

Оглавление

Лабораторная работа № 4 ПОСТРОЕНИЕ МОДЕЛИ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ. НОТАЦИЯ IDEF0.....	1
Лабораторная работа №5 ПОСТРОЕНИЕ МОДЕЛИ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ.ОПТИМИЗАЦИЯ БИЗНЕС-ПРОЦЕССОВ.....	2
Лабораторная работа №6 ПОСТРОЕНИЕ МОДЕЛИ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ В НОТАЦИИ IDEF3.....	3
Лабораторная работа №7 «СОЗДАНИЕ DFD-МОДЕЛИ БИЗНЕС ПРОЦЕССА».....	6
Лабораторная работа №8 «ПОСТРОЕНИЕ UML-МОДЕЛИ БИЗНЕС-ПРОЦЕССА».....	8
Варианты предметной области:.....	14

Лабораторная работа № 1 ПОСТРОЕНИЕ МОДЕЛИ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ. НОТАЦИЯ IDEF0

Цель работы:

- 1 Знакомство с графической нотацией формализации и описания бизнес-процессов IDEF0. Знакомство с понятием функциональной модели AS-IS («как есть»).
- 2 Описание и построение функциональной модели AS-IS выбранной предметной области с применением нотации IDEF0.

Постановка задачи:

Для заданной предметной области разработать модель AS-IS. Вы можете выбрать один из вариантов процессов, описанных в приложении, или предложить свой вариант.

Модель должна содержать:

- не менее трёх уровней;
- на диаграмме 1-го уровня (A0) должно быть не менее 4-х функциональных блоков;
- на диаграмме 2-го и далее уровнях должно быть не менее 2-х функциональных блоков.

Указания:

Модель AS-IS – это модель «как есть», т. е. модель уже существующего процесса/функции. Анализ процессов является обязательной частью любого проекта создания или развития системы. Построение функциональной модели AS-IS позволяет четко зафиксировать, какие процессы осуществляются на предприятии, какие информационные объекты используются при выполнении функций различного уровня детализации.

Модель AS-IS показывает зоны ответственности исполнителей процесса и ход самого процесса («кто что сделал», как взаимосвязаны этапы между собой и как каждый этап влияет на конечный результат). Функциональная модель AS-IS является отправной точкой для анализа потребностей предприятия, выявления проблем и «узких» мест, разработки проекта совершенствования деловых процессов. Анализ функциональной модели AS-IS позволяет понять, в чем заключается проблема, в чем будут состоять преимущества новых процессов и каким изменениям подвергнется существующая структура организации процесса в результате оптимизации.

Лабораторная работа №2 ПОСТРОЕНИЕ МОДЕЛИ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ.ОПТИМИЗАЦИЯ БИЗНЕС- ПРОЦЕССОВ

Цель работы:

1. Знакомство с понятием функциональной модели TO-BE («как будет»).
2. Доработка созданной модели AS-IS с учетом выявленных недостатков в организации бизнес-процессов.

Постановка задачи:

Для заданной предметной области преобразовать созданную модель AS-IS в модель TO-BE. Внедрив информационную систему или клиент-серверную архитектуру.

Модель должна содержать:

- не менее трёх уровней;
- на диаграмме 1-го уровня (A0) должно быть не менее 4-х функциональных блоков;
- на диаграмме 2-го и далее уровнях должно быть не менее 2-х функциональных блоков.

Указания:

Найденные в модели AS-IS недостатки исправляются путем создания модели TO-BE («как будет»), т. е. модели новой организации процессов на предприятии. Создание и внедрение информационной системы (ИС) приводит к изменению условий выполнения отдельных операций, структуры процессов и предприятия в целом.

Функциональная модель TO-BE позволяет уже на стадии проектирования будущей ИС определить эти изменения. Применение функциональной модели TO-BE позволяет не

только сократить сроки внедрения информационной системы, но также снизить риски, связанные с невосприимчивостью персонала к информационным технологиям. Модель TO-BE нужна для анализа альтернативных (лучших) путей выполнения функции и документирования того, как компания будет делать бизнес в будущем.

Лабораторная работа №3 ПОСТРОЕНИЕ МОДЕЛИ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ В НОТАЦИИ IDEF3.

Цель работы:

Получить практические навыки в построении IDEF3-модели бизнес-процесса.

Постановка задачи:

С помощью методологии IDEF3 декомпозировать 1 из функциональных блоков модели окружения (A0), используя все типы перекрестков.

