

# Высокоуровневые методы информатики и программирования

*Лекция 4*

*Начало разработки проекта*

# Содержание

- Определение и обоснование целей проекта
- Система регистрации платных курсов  
Университета Истерн
- Архитектура системы
- Диаграммы действий
- Проектирование БД
- Конструирование классов

# Определение и обоснование целей проекта

Первый вопрос, который нужно задать, приступая к работе над новой системой:

«Какая именно система должна быть создана и зачем?».

Процесс формулирования целей будущей системы с наряду с обдумыванием формы ее представления и общих ограничений составляет начальную фазу жизненного цикла проекта, которая завершается заявлениями вида:  
«Создаваемая система должна...»

# АИС регистрации университетских курсов

Кратко автоматизированную систему регистрации учебных курсов Университета Истерн можно описать следующим образом. После того как преподаватели Университета решат какие курсы они будут вести в течение семестра, служба регистрации курсов внесет эту информацию в компьютерную систему. Затем для преподавателей распечатают сводный отчет по курсам, которые они будут читать, а для студентов - каталог курсов.

# АИС регистрации университетских курсов

После получения каталога курсов студенты заполняют специальную регистрационную форму, где указывают выбранные курсы, и отдают ее в службу регистрации. Обычно студенты записываются на четыре курса, после чего информация заносится в компьютер. Далее запускается ночная программа, которая распределяет студентов по курсам. При возникновении конфликтной ситуации служба регистрации уточняет студенческие данные. После успешного распределения по курсам каждому студенту выдается его расписание для проверки. Обычно процесс регистрации на курсы занимает около недели, но иногда требуется до двух недель, чтобы уладить все вопросы. Затем преподаватели получают списки студентов для каждого курса, который они будут читать в новом семестре.

# АИС регистрации университетских курсов

Как уже было сказано, в начале каждого семестра студенты Университета Истерн могут запросить у Системы регистрации каталог курсов, в который включен список учебных курсов, предлагаемых для изучения в новом семестре. Информация о курсе содержит фамилию преподавателя, название факультета и краткое описание курса, помогающее студентам сделать выбор. Система разрешает студенту выбрать четыре курса из предложенных в наступающем семестре. Кроме того, каждый студент должен указать еще два курса, на случай если один (или более) из выбранных им четырех курсов будет переполнен или отменен. На каждый курс не должно быть записано более 10-ти или менее трех студентов. Курс, на который запишутся менее трех студентов будет отменен.

# АИС регистрации университетских курсов

По завершении регистрации Система регистрации направляет информацию в систему оплаты Университета для выставления счетов студентам. Преподаватели имеют возможность онлайн-доступа к данным о курсах, которые они будут читать, а также к данным о записавшихся на них студентов

# Архитектура системы

Язык UML может быть использован для проектированию практически любой компьютерной системы и ее программного обеспечения. Программная архитектура компьютерных систем многомерна и состоит из нескольких взглядов на нее, изображения которых называются представлениями взглядов на систему. Эти одновременно развивающиеся в процессе проектирования компьютерной системы представления о ней показаны на рисунке:

