеет один входной поток и несколько выходных параллельных потоков. Операции *Fill Order* (Заполнить заявку), *Send Invoice* (Послать счет) и следующие за ними выполняются параллельно.

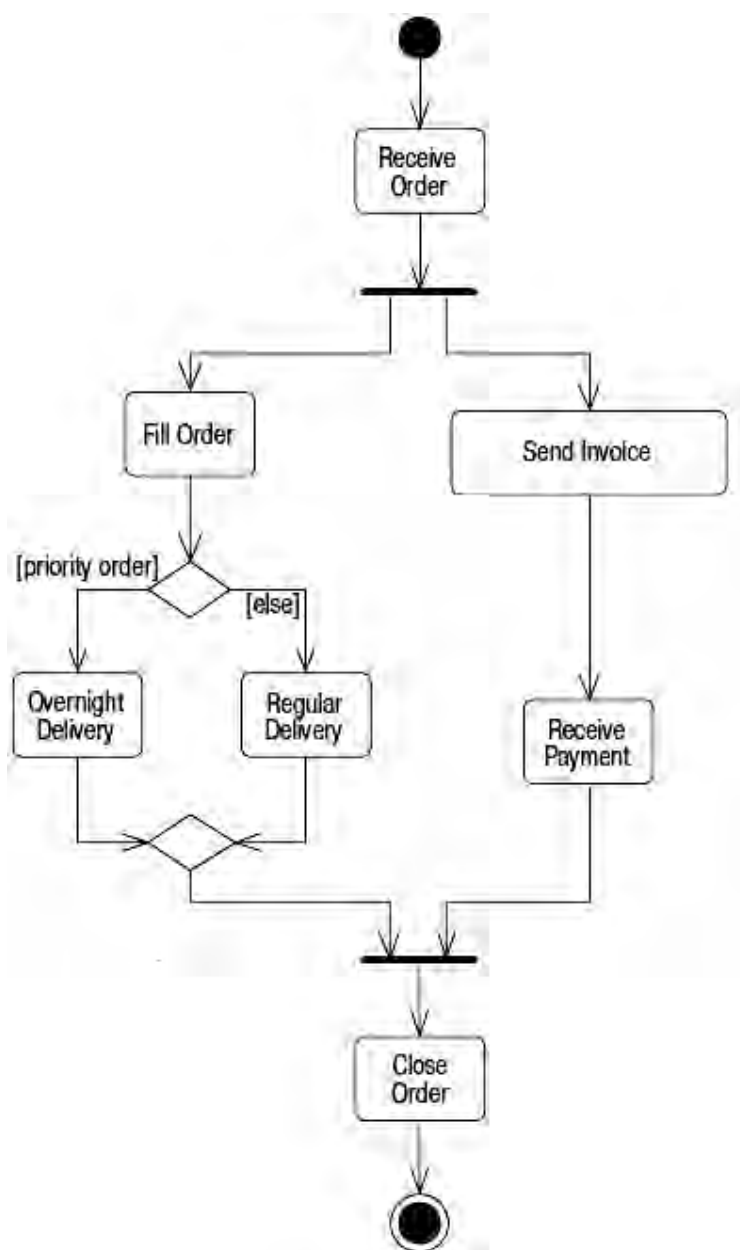


Рисунок 8.1 – Диаграмма деятельности

Практическое задание

Разработать диаграмму состояний АСОИУ.

Контрольные вопросы

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

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

Цель работы: изучение технологии формирования диаграммы компонентов.

Теоретический материал

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.

Практическое задание

Разработать диаграмму компонентов АСОИУ.

Контрольные вопросы

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



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

Цель работы: изучение технологии формирования диаграммы размещения.

Теоретический материал

Распределенная конфигурация системы моделируется с помощью диаграммы размещения (рисунок 10.1).

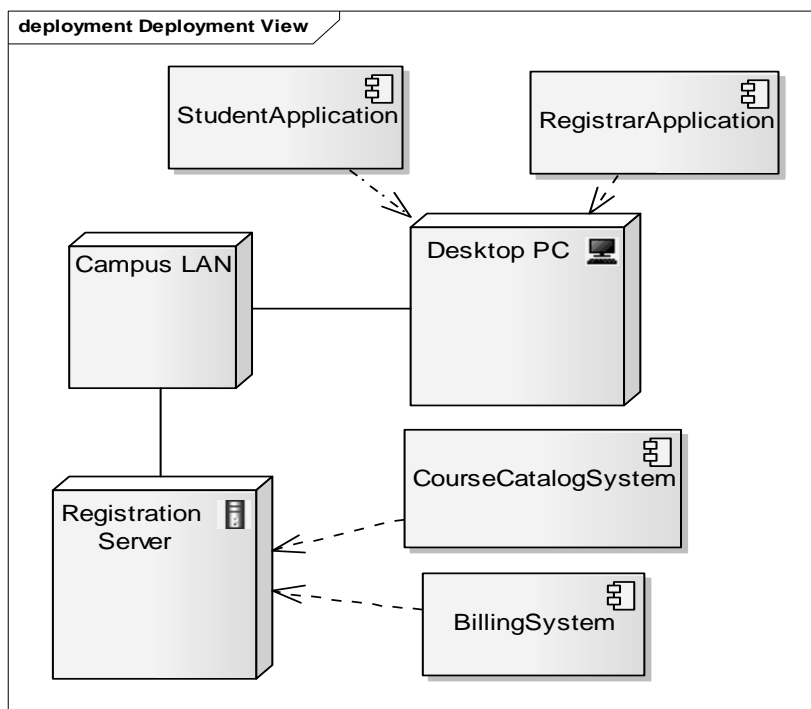


Рисунок 10.1 – Сетевая конфигурация системы регистрации

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.