Модель должна содержать:

Перекрестки типа AND, OR, XOR

Указания:

Для описания логики взаимодействия информационных потоков наиболее подходит IDEF3, называемая также workflow diagramming - методологией моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов. Диаграммы Workflow могут быть использованы в моделировании бизнес-процессов для анализа завершенности процедур обработки информации. С их помощью можно описывать сценарии действий сотрудников организации, например последовательность обработки заказа или события, которые необходимо обработать за конечное время. Каждый сценарий сопровождается описанием процесса и может быть использован для документирования каждой функции.

IDEF3 - это метод, имеющий основной целью дать возможность аналитикам описать ситуацию, когда процессы выполняются в определенной последовательности, а также описать объекты, участвующие совместно в одном процессе,

Техника описания набора данных IDEF3 является частью структурного анализа. В отличие от некоторых методик описаний процессов IDEF3 не ограничивает аналитика чрезмерно жесткими рамками синтаксиса, что может привести к созданию неполных или противоречивых моделей.

IDEF3 может быть также использован как метод создания процессов. IDEF3 дополняет IDEF0 и содержит все необходимое для построения моделей, которые в дальнейшем могут быть использованы для имитационного анализа.

Каждая работа в IDEF3 описывает какой-либо сценарий бизнес-процесса и может являться составляющей другой работы. Поскольку сценарий описывает цель и рамки модели, важно, чтобы работы именовались отглагольным существительным, обозначающим процесс действия, или фразой, содержащей такое существительное.

Точка зрения на модель должна быть задокументирована. Обычно это точка зрения человека, ответственного за работу в целом. Также необходимо задокументировать цель модели - те вопросы, на которые призвана ответить модель.

Диаграммы.

Диаграмма является основной единицей описания в IDEF3.

Единицы работы - Unit of Work (UOW). UOW, также называемые работами (activity), являются центральными компонентами модели. В IDEF3 работы изображаются прямоугольниками с прямыми углами и имеют имя, выраженное отглагольным существительным, обозначающим процесс действия, одиночным или в составе фразы, и номер (идентификатор); другое имя существительное в составе той же фразы обычно отображает основной выход (результат) работы, например, "Изготовление изделия".

Связи.


Связи показывают взаимоотношения работ. Все связи в IDEF3 однонаправлены и могут быть направлены куда угодно, но обычно диаграммы IDEF3 стараются построить так, чтобы связи были направлены слева направо. В IDEF3 различают три типа стрелок, изображающих связи:

- Старшая (Precedence) - сплошная линия, связывающая единицы работ (UOW), Рисуется слева направо или сверху вниз. Показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется.
- Отношения (Relational Link) - пунктирная линия, используемая для изображения связей между единицами работ (UOW) а также между единицами работ и объектами ссылок.
- Потоки объектов (Object Flow) - стрелка с двумя наконечниками, применяется для описания того факта, что объект используется в двух или более единицах работы, например, когда объект порождается в одной работе и используется в другой.

Старшая связь и поток объектов. Старшая связь показывает, что работа-источник заканчивается ранее, чем начинается работа-цель. Часто результатом работы-источника становится объект, необходимый для запуска работы-цели. В этом случае стрелку, обозначающую объект, изображают с двойным наконечником. Имя стрелки должно ясно идентифицировать отображаемый объект. Поток объектов имеет ту же семантику, что и старшая стрелка.

Перекрестки (Junction). Окончание одной работы может служить сигналом к началу нескольких работ, или же одна работа для своего запуска может ожидать окончания нескольких работ. Перекрестки используются для отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы. Различают перекрестки для слияния (Fan-in Junction) и разветвления (Fan-out Junction) стрелок. Перекресток не может использоваться одновременно для слияния и для разветвления. Для внесения перекрестка служит кнопка в палитре инструментов - добавить в диаграмму перекресток Junction. В диалоге Junction Type Editor необходимо указать тип перекрестка. Смысл каждого типа приведен в таблице 1.

Таблица 1 - Типы перекрестков

Обозначение	Наименование	Смысл в случае слияния стрелок	Смысл в случае разветвления стрелок
	Asynchronous AND	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Synchronous AND	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
	Asynchronous OR	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
	Synchronous OR	Один или несколько предшествующих процессов завершены одновременно	Один или несколько следующих процессов запускаются одновременно
	XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

В отличие от IDEF0 в IDEF3 стрелки могут сливаться и разветвляться только через перекрестки.