# Архитектура системы

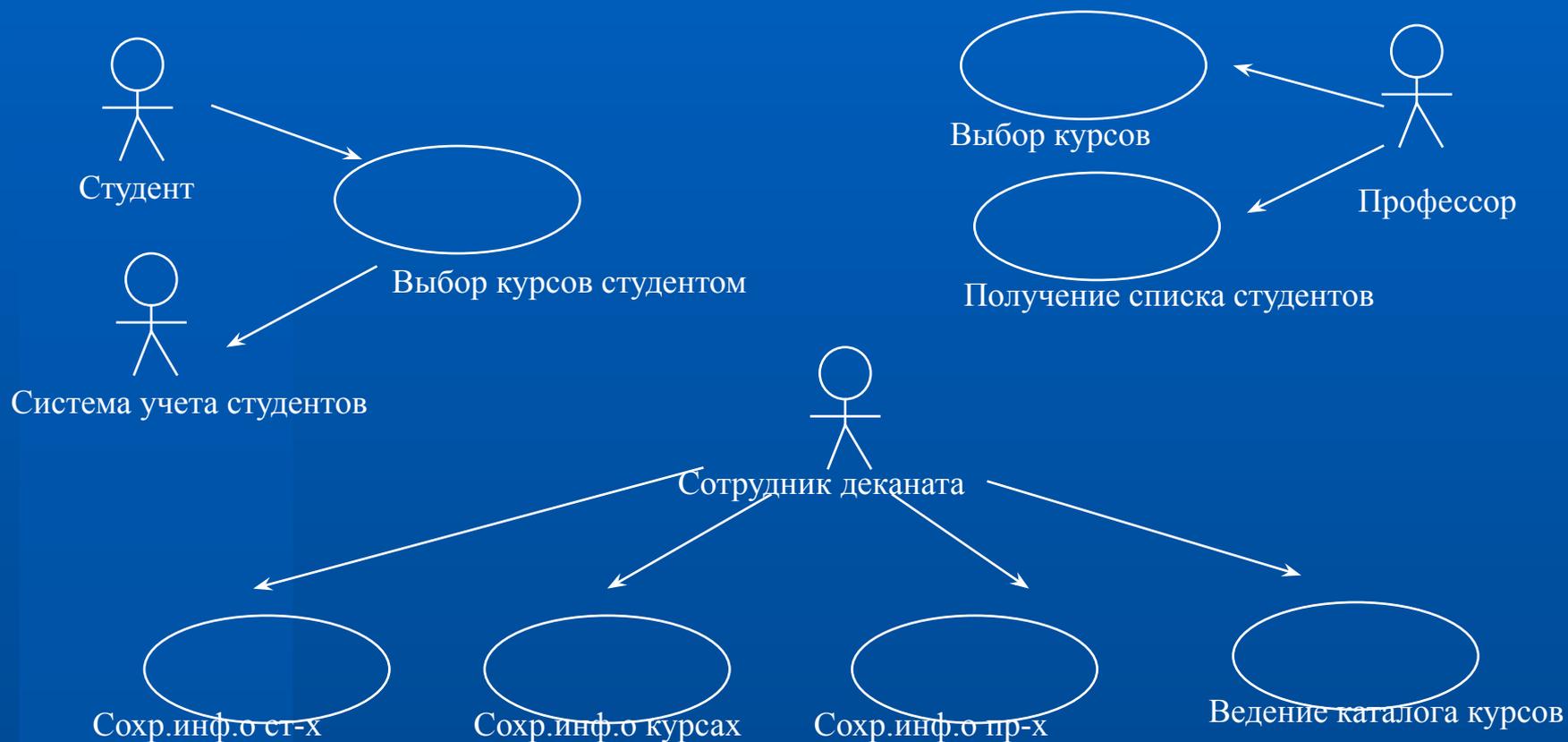


# Диаграммы действий

Проектирование любой АИС лучше всего начинать с построения диаграммы прецедентов, описывающей внешнюю границу АИС. Такая диаграмма называется главной диаграммой прецедентов.

Для АИС системы регистрации курсов Университета Истерн главная диаграмма прецедентов выглядит так:

# Диаграммы действий



# Диаграммы действий

Рассмотрим эту диаграмму подробнее. Она описывает внешнюю границу системы регистрации курсов. Как видите, внешняя граница АИС состоит из 4 актеров и 7 прецедентов (use cases). Актеры имеют имена Преподаватель, Студент, Регистратор, Система оплаты. Следует помнить, что имена Преподаватель и Студент обозначают всех преподавателей и всех студентов Университета. Система оплаты - это бухгалтерия. Имена прецедентов обозначены рядом с их пиктограммами. Физически прецеденты реализуются как внешние рабочие места АИС, оснащенные дисплеями и средствами печати. Чтобы быстро и качественно нарисовать диаграмму прецедентов можно воспользоваться программой Rational Rose.

