

Лабораторная работа №1

Знакомство с CASE-средой Modelio

Сдать до 7.09

1. Сведения о работе в среде Modelio

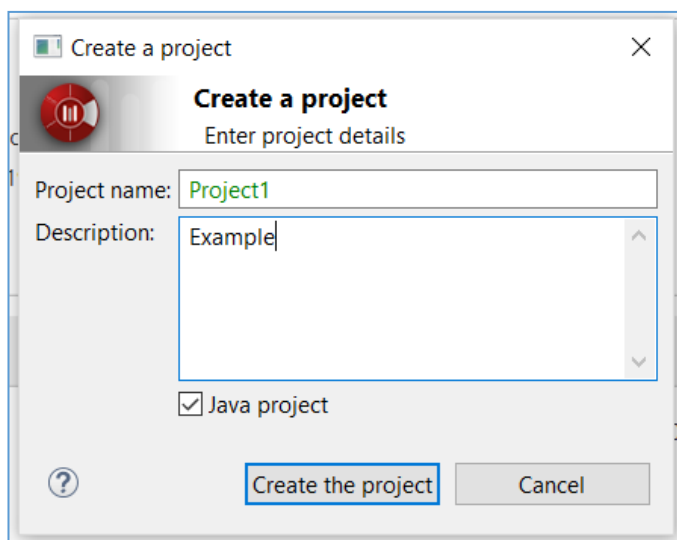
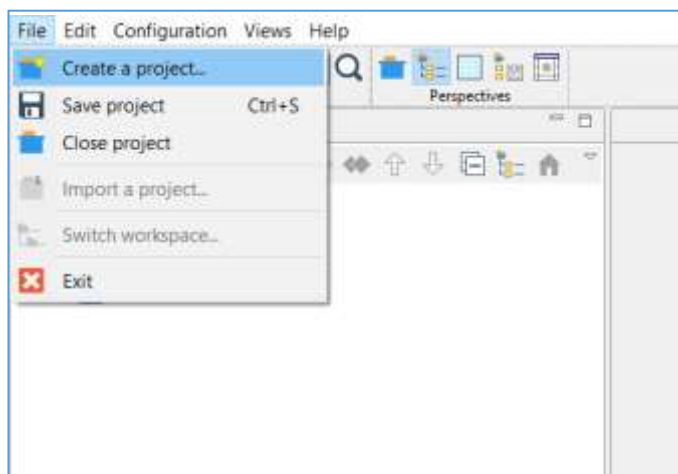
Для установки Modelio следует загрузить архив, соответствующий установленной у вас ОС, со страницы <https://www.modelio.org/downloads/download-modelio.html>.

Для Windows нужно учитывать следующие моменты:

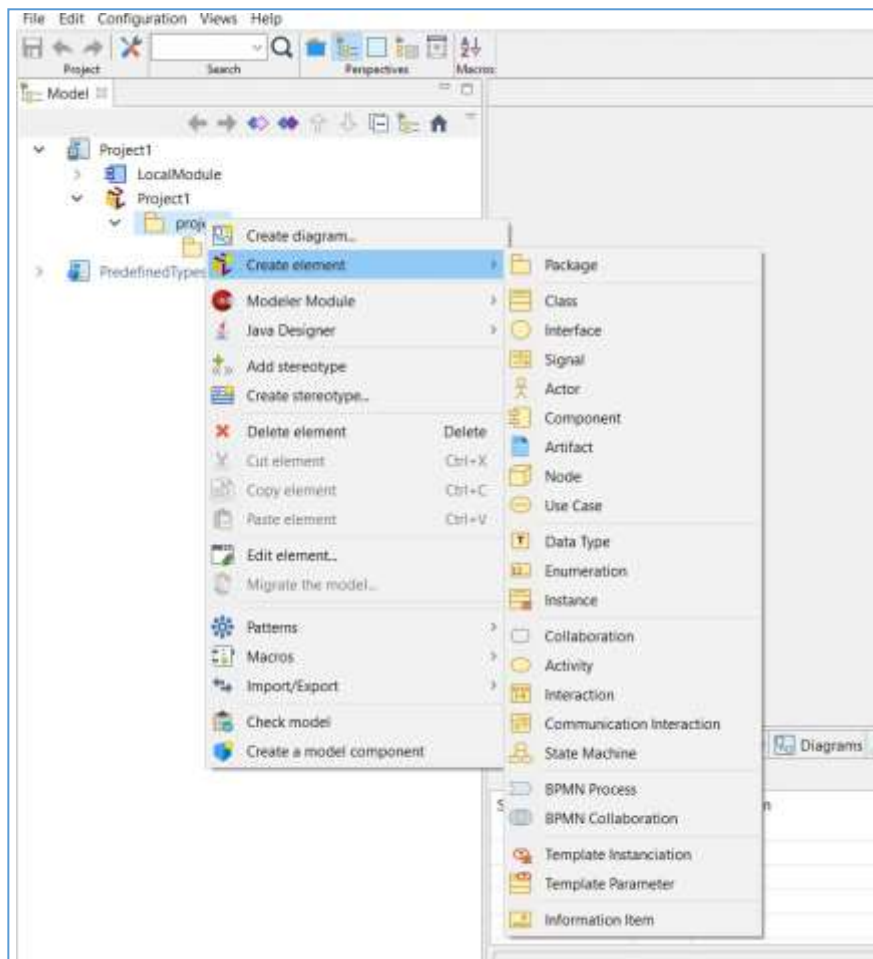
- Имя пользователя, под которым устанавливается среда не должно содержать пробелов
- Установка запускается самостоятельно (не надо распаковывать архив самому)
- При установке в Windows не рекомендуется разархивировать Modelio в каталог Program Files. Желательно распаковать его в свой каталог пользователя (изменить в процессе установки).

После установки, запустив среду Modelio, указываем путь к рабочей области (workspace), в которой будут находиться файлы нашего проекта (меню File -> Switch workspace).

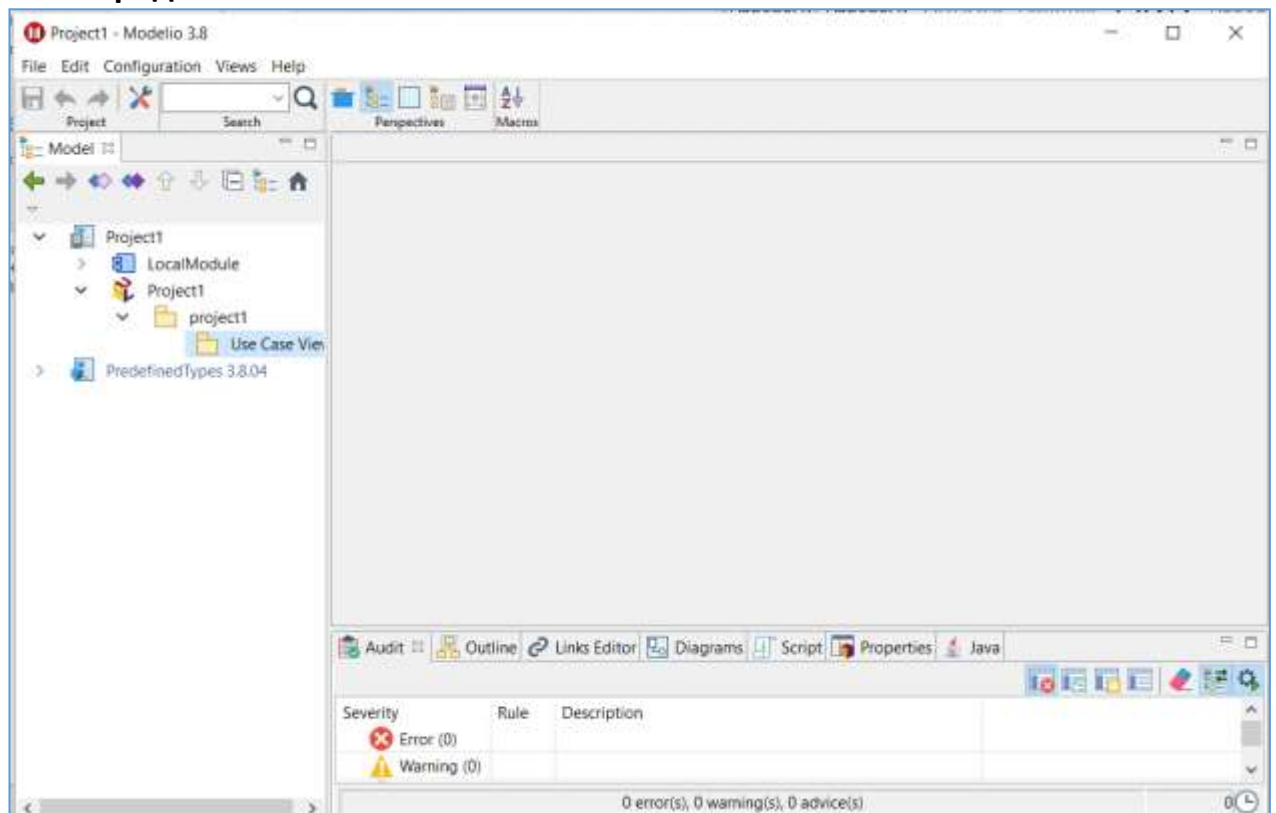
Теперь создаем свой первый проект Project1.