Декомпозиция работ. В IDEF3 декомпозиция используется для детализации работ. Методология IDEF3 позволяет декомпозировать работу многократно, т.е. работа может иметь множество дочерних работ. Это позволяет в одной модели описать альтернативные потоки.

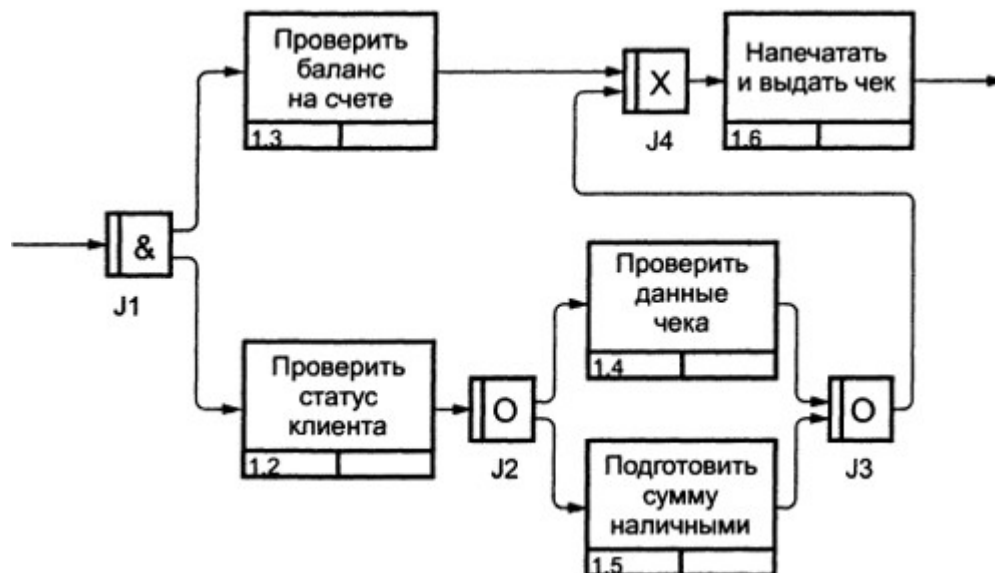


Рис. 0. Пример диаграммы в нотации IDEF3.

Лабораторная работа №4 «СОЗДАНИЕ DFD-МОДЕЛИ БИЗНЕС ПРОЦЕССА»

Цель работы: получить практические навыки в построении DFD-модели бизнес-процесса.

Постановка задачи:

С помощью методологии DFD декомпозировать 1 из функциональных блоков. Можно выбрать часть процесса, который моделировался на предыдущих лабораторных работах. При выборе учтите, что процесс обязательно должен предусматривать обработку информации, лучше, чтобы это была автоматизированная обработка с использованием одной или нескольких информационных систем.

Знакомство с основами методологии DFD.

Диаграммы потоков данных (Data flow diagramming, DFD) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации. DFD описывает: ⌚ процессы обработки информации (работы); ⌚ потоки данных (стрелки, arrow), которые могут моделировать и потоки материальных объектов (изделия, документы); ⌚ внешние ссылки (external references), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы; ⌚ хранилища данных (data store).

Как и в IDEF0, основными элементами DFD-диаграмм являются функциональные блоки, которые называются процессами или работами. Они преобразуют входы в выходы (чаще всего это преобразование входных данных в выходные). Блоки соединяются стрелками. В отличие от стрелок IDEF0, которые представляют собой жесткие ограничения на работу блоков, стрелки DFD показывают, как объекты (как правило, данные) двигаются от одной работы к другой. Это представление потоков совместно с хранилищами данных и внешними сущностями делает модели DFD более похожими на описание физических характеристик системы – движения объектов (data flow), хранения объектов (data stores), поставки и распространения объектов (external entities). При построении диаграмм потоков данных наиболее часто используют две нотации: Йордана и Гейна-Сарсона. Обе нотации имеют одинаковый по названиям и значению элементный состав, но имеют различное его графическое изображение.

Элементы DFD диаграммы

Выделяют 4 элемента в диаграмме:

1. Процесс.

Процессы, при которых идет изменение потока данных (обработка, трансформация и др. изменения). Процесс, как и в других диаграммах обычно прописывается с помощью глагола, например: “Отправка заполненной формы”.