Распределение процессов по узлам сети производится с учетом следующих факторов:

– используемые образцы распределения (трехзвенная клиент-серверная конфигурация, «толстый клиент», «тонкий клиент», равноправные узлы (peer-to-peer) и т. д.);

– время отклика;

– минимизация сетевого трафика;

– мощность узла;

– надежность оборудования и коммуникаций.

Создание диаграммы размещения системы регистрации.

Чтобы открыть диаграмму размещения, надо дважды щелкнуть мышью на представлении Deployment View (представлении размещения) в браузере.

Чтобы поместить на диаграмму процессор:

- на панели инструментов диаграммы нажмите кнопку Node;
- щелкните на диаграмме размещения в том месте, куда хотите его поместить;
- введите имя процессора.

В спецификациях процессора можно ввести информацию о его стереотипе, характеристиках. Стереотипы применяются для классификации процессоров.

Характеристики процессора – это его физическое описание. Оно может, в частности, включать скорость процессора и объем памяти.

Чтобы назначить процессору стереотип:

- щелкните правой кнопкой по изображению процессора;
- в отрывшемся окне выберите пункт Properties;
- на вкладке General выберите стереотип в поле Stereotype.

Практическое задание

Разработать диаграмму развертывания.

Контрольные вопросы

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

11 Лабораторная работа № 11. Определение трудоемкости разработки АСОИ

Цель работы: изучение определения трудоемкости разработки АСОИ.

Теоретический материал

Заключительным этапом в проектировании АСОИ является определение трудоемкости ее разработки. Данный этап относится к компетенции руководителя проектов (Computer and Information System Manager). Руководитель проектов – специалист высших квалификационных уровней, который осуществляет планирование, координацию и руководство разработкой проектов. Руководитель проекта определяет общую трудоемкость разработки программного обеспечения и календарный план – график разработки.

Трудозатраты на разработку каждого элемента программного обеспечения существенно зависят от квалификации специалистов. Примерные трудозатраты разработки элементов ПО приведены в таблице 11.1. Учитываются затраты на следующие элементы программного продукта: диалоговые элементы на форме,



вычисляемые процедуры, страницы печатной формы, запросы к БД, таблицы БД.

Таблица 11.1 – Примерные трудозатраты разработки элементов ПО

Элемент	Время разработки, ч.
Диалоговый элемент на форме, документе	0,5...1
Сто строк кода вычислительных процедур	2...16
Запрос к БД	1...2
Таблица в БД (10 полей)	0,5...1

Анализируя диаграммы классов, руководитель заполняет таблицу трудоемкости разработки, указывает количество элементов АСОИ и необходимое на их разработку количество времени. Календарный план-график разработки АСОИ формируется исходя из 8-часового рабочего дня.

Таблицы 11.2 и 11.3 являются примером расчета по определению общей трудоемкости проекта и календарного плана разработки.

Таблица 11.2 – Трудоемкость разработки программного обеспечения АСОИ

Компонент	Число полей	Число диалоговых элементов	Число вычисляемых процедур	Число страниц печатной формы	Трудоемкость разработки, ч
«table» tOrders	4				0,5
«table» tOrganisation	3				0,5
«table» tProduction	10				1
fOrder		10			10
fsProduction		14			14
fOrganisation		5			5
qrOrderProduction			2		4
docOrder	8			1	8
Итого					43

Таблица 11.3 – Календарный план-график разработки АСОИ

Наименование компонента	Срок выполнения	
	Начало	Окончание
«table» tOrders	01.03.2018 г.	04.03.2018 г.
«table» tOrganisation	01.03.2018 г.	04.03.2018 г.
«table» tProduction	01.03.2018 г.	04.03.2018 г.
fOrder	05.03.2018 г.	06.03.2018 г.
fsProduction	08.03.2018 г.	12.03.2018 г.
fOrganisation	13.03.2018 г.	13.03.2018 г.
qrOrderProduction	14.03.2018 г.	14.03.2018 г.
docOrder	17.03.2018 г.	20.03.2018 г.

Практическое задание

Рассчитать трудоемкость разработки проектируемой АСОИ и календарный план-график разработки.

Контрольные вопросы

- 1 Цель расчета трудоемкости АСОИ.
- 2 Цель календарного плана-графика разработки АСОИ.

Список литературы

- 1 **Олейник, П. П.** Корпоративные информационные системы : учебник для бакалавров и специалистов / П. П. Олейник. – Санкт-Петербург : Питер, 2014. – 176 с. : ил.
- 2 **Рудинский, И. Д.** Технология проектирования автоматизированных систем обработки информации и управления : учебное пособие / И. Д. Рудинский. – Москва : Горячая линия-Телеком, 2014 г. – 302 с.
- 3 **Советов, Б. Я.** Информационные технологии : учебник для прикладного бакалавриата / Б. Я. Советов, В. В. Цехановский. – 7-е изд., перераб. и доп. – Москва : Юрайт, 2018. – 256 с.
- 4 **Хетагуров, Я. А.** Проектирование автоматизированных систем обработки информации и управления (АСОИУ) : учебник / Я. А. Хетагуров. – Москва : БИНОМ, 2015. – 239 с.