В своем проекте создайте пакет (понятие пакетов и их виды будут рассмотрены на следующем занятии) «Use Case View». Представление вариантов использования (Use Case View) будет содержать модель требований к системе.



Окно среды Modelio



В левой части находится навигатор или браузер, показывающий структуру проекта и модели. Туда мы будем добавлять диаграммы, пакеты, и другие составляющие модели с помощью

контекстного меню. Правая часть окна состоит из редактора диаграмм (вверху) и находящегося внизу окна с закладками Outline, Audit и т.д. при работе с элементами диаграмм там появятся еще вкладки Element и Symbol. Вкладка Element служит для задания свойств элементов модели. Вкладка Symbol нужна для задания параметров отображения элемента модели на диаграмме.

2. ПРИМЕР: Система регистрации для вуза. Описание

Перед руководителем информационной службы университета ставится задача разработки автоматизированной системы регистрации студентов на курсы. Предполагается система обучения, при которой посеместровый учебный план-график студентов не является единым и определяется в начале очередного семестра по заявкам студентов и профессоров.

В университете уже функционирует база данных, содержащая сведения обо всех курсах, которые могут быть в нем прочитаны (каталог курсов). О каждом курсе хранится название, его длительность и требования к предварительному уровню подготовки (т. е. перечень курсов, которые необходимо пройти прежде, чем слушать данный курс). Предполагается, что существующая база данных будет использоваться с новой системой, а не переделываться заново.

Регистрация на курсы происходит следующим образом: в начале каждого семестра профессора обращаются в учебный отдел, чтобы указать, какие курсы они собираются прочитать в течение семестра. Для каждого предлагаемого курса указывается день недели и номер пары. По заявкам профессоров составляется список предлагаемых курсов. Студенты могут ознакомиться с этим списком в учебном отделе. Студент может выбрать 4 курса из списка. В дополнение к этому студент может указать 2 альтернативных курса на тот случай, если какой-либо из выбранных им курсов окажется уже заполненным или отмененным. Студент может регистрироваться на курс только в том случае, если им выполнены требования к предварительному уровню подготовки. На каждый курс может записаться не более 10 и не менее 3 студентов. Если курс окажется заполненным в процессе регистрации, регистрирующиеся студенты должны быть извещены об этом, лишние заявки на курс не принимаются. В ходе регистрации студенты могут изменить свои планы (отказаться от выбранных курсов, добавить новые). Всего на процесс регистрации отводится две недели. Регистрация на отдельный курс может быть закончена раньше по решению учебной части. Курс считается отмененным по окончании регистрации, если на него записалось менее 3 студентов.

После того, как процесс регистрации завершен, регистратор, работающий в учебном отделе, утверждает индивидуальные планы-графики студентов и передает данные в расчетную систему, установленную в бухгалтерии. Расчетная система определяет размер платы за семестр и отправляет студентам по почте счета. Профессора получают в учебном отделе списки студентов, записавшихся на их курсы. В течение семестра профессора сообщают оценки в учебную часть. В конце семестра студенты могут просмотреть свои таблицы успеваемости.

3. Определение требований к создаваемой системе

Выполняя упражнения, мы сразу переходим к определению требований, минуя моделирование бизнес-процессов. Позволительно так поступать в случаях, когда предметная область, для которой разрабатывается ПО, сравнительно проста или хорошо знакома разработчикам.

Определение требований – процесс, в ходе которого определяются задачи, поставленные перед разработчиками, и создаются модели, на основе которых планируется разработка системы. Требование – это условие, которому должно удовлетворять программное

обеспечение, или свойство, которым оно должно обладать, чтобы удовлетворить потребность пользователя в решении некоторой задачи, или чтобы соответствовать условиям контракта, спецификации или стандарта. Все требования делятся на функциональные и нефункциональные. Функциональные требования определяют действия, которые должна выполнять система, без учета ограничений, связанных с ее реализацией. Нефункциональные требования не определяют поведение системы, но описывают ее свойства или атрибуты системного окружения. Например, нефункциональными являются требования к производительности системы и требования к аппаратуре, на которую может быть установлена система.

Требования оформляются в виде ряда документов и моделей. К основным документам относятся: *концепция, глоссарий предметной области, дополнительная спецификация.*

Концепция определяет глобальные цели проекта и основные особенности разрабатываемой системы. Существенной частью концепции является постановка задачи разработки, определяющая требования к функциям системы. Словарь предметной области (глоссарий) определяет общую терминологию для всех моделей и описаний требований к системе. Дополнительная спецификация (технические требования) содержит описание нефункциональных требований к системе, таких, как надежность, удобство использования, производительность, сопровождаемость и др.

Постановка задачи разработки системы регистрации курсов:

Перед руководителем информационной службы университета ставится задача разработки новой клиент-серверной системы регистрации студентов взамен старой системы на мэйнфрейме. Пользователями новой системы будут студенты, профессора и регистраторы – работники учебного отдела, управляющие ходом регистрации.

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 чтения, без обновления. В начале каждого семестра студенты могут запросить список курсов, предлагаемых в данном семестре. Информация о каждом курсе должна включать название, его длительность, время проведения занятий (день недели и номер пары), имя лектора (если назначен) и требования к предварительному уровню подготовки (т. е. перечень курсов, которые необходимо пройти прежде, чем слушать данный курс).

Новая система должна позволять студентам выбирать 4 курса в предстоящем семестре. В дополнение, каждый студент может указать 2 альтернативных курса на тот случай, если какой-либо из выбранных им курсов окажется уже заполненным или отмененным. На каждый курс может записаться не более 10 и не менее 3 студентов (если менее 3, то курс будет отменен). Студент может регистрироваться на курс только в том случае, если им выполнены требования к предварительному уровню подготовки. До окончания регистрации студенты могут изменить свои планы. В это время студенты должны иметь доступ к системе, чтобы добавить или удалить выбранные курсы. После того, как процесс регистрации некоторого студента завершен, система регистрации направляет информацию в расчетную систему, функционирующую в бухгалтерии, чтобы студент мог внести плату за семестр. Если курс окажется заполненным в процессе регистрации, студент должен быть извещен об этом до окончательного формирования его учебного плана-графика, лишние заявки на курс не принимаются.

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

Профессора должны иметь доступ к онлайн-системе, чтобы выбрать курсы, которые они будут читать. Для каждого предлагаемого курса указывается день недели и номер пары.

Регистрация на отдельный курс может быть закончена раньше по решению регистратора. Курс считается отмененным по окончании регистрации, если записалось менее 3 студентов. Регистрация на все курсы закрывается регистратором спустя две недели после ее начала. По окончании регистрации профессору доступен список студентов, записавшихся на его курсы. Кроме этого, профессора должны иметь возможность проставить оценки за курсы.

Глоссарий предметной области создается на основе описания системы регистрации вуза и постановки задачи, а также глоссария деятельности предприятия, созданного в ходе бизнес-моделирования, и концепции. Глоссарий предназначен для описания терминологии области, в которой будет работать ПО. Выделяются термины, им дается описание, рассчитанное на широкий круг читателей (пользоваться этим описанием будут все лица, заинтересованные в разработке системы). Глоссарий составляется на русском языке. Термины сопровождаются переводом на английский на тот случай, если термин будет использован в модели системы как название класса, пакета и т. п.