2. Внешняя сущность.

Сущность (объект), которая получает или отправляет данные при взаимодействии с описанным процессом.

3. Хранилище данных.

Все хранилища данных или отдельные файлы, которые хранят исходные или выходные данные, а также все промежуточные хранилища.

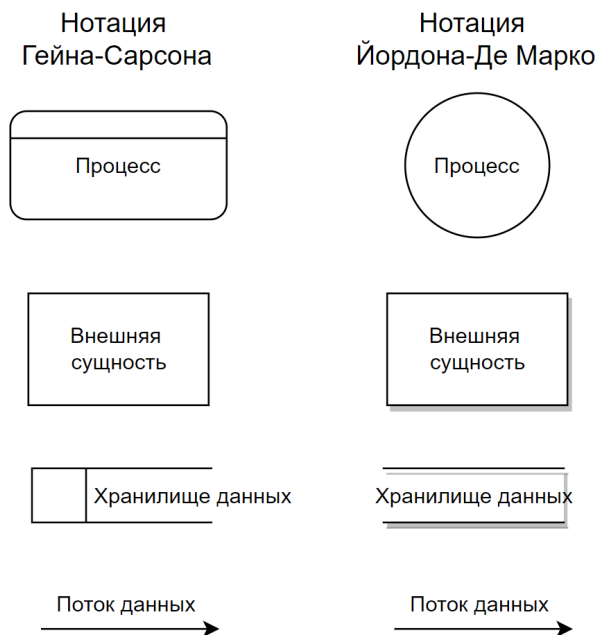
4. Поток данных.

Поток данных, который отображает направление и сами данные, которые перемещаются между внешними сущностями и хранилищами данных с помощью процессов.

Несколько правил построения диаграмм:

- Процесс должен иметь входной и выходной поток данных.
- Хранилища данных также должны иметь входные и выходные потоки данных.
- Данные с внешних сущностей должны обязательно проходить через процесс чтобы попасть в хранилище.

В DFD диаграммах также выделяют 2 разные нотации. Поэтому стоит обращать внимание на условные обозначения каждого элемента в зависимости от используемой нотации. Ниже представлена картинка сравнения элементов разных нотаций.



Лабораторная работа №5 «ПОСТРОЕНИЕ UML-МОДЕЛИ БИЗНЕС-ПРОЦЕССА»

Цель работы: получить практические навыки в построении прецедентной UML-модели бизнес-процесса.

Порядок выполнения работы.

Выберите бизнес-процесс, для которого будете формировать модель. Вы можете выбрать один из вариантов процессов, описанных в приложении, или предложить свой вариант. Можно выбрать один из процессов, для которого на предыдущих лабораторных работах строилась модель по одной из структурных методологий. Желательно, чтобы процесс имел различные версии, т.е. альтернативные потоки событий.

Для заданной предметной области:

- построить диаграмму классов;
- построить диаграмму последовательности;
- построить диаграмму взаимодействий (диаграмму коммуникаций);
- построить диаграмму пакетов;

Знакомство с основами языка моделирования UML.

Унифицированный язык моделирования UML (Unified Modeling Language) предназначен для описания, визуализации и документирования бизнес-систем на базе

объектно-ориентированного подхода с целью последующего использования моделей бизнес-процессов для реализации их в виде программного обеспечения. Бурное развитие объектно-ориентированных языков программирования, сопровождающееся возрастанием сложности прикладных программ, вызвало потребность в создании объектно-ориентированного языка для формирования предварительной модели предметной области, для которой разрабатывается программа.

Такая модель необходима заказчикам, программистам и менеджерам проекта по созданию информационной системы для того, чтобы они могли выработать общий взгляд на цели и функции системы. И хотя модели предметной области, формируемые с помощью UML, предназначены, прежде всего, для последующей реализации в виде программного обеспечения, они имеют и самостоятельную ценность, т.к. позволяют наглядно отобразить функции и процессы бизнес-системы, объекты, участвующие в бизнеспроцессах, их отношения, а также динамику выполнения процессов. Начало работ над созданием унифицированного объектно-ориентированного языка моделирования относится к середине 1990-х годов. К тому времени уже было разработано более 50 различных языков объектно-ориентированного моделирования. Авторы наиболее распространенных языков – Г. Буч, Д. Румбах и А. Джекобсон, – собравшись «под крылом» компании Rational Software Corporation, начали работу над унифицированным методом. Ими был создан ряд версий унифицированного метода, который они называли Unified Modeling Language (UML). В настоящее время большинством производителей информационных систем и такими комитетами по стандартам, как ANSI и OMG, язык UML был признан в качестве стандарта.