# Диаграммы действий

После построения главной диаграммы прецедентов разработчики системы должны последовательно описать потоки событий для каждого прецедента и связанного с ним актера.

Такие описания называются спецификациями прецедентов.

# Диаграммы действий

Рассмотрим описание потока событий для прецедента "выбор курсов для преподавателя". При этом приведем описания не всех событий потока, а только нескольких начальных.

## Главный поток

Прецедент начинает выполняться, когда преподаватель подключится к системе регистрации и введет в нее свой пароль. Система проверяет правильность пароля (код операции E-1) и просит преподавателя выбрать текущий или следующий учебный семестр (E-2). Преподаватель вводит нужный семестр. Система предлагает выбрать требуемую операцию: добавить (Add), удалить (Delete), просмотреть (Review), напечатать (Print) или выйти (Quit).

# Диаграммы действий

Если выбрана операция добавить (Add), S-1: выполняется поток добавить учебный курс (Add a Course Offering).

Если выбрана операция удалить (Delete), S-2: выполняется поток удалить учебный курс (Delete a Course Offering).

Если выбрана операция просмотреть (Review), S-3: выполняется поток просмотреть расписание (Review Schedule).

Если выбрана операция напечатать (Print), S-4: выполняется поток напечатать расписание (Print Schedule).

Если выбрана операция выйти (Quit): прецедент завершается.

# Диаграммы действий

Подпоток S-1: добавить учебный курс (Add a Course Offering)

Система отображает окно, содержащее поле для ввода названия и номера предмета. Преподаватель вводит название и номер предмета (E-3). Система отображает список учебных курсов для указанного предмета (E-4). Преподаватель выбирает учебный курс. Система закрепляет за преподавателем выбранный учебный курс (E-5).