Глоссарий:

Каталог курсов (Course Catalog)	Внешняя система, у которой можно запросить перечень всех курсов университета.
Курс (Course)	Учебный курс по некоторому предмету, который может быть прочитан в университете. Курсы различаются названиями, длительностью. У курса могут быть требования на предварительно прослушанные курсы.
Оценка (Mark)	Количество баллов (от 2 до 5), полученных студентом за конкретный курс.
План-график (Schedule)	Набор предлагаемых курсов, выбранных студентом в некотором семестре. План-график включает в себя 4 основных и 2 альтернативных курса.
Предлагаемый курс (Course Offering)	Запись о курсе, предлагаемом для чтения в конкретном семестре (один и тот же курс может вестись в нескольких разных семестрах). Включает день недели и номер пары, когда будут проходить лекции.
Профессор (Professor)	Пользователь системы регистрации. Лектор произвольного количества курсов в течение семестра. Отмечает в системе читаемые им курсы, ставит оценки.
Расчетная система (Billing System)	Внешняя система, в которую передаются сведения для формирования счетов за обучение.
Регистратор (Registrar)	Пользователь системы регистрации. Работник учебного отдела, управляющий процессом регистрации на курсы.
Регистрация на курсы (Registration)	Процесс привязки студентов и профессоров к курсам, предлагаемым в семестре. Длится 2 недели.
Список курса (Roster)	Список всех студентов, записавшихся на некоторый предлагаемый курс.
Студент (Student)	Пользователь системы регистрации. Учащийся, который выбирает перечень курсов для обучения в течение семестра. Основные сведения о студенте - имя и почтовый адрес (для отправки счетов за обучение).

Табель успеваемости (Report Card)	Все оценки за все курсы, полученные студентом в некотором семестре.
-----------------------------------	---

Дополнительная спецификация определяет нефункциональные требования к системе, такие, как надежность, удобство использования, производительность, сопровождаемость, а также ряд функциональных требований, являющихся общими для нескольких вариантов использования. Назначение дополнительных спецификаций -- определить требования к системе регистрации курсов, которые не отражены в других документах и моделях. Вместе они образуют полный набор требований к системе. Рассмотрим дополнительную спецификацию:

1. **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.

Система должна обеспечивать выполнение следующих правил:

- Студент может регистрироваться на курс только в том случае, если им выполнены требования к предварительному уровню подготовки.
- На каждый курс может записаться не более 10 и не менее 3 студентов.
- На процесс регистрации отводится две недели.
- Курс считается отмененным по окончании регистрации, если на него записалось менее 3 студентов. И т. п.

2. **Требования по реализации**

Система должна быть совместима с Windows.

3. **Надежность**

Система должна быть в работоспособном состоянии 24 часа в день 7 дней в неделю, время простоя -- не более 10%.

4. **Производительность**

Система должна поддерживать до 2000 одновременно работающих пользователей.

5. **Безопасность**

Система не должна позволять студентам изменять любые учебные графики, кроме своих собственных (за текущий семестр), а также не должна позволять профессорам выбирать для чтения предлагаемые курсы лекций, уже выбранные другими профессорами. Только профессора имеют право ставить студентам оценки только за те курсы, которые они читают. Только регистратор может изменять любую информацию о студентах и профессорах.

6. **Проектные ограничения**

Система должна быть интегрирована с существующей системой каталога курсов, функционирующей на основе реляционной СУБД. Система должна поддерживать протокол обмена данных с расчетной системой.

Функциональные требования к системе моделируются и документируются с помощью вариантов использования (use case). Вариант использования (use case) - связный элемент функциональности, предоставляемый системой при взаимодействии с действующими лицами. Действующее лицо (actor) -- роль, обобщение элементов внешнего окружения системы, ведущих себя по отношению к системе одинаковым образом.

В контексте процесса управления требованиями варианты использования трактуются следующим образом:

- вариант использования фиксирует соглашение между участниками проекта относительно поведения системы;
- вариант использования описывает поведение системы при различных условиях, когда система отвечает на запрос одного из участников, называемого основным действующим лицом;
- основное действующее лицо инициирует взаимодействие с системой, чтобы добиться некоторой цели. Система отвечает, соблюдая интересы всех действующих лиц.

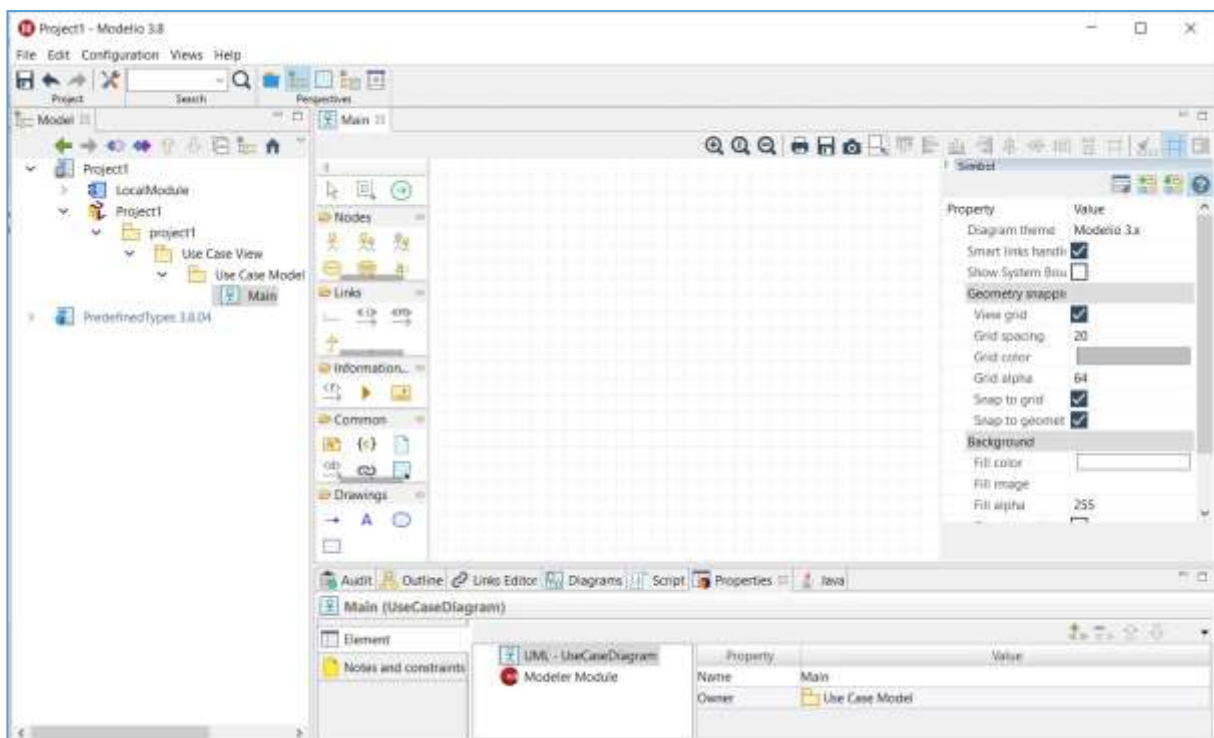
Модель вариантов использования состоит из диаграмм вариантов использования, текстовых описаний вариантов использования и диаграмм деятельности, моделирующих потоки событий вариантов использования. Диаграмма вариантов использования составляется системным аналитиком, который сначала выявляет элементы модели, а затем устанавливает связи между ними. Элементами диаграмм вариантов использования являются варианты использования и действующие лица, соединенные разного рода связями.

Из постановки задачи разработки и глоссария системный аналитик может выделить список действующих лиц и их интересов:

- Студент – регистрируется на курсы, смотрит свой табель;
- Профессор – выбирает курсы для преподавания, ставит оценки;
- Регистратор – управляет процессом регистрации, ведет (т. е. вводит, изменяет, удаляет) данные о профессорах и студентах;
- Расчетная система – получает данные, нужные для формирования счетов за обучение;
- Каталог курсов – предоставляет данные об университетских курсах.