В технологии реинжиниринга бизнес-процессов, пожалуй, впервые UML стали применять не только и не столько для создания информационных систем (ИС), сколько для анализа и перепроектирования бизнеса. Вместо моделей процессов, реализуемых информационной системой, строятся модели бизнес-процессов, даже если они и не будут подвергнуты автоматизации, вместо объектов ИС (программных объектов) в моделях отражаются объекты бизнеса (исполнители, продукция, услуги и т.д.), вместо окружения ИС (пользователей ИС) моделируется окружение бизнеса (поставщики, партнеры, клиенты).

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций (схем, графов), получивших название диаграмм. Предполагается, что никакая единственная модель не может с достаточной степенью адекватности описывать различные аспекты сложной системы. Таким образом,

модель сложной системы состоит из некоторого числа диаграмм, каждая из которых отражает некоторый аспект поведения или структуры системы.

Диаграммы UML подразделяют на два типа — это структурные диаграммы и диаграммы поведения.

Структурные диаграммы показывают статическую структуру системы и ее частей на разных уровнях абстракции и реализации, а также их взаимосвязь. Элементы в структурной диаграмме представляют значимые понятия системы и могут включать в себя абстрактные, реальные концепции и концепции реализации. Существует семь типов структурных диаграмм:

- Диаграмма составной структуры
- Диаграмма развертывания
- Диаграмма пакетов
- Диаграмма профилей
- Диаграмма классов
- Диаграмма объектов
- Диаграмма компонентов

Диаграммы поведения показывают динамическое поведение объектов в системе, которое можно описать, как серию изменений в системе с течением времени. А к диаграммам поведения относятся:

- Диаграмма деятельности
- Диаграмма прецедентов
- Диаграмма состояний
- Диаграмма последовательности
- Диаграмма коммуникаций
- Диаграмма обзора взаимодействия
- Временная диаграмма

Диаграмма вариантов использования представляет собой наиболее общую концептуальную модель системы, которая является исходной для построения всех остальных диаграмм. Представление вариантов использования детализируется с помощью диаграмм состояний, деятельности, последовательности и кооперации. Диаграммы классов используются для представления логической структуры информационной

системы, диаграммы компонентов и диаграммы развертывания — для представления физических компонентов информационной системы.

Диаграмма классов

Диаграмма классов — это центральная методика моделирования, которая используется практически во всех объектно-ориентированных методах. Эта диаграмма описывает типы объектов в системе и различные виды статических отношений, которые существуют между ними.

Три наиболее важных типа отношений в диаграммах классов (на самом деле их больше), это:

- ✓ Ассоциация, которая представляет отношения между экземплярами типов, к примеру, человек работает на компанию, у компании есть несколько офисов.
- ✓ Наследование, которое имеет непосредственное соответствие наследованию в Объектно-Ориентированном дизайне.
- ✓ Агрегация, которая представляет из себя форму композиции объектов в объектно-ориентированном дизайне.