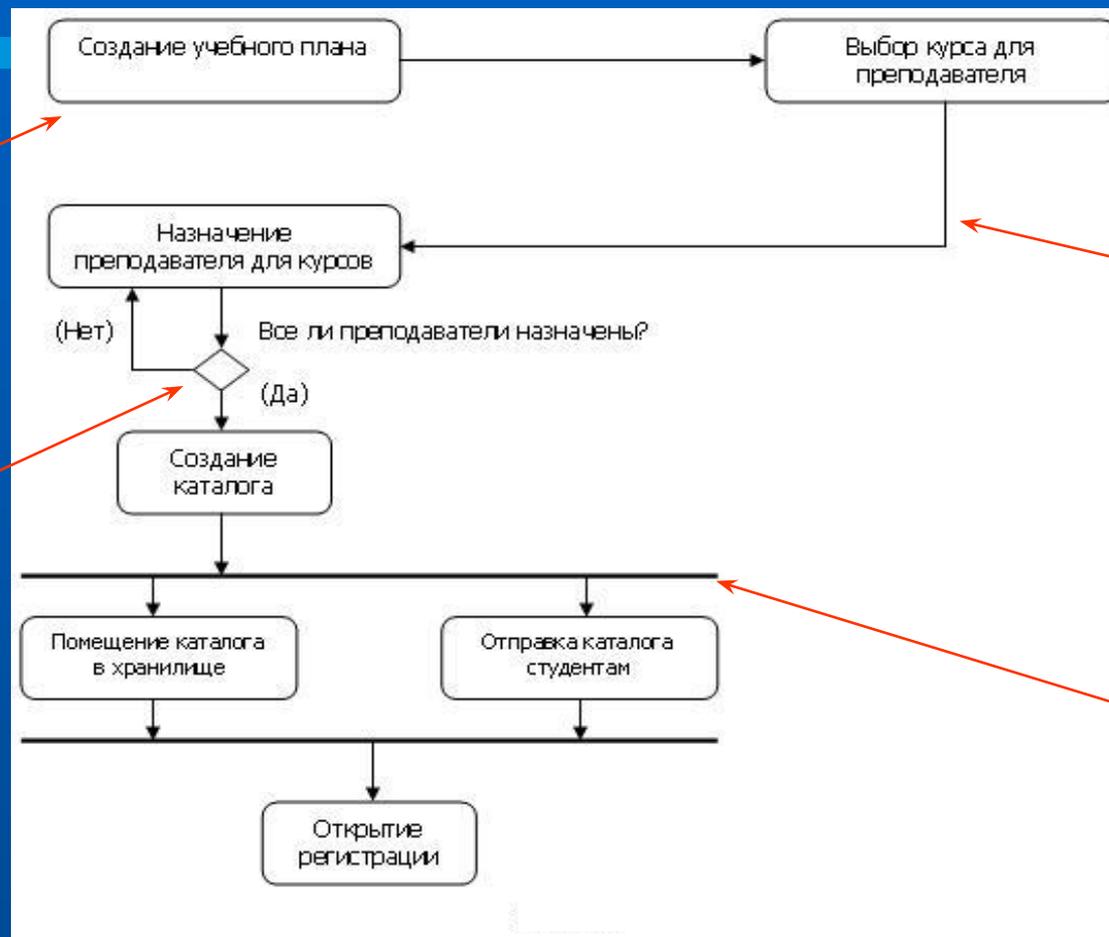
# Диаграммы действий

Далее для прецедента описываются подпотоки S-2, S-3, S-4. После этого описываются альтернативные потоки, которые определяют операции, выполняемые после ошибочных действий преподавателя, например после того как преподаватель ошибется и неправильно введет свой идентификационный номер.

Описания потоков событий составляются для всех прецедентов, показанных на рисунке.

После описания внешней границы АИС обычно составляются диаграммы действий. Одна из диаграмм действий для АИС Истерн:

# Диаграммы действий



Действие

Переход

Точка принятия решения

Полосы синхронизации