Упражнение 3.1. Создание действующих лиц

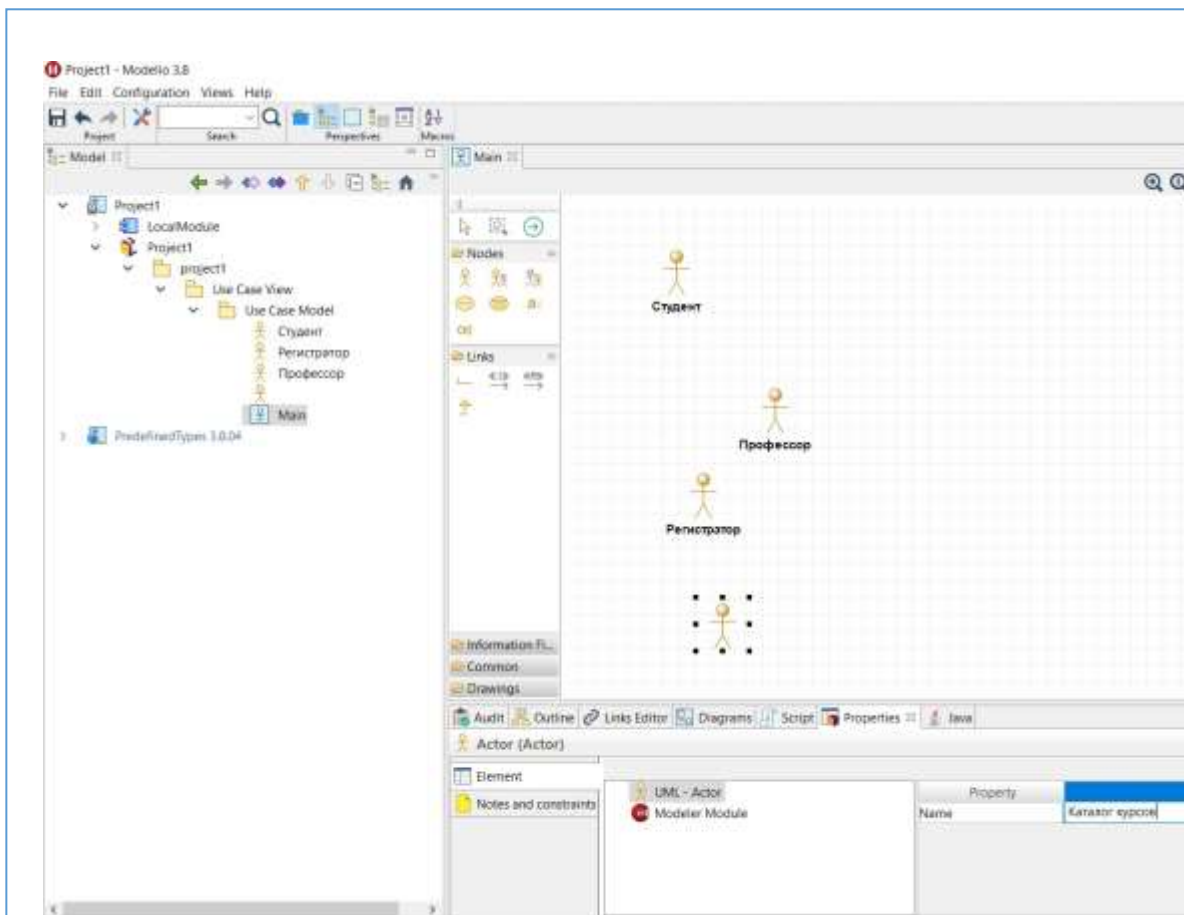
В браузере находим пакет Use Case View, а внутри него создаем пакет «Use Case Model». Выделим Use Case Model и вызовем контекстное меню Create diagram -> Create a Use Case diagram. Дадим диаграмме имя Main (переименование можно осуществлять с помощью F2).



Дадим имя Main диаграмме. Диаграмма открылась в окне редактора и появилась палитра с элементами диаграмм.

В палитре редактора выберем элемент Actor и добавим действующее лицо на диаграмму. Введем имя актора: Студент.

Повторим те же действия и добавим оставшихся действующих лиц: профессора, регистратора, каталог курсов, расчетную систему.



Исходя из потребностей действующих лиц, системный аналитик может предложить следующие варианты использования: Войти в систему, Зарегистрироваться на курсы, Просмотреть таблицу, Выбрать читаемые курсы, Поставить оценки, Открыть регистрацию, Закрыть регистрацию, CRUD данных о профессорах, CRUD данных о студентах. CRUD расшифровывается как Create, Read, Update, Delete (или как Create, Retrieve, Update, Destroy). Предполагается, что вариант использования «CRUD данных о профессорах» описывает все функции системы, предоставляемые регистратору для управления сведениями о профессорах.

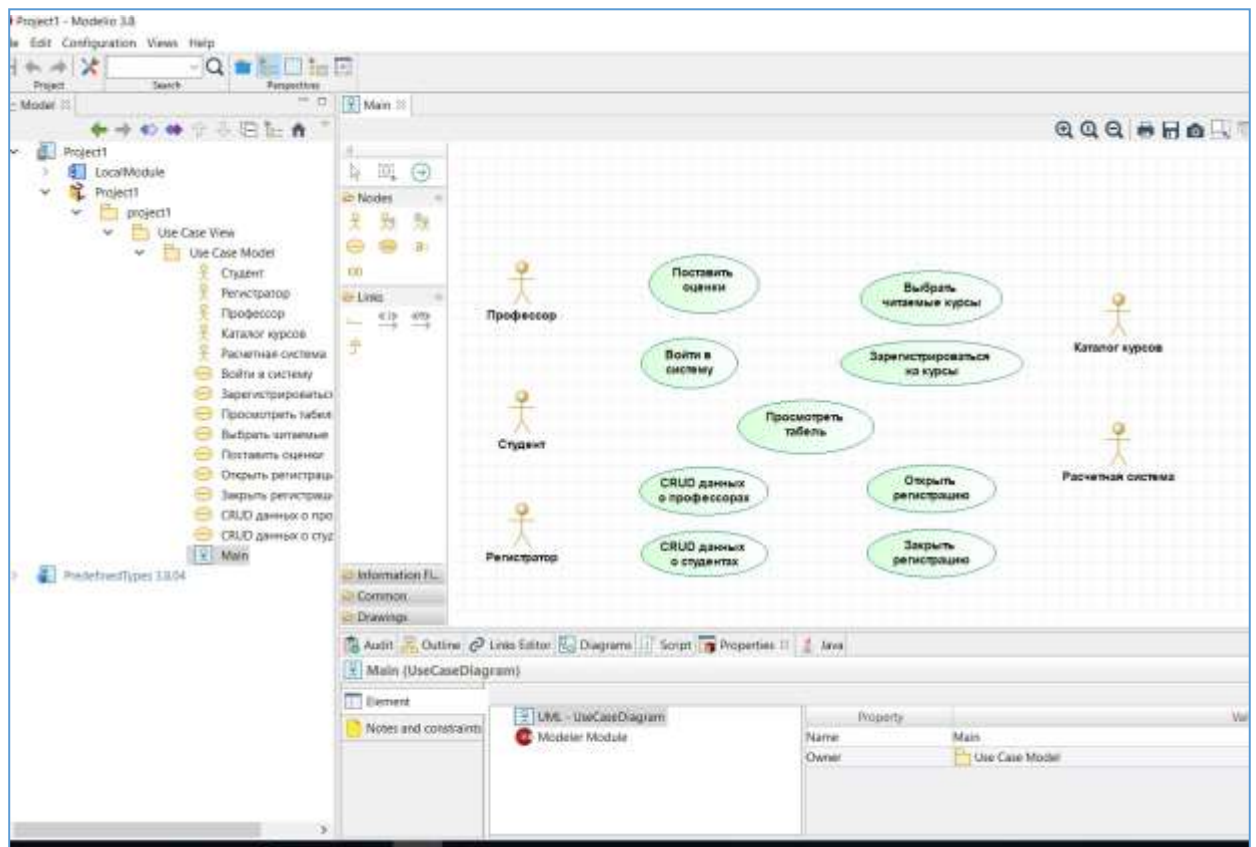
Вариант использования «Войти в систему» не соответствует какой-либо явной цели какого-нибудь действующего лица. Он описывает функциональные требования к системе по обеспечению защиты данных. Про каждого своего пользователя система должна знать, какую роль по отношению к ней он выполняет: студента, профессора или регистратора, к каким функциям и данным ему можно предоставлять доступ.