Диаграмма Классов

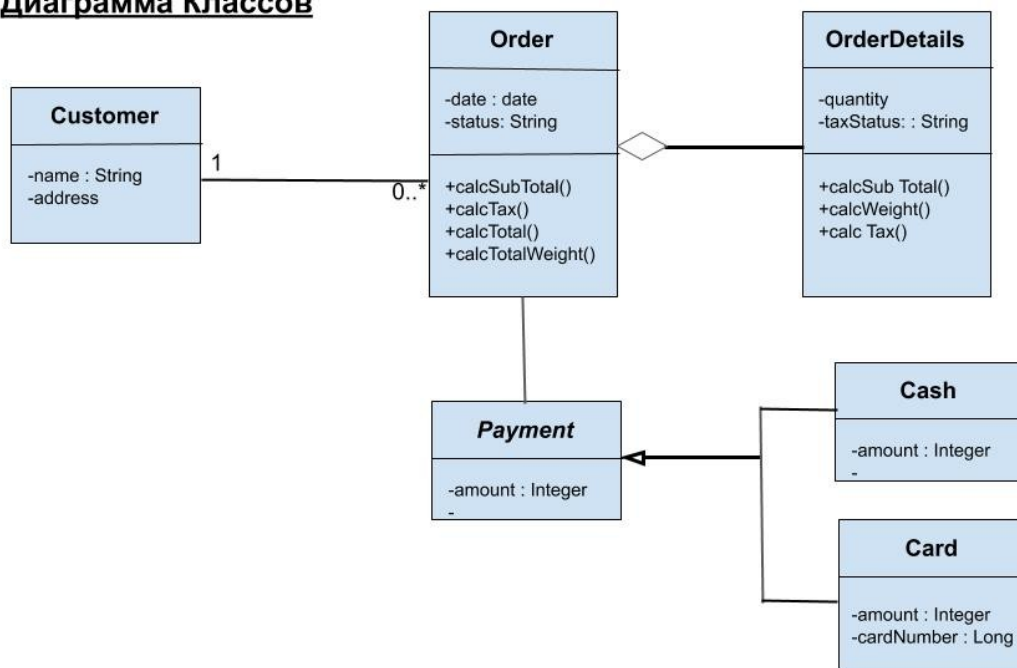


Диаграмма последовательности

Диаграмма последовательности моделирует взаимодействие объектов на основе временной последовательности. Она показывает, как одни объекты взаимодействуют с другими в конкретном прецеденте.

Диаграмма Последовательности

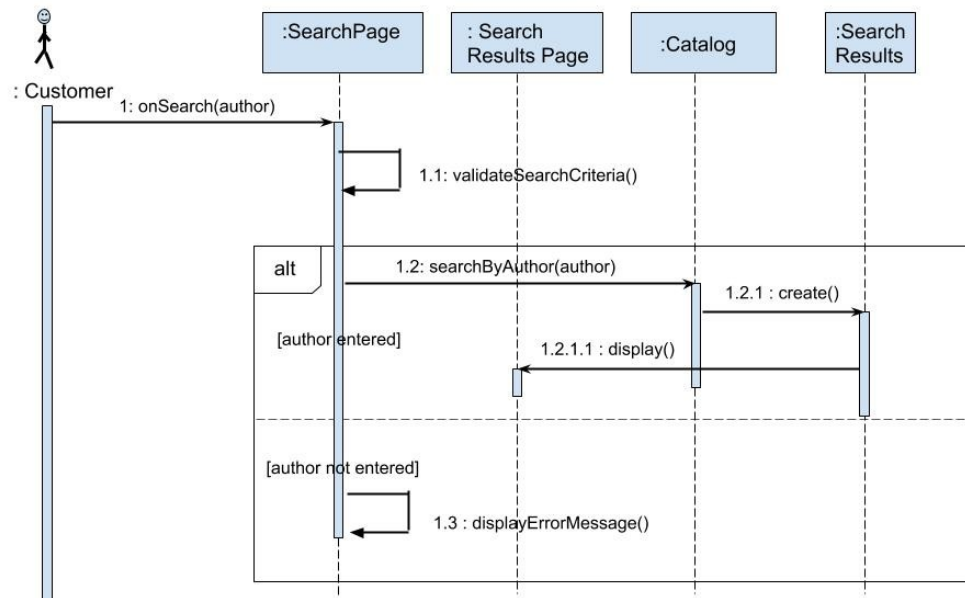


Диаграмма обзора взаимодействия

Диаграмма обзора взаимодействий фокусируется на обзоре потока управления взаимодействиями. Это вариант Диаграммы деятельности, где узлами являются взаимодействия или события взаимодействия. Диаграмма обзора взаимодействий описывает взаимодействия, в которых сообщения и линии жизни скрыты. Мы можем связать «реальные» диаграммы и добиться высокой степени навигации между диаграммами внутри диаграммы обзора взаимодействия.