# Диаграммы действий

*Действие* (activity) описывает некоторый фрагмент поведения системы в контексте потока функций управления.

Элемент *перехода* (transition) применяется в диаграмме с целью обозначения направления передачи управления от одного действия к другому.

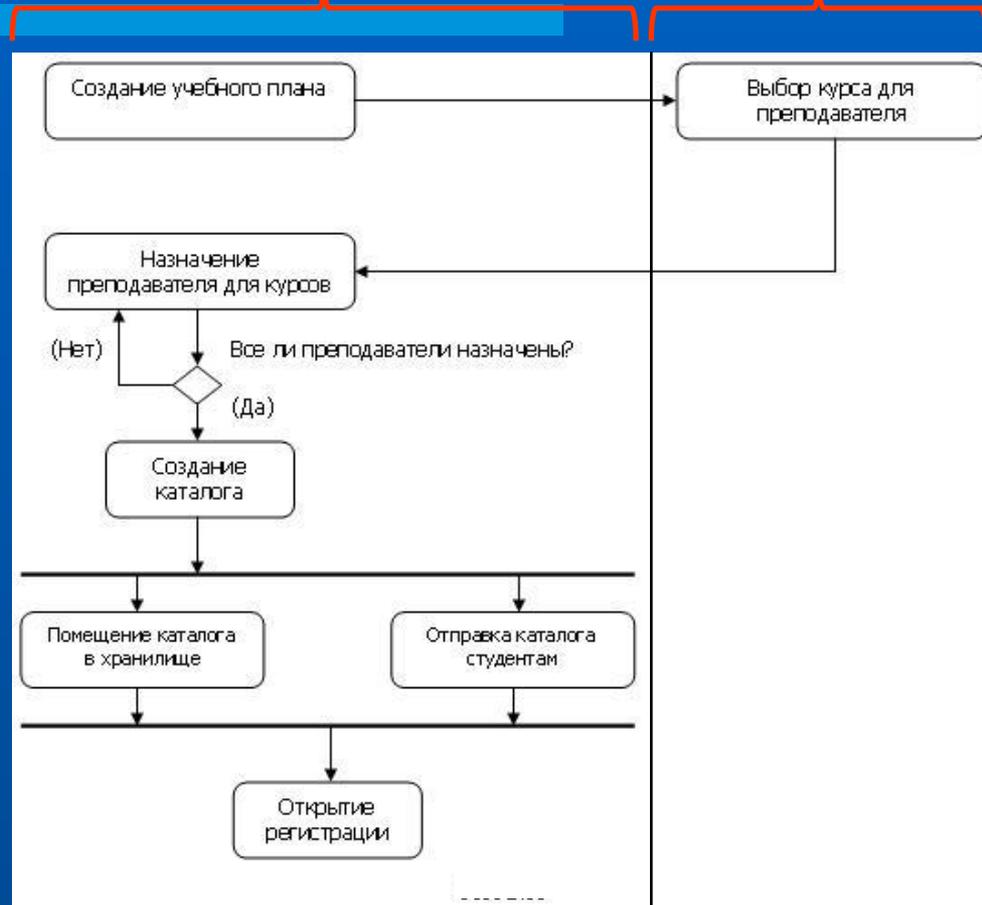
В процессе моделирования системы зачастую необходимо определить, в какие моменты и в каких точках поток претерпевает ветвление в зависимости от принимаемых системой или пользователем решений. Переход, который берет начало в *точке принятия решения* (decision point), содержит *контролируемое условие* (guard condition), определяющее направление ветвления.