Упражнение 3.2. Создание вариантов использования и связей между ними

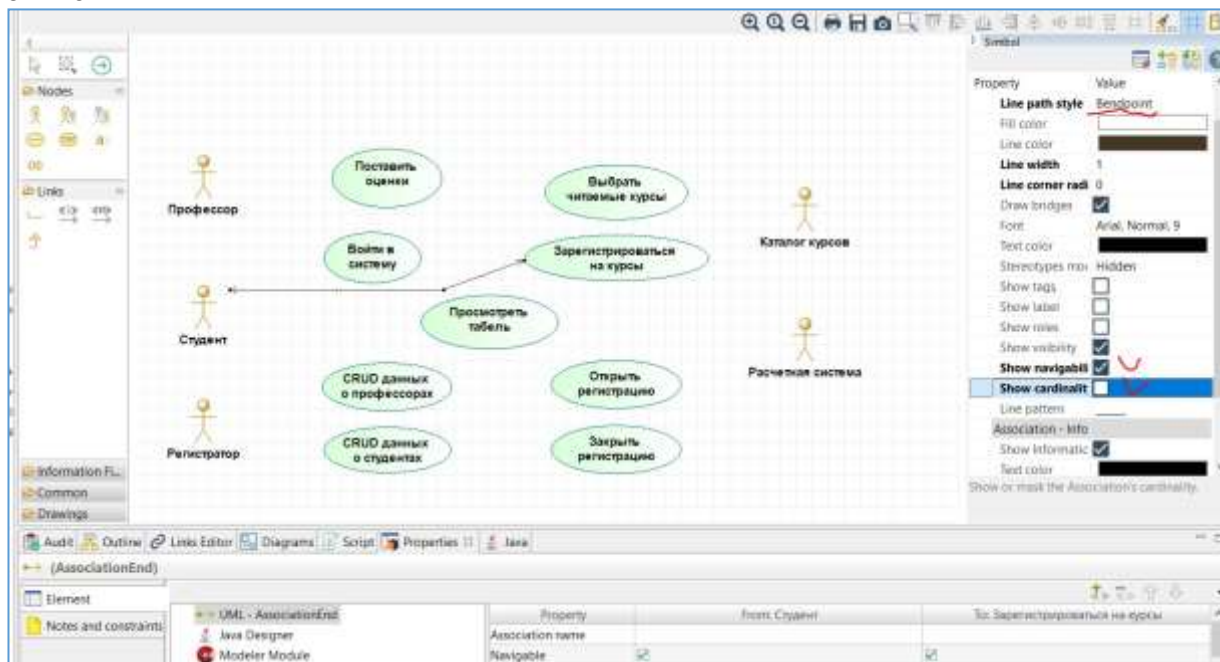
В палитре редактора выберем элемент Use Case и добавим вариант использования на диаграмму. Введем название: Зарегистрироваться на курсы.

Повторим те же действия для добавления оставшихся вариантов использования: Войти в систему, Просмотреть таблицу, Выбрать читаемые курсы, Поставить оценки, CRUD данных о профессорах, CRUD данных о студентах, Открыть регистрацию, Закрыть регистрацию.

Размещаем варианты примерно так, как показано на рисунке.

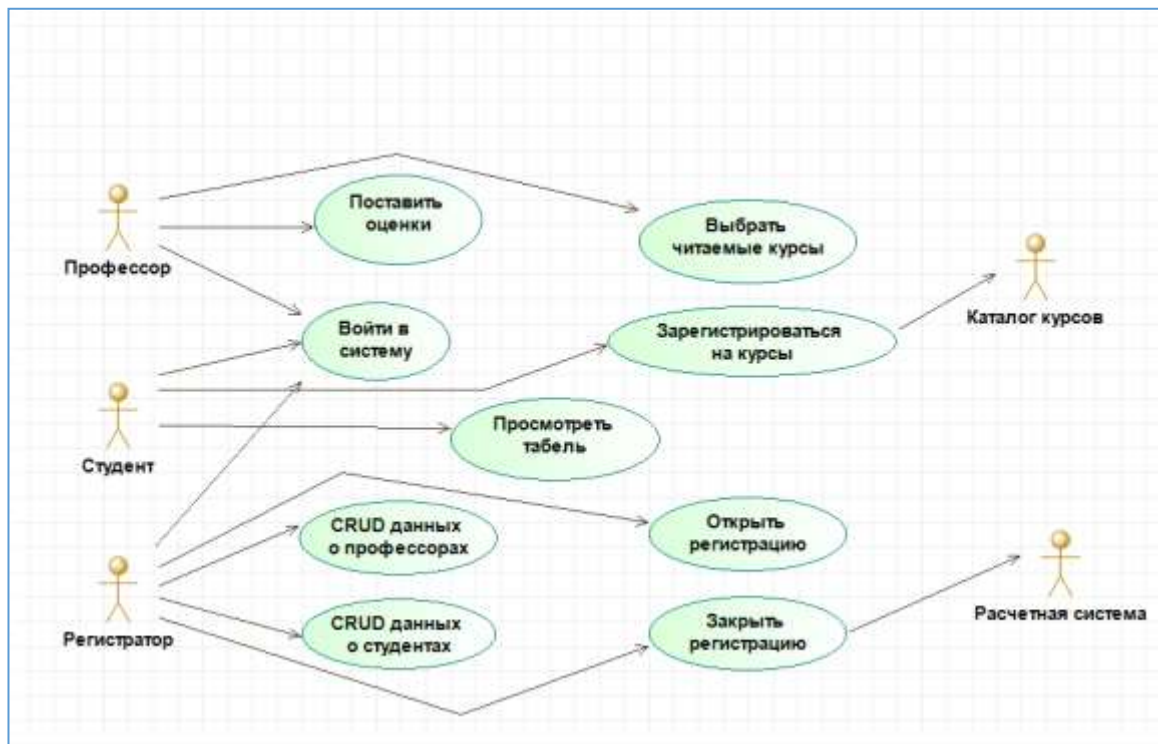


В палитре редактора выберем связь Communication Link и проведем связь от действующего лица Студент к варианту использования Зарегистрироваться на курсы. В свойствах отображения (вкладка Symbol, Association Show navigability) зададим отображение направление, поставив галку. Также можно указать тип линии (Line path style): прямая (Direct), ломанная (Bendpoint), прямоугольная ломанная (Orthogonal). Можно управлять выводом мощностей связи (Association Show cardinality). Если мощность выводится, следует ее отключить.



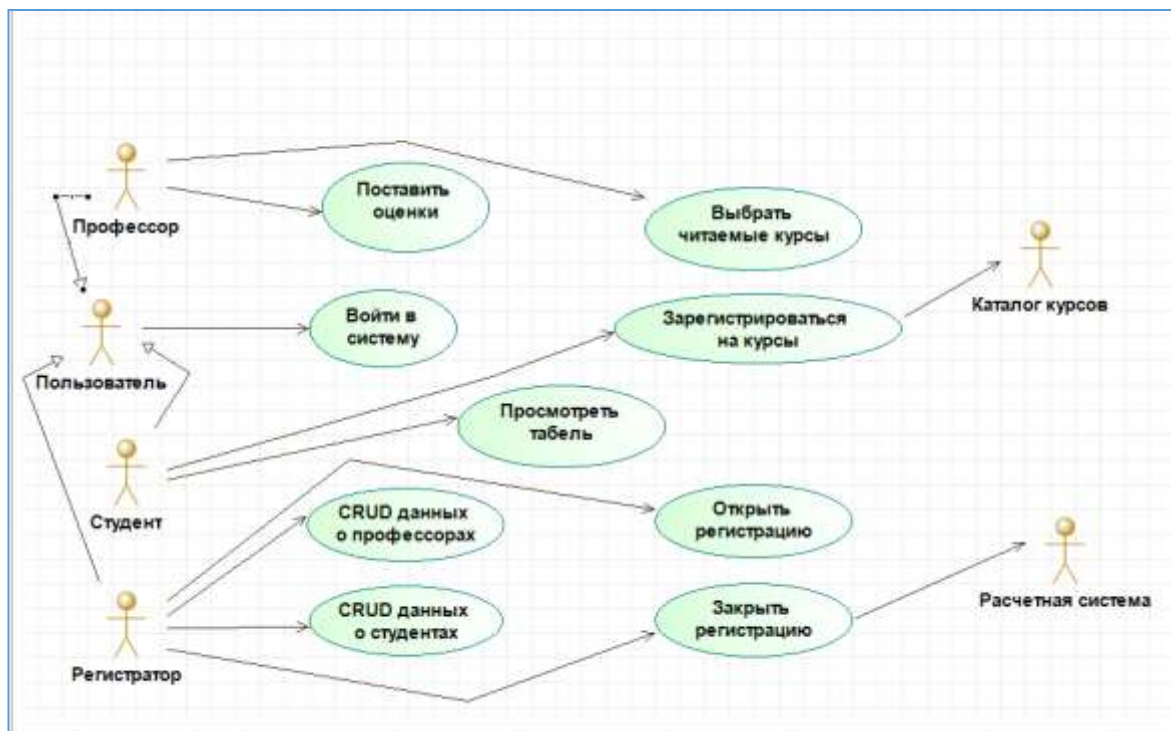
Повторим те же действия для добавления связей других действующих лиц и вариантов использования. Должна получиться диаграмма схожая с изображением на рисунке ниже. Направления связей между действующими лицами и вариантами использования показывают, какое лицо является основным, а какое второстепенным. Например, для