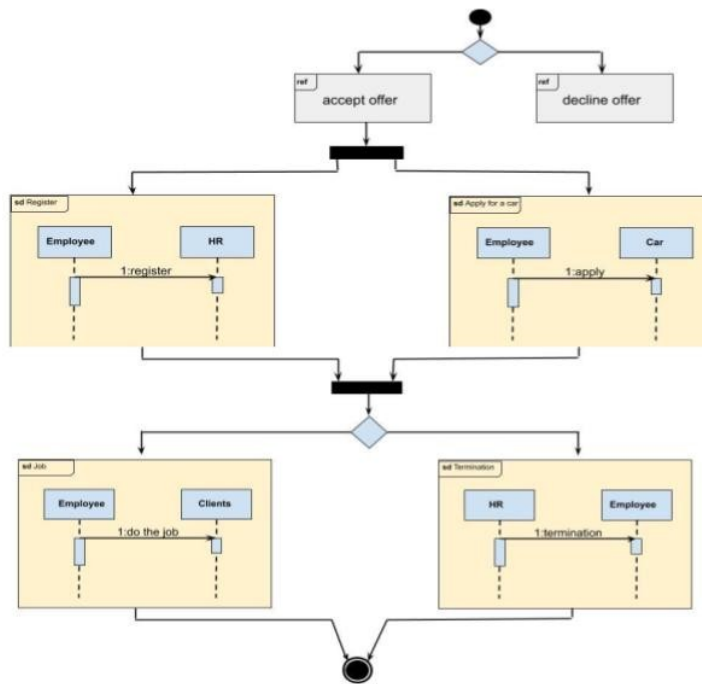
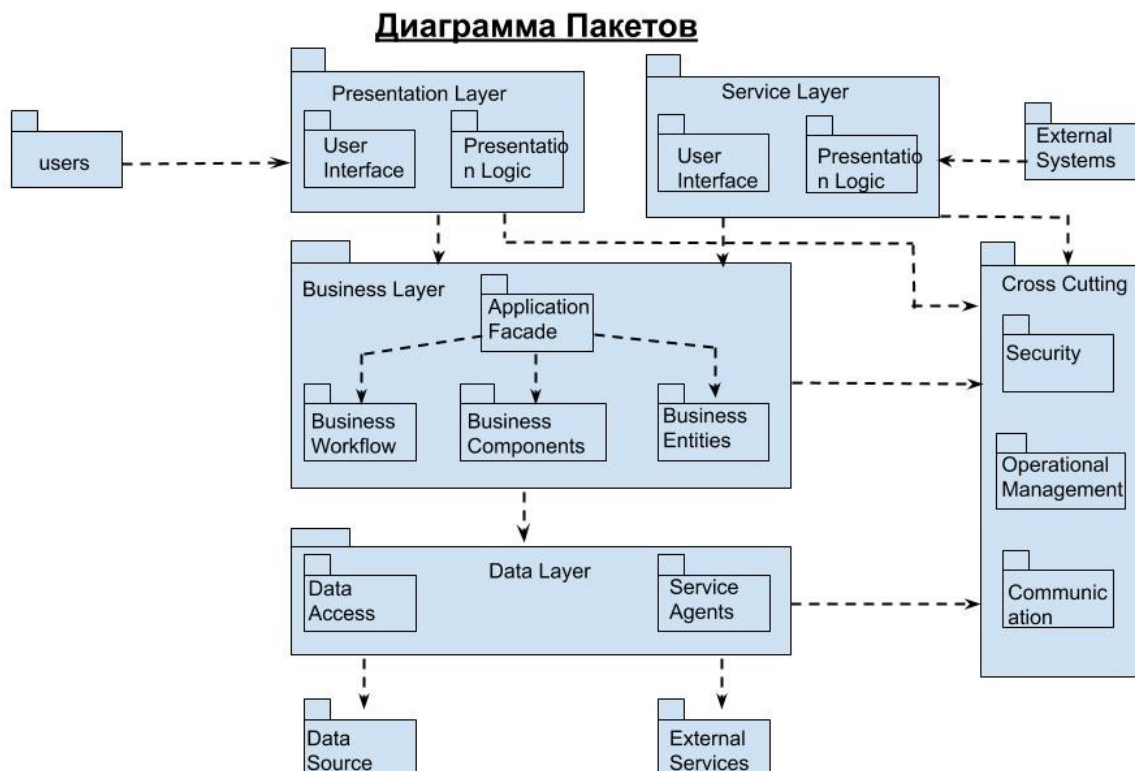


Диаграмма пакетов

Диаграмма пакетов — это структурная схема UML, которая показывает пакеты и зависимости между ними.

Она позволяет отображать различные виды системы, например, легко смоделировать многоуровневое приложение.



Варианты предметной области:

1. Моделирование бизнес-процессов розничного магазина;
2. Моделирование бизнес-процессов гостиницы;
3. Моделирование бизнес-процессов салона красоты;
4. Моделирование бизнес-процессов стоматологической клиники;
5. Моделирование бизнес-процессов ресторана;
6. Моделирование бизнес-процессов кафедры в университете;