# Диаграммы действий

Процесс функционирования системы зачастую содержит стадии, которые могут выполняться параллельно. *Полосы синхронизации* (synchronization bars) позволяют указать какие действия допускают единовременное выполнение или подлежат логическому объединению. Возможны ситуации, когда полоса синхронизации снабжается несколькими входящими связями и единственной исходящей и наоборот.

Диаграмма действий может быть разделена на *зоны* (swimlanes), каждая из которых обычно связана с определенным активным субъектом, ответственным за выполнение подмножества действий.

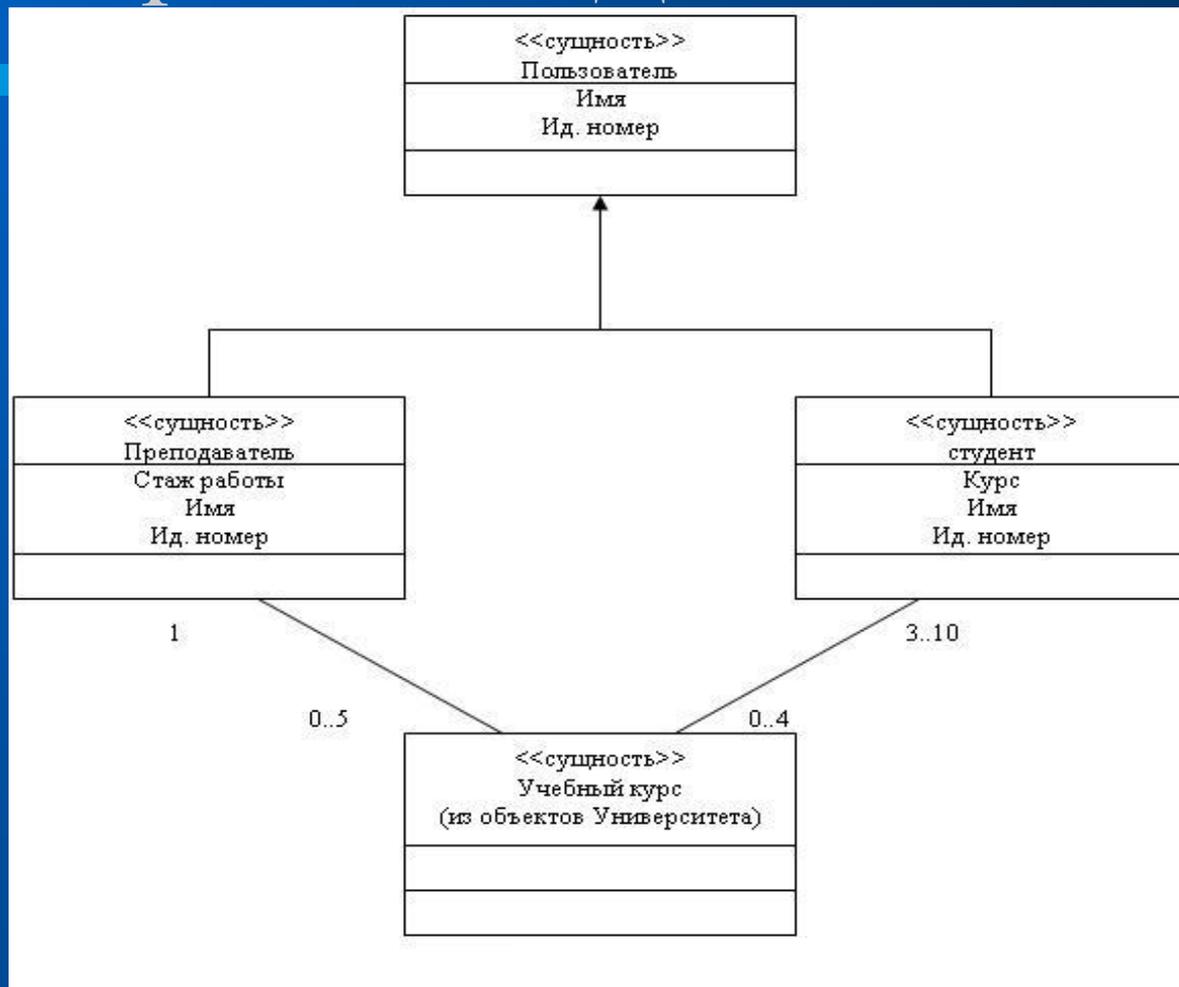
# Диаграммы действий



# Проектирование БД

Очевидно, что для компьютерной системы Истерн целесообразно использовать реляционную (табличную) базу данных. Проектирование базы данных обычно начинается с построения ее концептуальной схемы, на которой реляционные таблицы представлены в виде классов (сущностей), а логические соединения между таблицами изображены линиями, имитирующими двунаправленные или однонаправленные отношения между сущностями. На следующем рисунке показан один из фрагментов обширной концептуальной схемы базы данных компьютерной системы Истерн.

# Проектирование БД



# Проектирование БД

Этот фрагмент концептуальной схемы базы данных может быть построен в виде диаграммы классов с помощью программы Rational Rose, основанной на языке UML. Линии, заканчивающиеся стрелкой показывают иерархию наследования для класса "пользователь". Правая линия без стрелки с числовыми метками изображает двунаправленное отношение между классами Учебный курс и Студент. Метки 0..4 и 3..10 означают, что в данном семестре конкретный студент может посещать от 0 до 4-х учебных курсов, а данный курс может читаться несколькими студентами - от 3-х до 10-ти.

Если мы обратимся к модели архитектуры компьютерной системы, то диаграмму классов надо рассматривать как часть логической схемы компьютерной системы.

# Конструирование классов

Объект – представление реальной или абстрактной сущности.

Класс – группа объектов с общими свойствами.

Предложение Курса
Место Проведения Занятий
Время Проведения Занятий
Наименование Курса
Номер Предложения
Назначить Место Проведения Занятий
Назначить Время Проведения Занятий
Зарегистрировать студента
Отменить регистрацию студента

# Конструирование классов

*Стереотипы* классов позволяют создавать новые разновидности элементов моделируемой системы.

Наиболее употребительные стереотипы классов:

- Сущность (entity),
- Граница (boundary),
- Управление (control),
- Прикладной класс (utility),
- Исключение (exception).

На диаграммах названия стереотипов заключаются в угловые кавычки и располагаются над именами классов.

# Конструирование классов

Стереотип - это не то же самое, что родительский класс в отношении обобщения "родитель/потомок". Точнее было бы охарактеризовать его как некоторый метатип, поскольку каждый стереотип создает эквивалент нового класса в метамодели UML. Например, при моделировании бизнес-процесса вам потребуется ввести в модель элементы, представляющие работников, документы и стратегии.