варианта использования Зарегистрироваться на курсы Студент - основное действующее лицо (он инициирует запуск варианта использования), а Каталог курсов - второстепенное действующее лицо (в ходе регистрации из каталога запрашиваются данные о курсе).



Созданная диаграмма имеет недостаток в том, что у варианта использования «Войти в систему» несколько основных действующих лиц. Полагая поведение системы одинаковым при входе любого пользователя, введем действующее лицо Пользователь, подвидами которого будут лица Студент, Профессор, Регистратор.

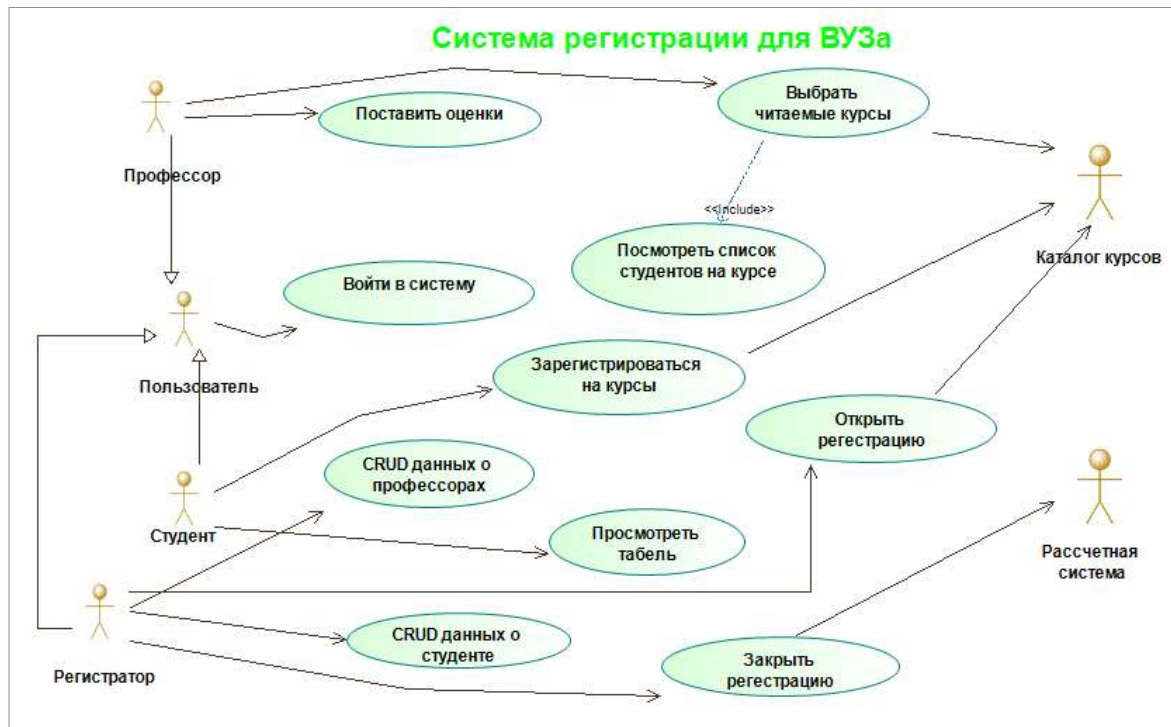
Диаграмма примет вид:



Для добавления связей обобщения используйте связь Generalization. Лишние ассоциации удалите из модели (Del). Обратите внимание, что нажатие на Del удаляет элемент и с

диаграммы, и из модели. Если необходимо сохранить элемент в модели и всего лишь убрать его с диаграммы, на которую он случайно попал по какой-то причине, нажимайте Ctrl+M или используйте пункт контекстного меню Mask Selection. Следует четко понимать удаляете ли Вы текущий элемент или всего лишь маскируете. Удаление вместо маскирования (как и обратное) может быть нежелательным. Случайно удаленный элемент можно попытаться восстановить, нажав Ctrl+Z. Маскированный по ошибке элемент можно снова поместить на диаграмму, перетаскивая его из браузера.

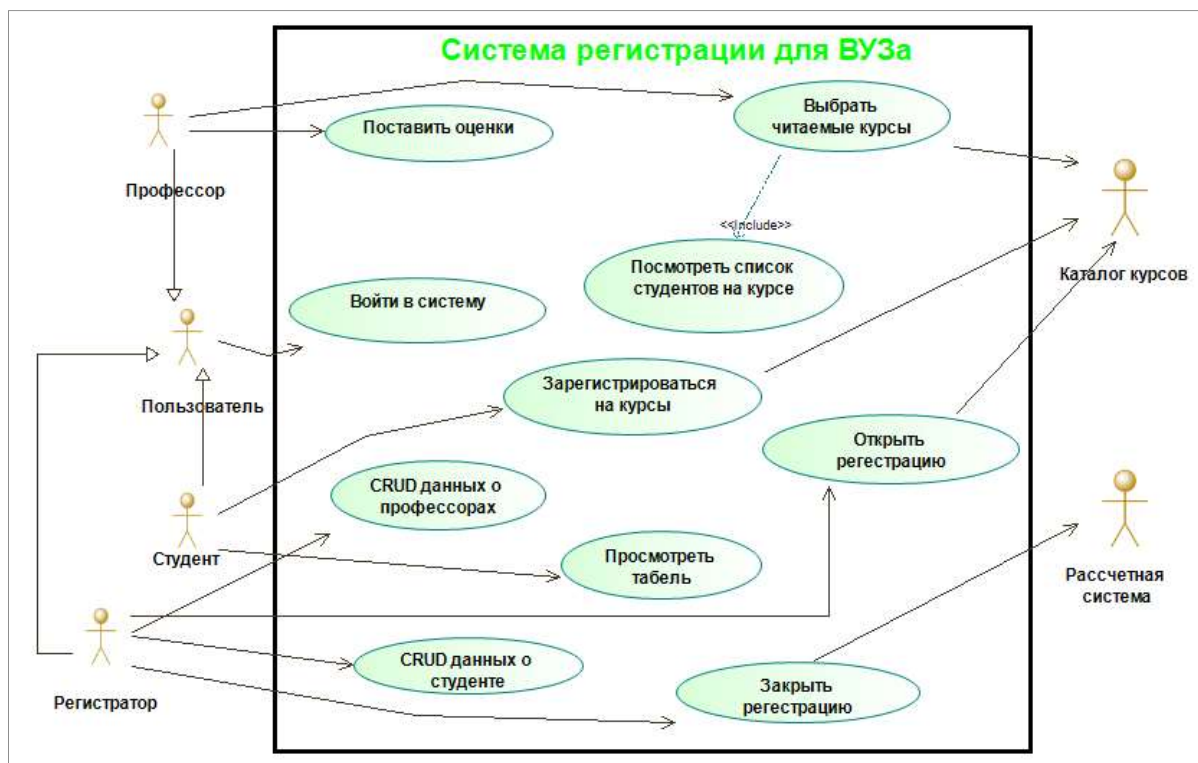
Добавим вариант использования на диаграмму. Введем название: Посмотреть список студентов на курсе. В палитре редактора выберем связь Inclusion relationship и проведем связь от варианту использования «Выбрать читаемые курсы» к варианту использования «Посмотреть список студентов на курсе»



Теперь на нашей диаграмме каждый вариант использования связан ровно с одним действующим лицом, которое для него является основным, и с произвольным количеством дополнительных действующих лиц. Каждое действующее лицо связано хотя бы с одним вариантом использования. Никакие два варианта использования между собой не связаны ассоциацией. Связи между вариантами использования допускаются, если они имеют тип, отличный от ассоциации. Это могут быть связи обобщения, включения и расширения. О них будет рассказано на лекции. С их помощью производится структурирование модели вариантов использования.

Модель с большим количеством вариантов использования может быть разбита на части – пакеты. Никакие действующие лица не связаны между собой ассоциацией. Между ними допускаются лишь связи обобщения. Самостоятельно создавая диаграммы вариантов использования, старайтесь, чтобы их элементы и связи удовлетворяли таким же ограничениям. Также рекомендуется основных действующих лиц размещать в левой части диаграммы, а второстепенных – на правой.

Завершим создание диаграммы выделением области вариантов использования и обозначением, что они относятся к Системе регистрации для ВУЗа (элементы из Drawings с изменением параметров).



Получившаяся диаграмма является примерным оглавлением проекта. Мы видим, какие функции и кому предоставляет система. По ней архитектор проекта может составить оценку трудоемкости всего проекта, основанную на прогнозируемой трудоемкости реализации каждого из вариантов использования. Также архитектор ранжирует варианты использования по приоритету. Отмеченные им более важные варианты использования будут реализованы в первую очередь.

Для каждого варианта использования составляется описание.

Упражнение 3.3. Добавление описаний вариантов использования

Функциональные требования подробно фиксируются в описаниях вариантов использования. Описания составляются специальным образом, чтобы уменьшить вероятность неверного толкования и облегчить восприятие текста. Каждое описание включает в себя:

- а) краткое описание, являющееся сжатым обзором варианта использования;
- б) основной поток событий, описывающий взаимодействие системы и действующих лиц, при котором достигается цель основного действующего лица;
- в) альтернативные потоки, описывающие обработку ошибок и исключительных ситуаций;
- г) подчиненные потоки, которые облегчают описание основного и альтернативных потоков;
- д) предусловия (которые могут отсутствовать) и постусловия (которые всегда есть).

Каждый поток событий задается перенумерованным набором шагов. Используются шаги трех типов: действие системы (например, «Система запрашивает имя пользователя и пароль»); реакция действующего лица («Пользователь вводит имя и пароль»); управление потоком («Выполнение переходит на начало основного потока»). Структура предложений, описывающих шаги, одинакова: подлежащее, сказуемое, остальные части речи. От нее отходят лишь при описании циклов и ветвлений.

Цикл задается составным описанием, в начале которого указывается условие цикла («Для каждого незафиксированного курса в графике выполняется») или количество повторений. Далее следует тело цикла -- последовательность вложенных шагов (см. подчиненный поток «8Г. Утвердить график» варианта использования «Зарегистрироваться на курсы»).

Ветвление в тривиальных случаях, когда альтернативная ветвь пуста, допускается описывать предложением с союзом *если* («Система помечает курс как закрытый, если в списке студентов содержится 10 записей»). Чаще ветвление описывают с помощью альтернативных потоков. В основном потоке варианта использования «Войти в систему» 3-ий шаг указывает основное продолжение потока, а альтернативный поток «3А. Неправильное имя/пароль» содержит второй вариант развития событий. Как правило, действия по проверке условия ветвления не описывают. Вместо этого указывается шаг, на котором система (или

действующее лицо) подтверждает, что условие выполнено, в основном потоке, или обнаруживает, что условие нарушено, в альтернативном потоке.

Добавьте в модель описания, приведенные ниже. Для этого скопируйте текст описания, выделите нужный вариант использования, откройте вкладку Notes and constraints, находящуюся по соседству с вкладкой Element и вставьте текст в поле Description. Обратите внимание, что в описании любого варианта использования должен присутствовать хотя бы один шаг для каждого действующего лица, связанного с ним. Постусловие в описании не может отсутствовать, так как по нему тестировщики составляют тестовые сценарии. Постусловие состоит из двух частей: гарантии успеха и минимальных гарантий. Первая часть описывает, что должно быть истинно при успешном завершении варианта использования. Вторая часть -- это гарантии системы в случаях неуспеха. Например, при успешном входе в систему гарантируется доступ пользователя к главному меню, при неуспешном -- гарантируется, что такой доступ не будет предоставлен. Пустое постусловие допускало бы произвольное поведение системы и было бы не пригодно для тестирования.

Описания составляются для всех вариантов использования. Выполняя упражнения, мы создадим лишь три описания.

Вариант использования «Войти в систему»:

Краткое описание

Данный вариант использования описывает вход пользователя в систему регистрации курсов.

Основной поток событий

1. Система запрашивает имя пользователя и пароль.
2. Пользователь вводит имя и пароль.
3. Система подтверждает правильность имени и пароля, определяет тип пользователя (студент, профессор или регистратор) и выводит главное меню, дающее доступ к функциям системы в соответствии с типом пользователя.

Альтернативные потоки

3А. Неправильное имя/пароль

1. Система обнаруживает, что комбинация имени и пароля не верна.
2. Система сообщает об ошибке и предлагает пользователю либо заново ввести имя и пароль, либо отказаться от входа в систему.
3. Пользователь сообщает системе свой выбор.
4. В соответствии с выбором пользователя либо выполнение переходит на начало основного потока, либо вариант использования завершается.

Предусловия

Отсутствуют.

Постусловия

Если вариант использования выполнен успешно, система предоставляет доступ к главному меню пользователю, сообщившему верную комбинацию имени и пароля. В противном случае система гарантирует, что пользователю, сообщившему неверную комбинацию имени и пароля, доступ к меню не будет предоставлен.

Обратите внимание на номер альтернативного потока. Цифра указывает номер шага основного потока, на котором может произойти переключение на альтернативный поток, буква позволяет различить несколько альтернативных потоков, на которые можно переключиться на одном и том же шаге. Если переход на альтернативный поток может происходить в течение нескольких подряд идущих шагов, указывают их номера через дефис. Например, 1-3Б. Если поток вызывается из разных шагов, он может иметь несколько номеров, перечисленных через запятую.

Вариант использования «Зарегистрироваться на курсы»:

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

1. Студент сообщает о желании зарегистрироваться на курсы.
2. Система подтверждает, что регистрация на курсы в текущем семестре открыта.
3. Система запрашивает связь с каталогом курсов.
4. Каталог курсов подтверждает, что связь установлена.
5. Система запрашивает требуемое действие (создать график, обновить график, удалить график, утвердить график).
6. Студент сообщает системе свой выбор.
7. Система подтверждает, что требуемое действие можно выполнить.
8. Согласно выбору студента выполняется один из подчиненных потоков (создать, обновить, удалить или утвердить график).
9. Система заканчивает сеанс связи с каталогом курсов.
10. Каталог курсов подтверждает, что сеанс закончен.

Подчиненные потоки событий:

8А. Создать график

1. Система запрашивает у каталога курсов список доступных конкретных курсов.
2. Каталог курсов передает системе запрашиваемый список.
3. Система выводит список курсов и пустой план-график.
4. Студент выбирает из списка 4 основных курса и 2 альтернативных курса.
5. Система создает график студента и заносит в него выбранные курсы, помечая их как незафиксированные.
6. Система сообщает, что создание графика завершено.
7. Выполнение переходит на шаг 9 основного потока.

8Б. Обновить график

1. Система выводит текущий график студента.
2. Система запрашивает у каталога курсов список доступных конкретных курсов. Каталог курсов передает системе запрашиваемый список.
3. Система выводит список курсов.
4. Студент обновляет свой выбор курсов, удаляя или добавляя конкретные курсы.
5. Система обновляет график в соответствии с пожеланиями студента. Для каждого зафиксированного курса, удаленного из графика, система удаляет студента из списка студентов, записавшихся на курс. Каждый добавленный курс система помечает как незафиксированный.
6. Система сообщает, что обновление графика завершено.
7. Выполнение переходит на шаг 9 основного потока.