# Конструирование классов

Приписывая стереотип таким элементам, как узел или класс, вы по сути дела расширяете UML, создавая новый строительный блок, который напоминает существующий, но обладает новыми специальными свойствами (каждый стереотип может содержать свой набор помеченных значений), новой семантикой (у каждого стереотипа могут быть собственные ограничения) и нотацией (каждому стереотипу может быть присвоена пиктограмма).



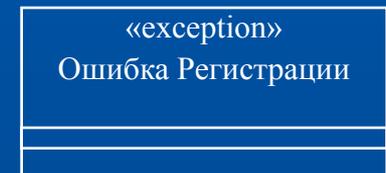
Форма Регистрации  
«boundary»



Предложение Курса  
«entity»



Контроль Регистрации  
«control»



# Конструирование классов

*Класс сущностей (entity class)* моделирует структуру данных и поведение, отличающиеся стабильным характером. Класс подобного типа отражает качества сущности реального мира или применяется для выполнения внутренних функций системы. Классы сущностей обычно не зависят от окружения, т.е. не чувствительны к тому, каким образом внешние обстоятельства влияют на работу системы. Во многих случаях такие классы удается с успехом применять в нескольких различных приложениях.

Первый шаг состоит в разграничении сфер ответственности системы на основе результатов анализа потока событий, охватывающего определенные варианты использования.

# Конструирование классов

*Классы границ* (boundary classes) обслуживают процессы взаимодействия между системой и ее окружением, обеспечивая интерфейсы для пользователей и сторонних систем (т.е. активных субъектов). Такие классы образуют часть системы, которая непосредственно "общается" с внешним миром.

Для отыскания классов границ анализируется каждая пара вида "активный субъект/вариант использования". Классы границ, выбранные на этапе планирования, обычно отличаются низким уровнем детализации. В этот момент, напр., целесообразно моделировать и документировать функции окна графического интерфейса, рассматриваемого в целом, не оговаривая особенности реализации его частей (кнопок, переключателей, опций и т.п.).

# Конструирование классов

Требования, предъявляемые к элементам графического интерфейса пользователя, зачастую весьма неоднозначны. Хотя термины "дружественный интерфейс" или гибкий интерфейс" употребляются повсеместно, о единодушии при их толковании говорить не приходится. Это как раз тот случай, когда весьма уместны подходы, связанные с коллективным обсуждением решений и созданием прототипов: потребитель получает возможность заранее "пощупать" будущую систему и прийти к осмысленному выводу о том, насколько в действительности "дружественны" предлагаемые интерфейсы.

# Конструирование классов

Затем найденные критерии "дружественности" закрепляются в виде структур и характеристик поведения классов границ. В процессе проектирования классы уточняются и "шлифуются" с учетом особенностей выбранных механизмов их реализации.

Как упоминалось выше, классы границ используются и для моделирования способов взаимодействия разрабатываемой системы с другими системами.

# Конструирование классов

*Классы управления (control classes) находят применение при реализации характеристик поведения системы, присущих одному или нескольким вариантам использования, и координируют события, возникающие по мере функционирования системы в рамках этих вариантов. Класс управления можно воспринимать как абстракцию, отображающую динамику варианта использования системы. Классы управления обычно тесно "привязаны" к особенностям конкретного приложения.*

На ранних стадиях жизненного цикла системы для каждой пары вида "активный субъект/вариант использования" создается по одному классу управления, на который возлагаются обязанности по контролю за потоком событий, происходящих по мере выполнения этого варианта.

# Конструирование классов

Классы границ: Внесение предложения курса,  
Опции выбора курсов

Классы сущностей: Курс, Предложение Курса, Профессор

Классы управления: Контроль предложения курса

# Вопросы?

---

07/19/2022

МФ ПГУ им.М.В.  
Ломоносова

35