8В. Удалить график

1. Система выводит текущий график студента.
2. Система запрашивает у студента подтверждения удаления графика.
3. Студент подтверждает удаление.
4. Система удаляет график. Для каждого зафиксированного курса из удаляемого графика, система удаляет студента из списка студентов, записавшихся на курс.
5. Выполнение переходит на шаг 9 основного потока.

8Г. Утвердить график

1. Для каждого незафиксированного курса в графике выполняется:
 - 1.1. Система подтверждает выполнение студентом предварительных требований (прохождение определенных курсов), и подтверждает, что курс открыт для регистрации, и отсутствуют конфликты (в графике не должно быть зафиксированного курса, читаемого в тот же день и на той же паре, что и проверяемый курс).
 - 1.2. Система добавляет студента в список записавшихся на курс.
 - 1.3. Система помечает курс в графике как зафиксированный.
 - 1.4. Система помечает курс как закрытый, если в списке студентов содержится 10 записей.
2. Система сообщает студенту результаты утверждения графика.
3. Выполнение переходит на шаг 9 основного потока.

Альтернативные потоки

2А. Регистрация на курсы закрыта

1. Система обнаруживает, что регистрация на курсы в текущем семестре закрыта.
2. Система выдает сообщение об ошибке.

3. Вариант использования завершается.

3А. Каталог курсов недоступен

1. Система обнаруживает, что невозможно установить связь с каталогом курсов.
2. Система выдает сообщение об ошибке.
3. Вариант использования завершается.

8Г.1.1А. Не выполнены предварительные требования, курс заполнен, или имеют место конфликты графика

1. Система обнаруживает, что студент не выполнил необходимые предварительные требования, или выбранный им конкретный курс заполнен, или имеют место конфликты графика.
2. Система выдает сообщение об ошибке.
3. Система переходит к следующему незафиксированному курсу и продолжает выполнение потока «Утвердить график»

7А. График не найден

1. Система обнаруживает, что требуемое действие (обновить, удалить или утвердить график) нельзя выполнить, так как график студента на текущий семестр отсутствует.
2. Система выдает сообщение об ошибке.
3. Выполнение переходит на шаг 5 основного потока.

7Б. График найден

1. Система обнаруживает, что требуемое действие (создать график) нельзя выполнить, так как график студента на текущий семестр создан ранее.
2. Система выдает сообщение об ошибке.
3. Выполнение переходит на шаг 5 основного потока.

8В.3А. Удаление отменено

1. Студент отменяет удаление графика.
2. Выполнение переходит на шаг 5 основного потока.

Предусловия

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

Постусловия

Если вариант использования завершится успешно, система создаст, обновит, удалит или утвердит график студента в соответствии с выбором пользователя. В противном случае гарантируется что: при закрытой регистрации изменения в графики студентов не производятся; при недоступном каталоге курсов изменения в графики студентов не производятся; при утверждении графика студента игнорируются курсы, для которых не выполнены предварительные требования, или которые закрыты, или которые вызывают конфликты в графике; при отсутствии графика на текущий семестр его обновление, удаление или утверждение не производятся; при наличии графика на текущий семестр добавление еще одного графика не производится.

Вариант использования «Закрыть регистрацию»:

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

1. Регистратор запрашивает прекращение регистрации.
2. Система подтверждает возможность закрыть регистрацию и фиксирует, что регистрация закрыта.
3. Для каждого предлагаемого курса, который открыт для регистрации, выполняется:
 - 3.1. Система подтверждает, что курс взятся провести какой-либо профессор, и что на курс записалось не менее трех студентов.
 - 3.2. Для каждого графика студента, в котором курс помечен основным и не зафиксированным, выполняется подчиненный поток «Зафиксировать курс в графике».

4. Для каждого студенческого графика система проверяет наличие в нем 4 зафиксированных курсов; если их недостаточно, система дополняет график альтернативными курсами по схеме, описанной в шаге 3 основного потока.
5. Для каждого открытого предлагаемого курса выполняется:
 - 5.1. Система подтверждает, что в списке не менее трех студентов.
 - 5.2. Система помечает курс как закрытый.
6. Система запрашивает связь с расчетной системой.
7. Расчетная система подтверждает готовность к приему данных.
8. Система передает в расчетную систему графики студентов.
9. Расчетная система подтверждает прием графиков студентов.

Подчиненные потоки:

3.2. Зафиксировать курс в графике

1. Система подтверждает, что курс открыт для регистрации.
2. Система добавляет студента в список записавшихся на курс.
3. Система помечает курс в графике как зафиксированный.
4. Система помечает курс как закрытый, если в списке студентов содержится 10 записей.

Альтернативные потоки:

2А. Регистрация не может быть прекращена

1. Система обнаруживает, что процесс регистрации нельзя прекратить немедленно.
2. Система выдает сообщение регистратору и предлагает выбрать повтор или отмену закрытия регистрации.
3. Регистратор сообщает свой выбор.
4. Система в соответствии с выбором либо продолжает выполнение шага 2 основного потока, либо завершает выполнение варианта использования.

3.1А. Курс никто не ведет

1. Система помечает курс как отмененный.
2. Система исключает данный курс из каждого содержащего его графика и удаляет студентов из списка записавшихся на курс.
3. Система выбирает следующий курс и продолжает выполнение шага 3 основного потока.

3.1Б. На курс записалось мало студентов

1. Система выбирает следующий курс и продолжает выполнение шага 3 основного потока.

3.2.1 А Регистрация на курс закрыта

1. Выполнение подчиненного потока «Зафиксировать курс в графике» завершается.

5.1А. Менее трех студентов на курсе

1. Пока в списке меньше 3 студентов и есть графики, в которых меньше 4 зафиксированных курсов выполняется подчиненный поток «Зафиксировать курс в графике»
2. Система подтверждает, что на курсе 3 или более студентов.
3. Продолжается выполнение основного потока с шага 5.2.

5.1А.2А. Невозможно добавить студентов на курс

1. Система обнаруживает, что на курсе 2 или менее студентов.
2. Система помечает курс как отмененный.
3. Система вычеркивает его из всех графиков и переходит к следующему открытому курсу.
3. Продолжается выполнение основного потока с шага 5.

7А. Расчетная система недоступна

1. Система ожидает некоторое установленное время.
2. Выполнение передается на шаг 6 основного потока.

Предусловия

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

Постусловия

Если вариант использования завершится успешно, регистрация закрывается, графики студентов передаются в расчетную систему. В противном случае регистрация остается открытой.

В проекте , например, для ВИ «войти в систему», должно быть (закладку Exceptions тоже заполнить):

The screenshot shows a software development tool window titled "Edition of Войти в систему Use Case". It has a tabbed interface with "Use Case", "Properties", "Notes and constraints", and "Audit". The "Use Case" tab is active, showing the title "Войти в систему". Below the title, there is a "Name/Id" field containing "Войти в систему". A sub-tabbed interface below shows "Use Case", "Exceptions", "Conditions", and "Constraints". The "Description" sub-tab is active, containing the text: "Данный вариант использования описывает вход пользователя в систему регистрации курсов." A "Close" button is at the bottom right.

This screenshot shows the same window, but with the "Conditions" sub-tab selected. It is divided into "Pre-conditions" and "Post-conditions". The "Pre-conditions" field contains "Отсутствуют." The "Post-conditions" field contains a detailed description: "Если вариант использования выполнен успешно, система предоставляет доступ к главному меню пользователю, сообщившему верную комбинацию имени и пароля. В противном случае система гарантирует, что пользователю, сообщившему неверную комбинацию имени и пароля, доступ к меню не будет предоставлен." A "Close" button is at the bottom right.

Упражнение 4 Сделать отчет по 1 лабораторной.

Выполнить задания лабораторной работы и предоставить в качестве ответа отчет со скринами диаграммы и скринами комментариев к вариантам использования.

Продемонстрировать процесс выполнения преподавателю на занятиях.

Использованы материалы <http://sp.cs.msu.su/ooap/exer2013.html>