

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**СТАВРОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ
УНИВЕРСИТЕТ**

Кафедра информационных систем

УТВЕРЖДАЮ

Заведующий кафедрой

«__» _____ 20__ г.

**Методическая разработка и указания к практическим занятиям
по дисциплине «Комплексные системы управления в структуре бизнеса»
для студентов направления подготовки 38.03.05– «Бизнес-информатика»**

**Практическое занятие №3 «Реализация объектно-ориентированного
подхода к проектированию комплексных систем управления с помощью
CASE-средства ARIS-EXPRESS»**

Рассмотрено УМК

«__» _____ 20__ г.

Протокол № _____

Председатель УМК

Ставрополь, 2021

Рецензент:

доктор технических наук, профессор Федоренко В.В.

Одобрено учебно-методической комиссией экономического факультета
Ставропольского государственного аграрного университета

Методические указания к практическим занятиям разработаны в соответствии с программой курса «Комплексные системы управления в структуре бизнеса» и предназначены для студентов направления подготовки 38.03.05 – «Бизнес-информатика»

Составитель:

к.т.н., доцент Рачков В.Е.

ОГЛАВЛЕНИЕ

1.	Меры безопасности при работе на компьютере	4
2.	Введение	5
3.	Реализация объектно-ориентированного подхода к проектированию комплексных систем управления	6
4.	Практическое занятие №3	13
5.	Список литературы	14

1. Меры безопасности при работе на компьютере

Конструкция компьютера обеспечивает электробезопасность для работающего на нем человека. Тем не менее, компьютер является электрическим устройством, работающим от сети переменного тока напряжением 220 В., а в мониторе напряжение, подаваемое на кинескоп, достигает нескольких десятков киловольт. Чтобы предотвратить возможность поражения электрическим током, возникновения пожара и выхода из строя самого компьютера при работе и техническом обслуживании компьютера необходимо соблюдать следующие меры предосторожности:

- сетевые розетки, от которых питается компьютер, должны соответствовать вилкам кабелей электропитания компьютера;
- запрещается использовать в качестве заземления водопроводные и газовые трубы, радиаторы и другие узлы парового отопления;
- запрещается во время работы компьютера отключать и подключать разъемы соединительных кабелей;
- запрещается снимать крышку системного блока и производить любые операции внутри корпуса до полного отключения системного блока от электропитания;
- запрещается разбирать монитор и пытаться самостоятельно устранять неисправности (опасные для жизни высокие напряжения на элементах схемы монитора сохраняются длительное время после отключения электропитания);
- запрещается закрывать вентиляционные отверстия на корпусе системного блока и монитора посторонними предметами во избежание перегрева элементов, расположенных внутри этих устройств;
- повторное включение компьютера рекомендуется производить не ранее, чем через 20 секунд после выключения.

2 Введение

Практическое занятие предполагает отработку следующих вопросов:

1. Изучение основных этапов объектно-ориентированного проектирования комплексных систем управления организацией.
2. Формирование навыков в проектировании комплексных систем управления с применением объектно-ориентированного подхода.

3. Реализация объектно-ориентированного подхода к проектированию комплексных систем управления

3.1 Общая характеристика объектно-ориентированного подхода к проектированию

Объектно-ориентированный подход к проектированию ИС основан на **объектной декомпозиции системы**. Модель системы в данном случае должна отображать ее статическую структуру, которая описывается в терминах объектов и связей между ними, и поведение, определяемое в терминах обмена сообщениями между объектами.

Данный подход к проектированию ИС подразумевает выполнение трех этапов:

1. **Объектно-ориентированный анализ** — методология, направленная на создание моделей реальной действительности на основе объектно - ориентированного мировоззрения, когда объекты отражают реальные объекты- сущности и операции, выполняемые этими объектами.

2. **Объектно-ориентированное проектирование** — методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы. Особое внимание уделяется проектированию системной архитектуры.

3. **Объектно-ориентированное программирование** - методология программирования, предполагающая реализацию модели системы на объектно- ориентированном языке программирования, непосредственно выполняющего отражение определенных объектов и предоставляющего средства для определения классов объектов.

Концептуальной основой объектно-ориентированного подхода является объектная модель, в соответствии с которой система представляется в виде совокупности классов и объектов предметной области. Объект представляет собой неразрывную совокупность данных, определяющих его состояние, и набора доступных за пределом объекта процедур доступа к его данным, составляющих методы объекта. В качестве объекта выступают реальные предметы окружающего мира, а также абстрактные или реальные сущности (такие как контрагент, заказ и др.).

Каждый объект относится к некоторому объектному типу (классу), в котором определена внутренняя структура данных любого объекта данного класса, а также программный код методов объекта.

3.2 Основные принципы построения объектной модели

Основные принципы построения объектной модели:

– **принцип абстрагирования** предполагает выделение наиболее значимых характеристик рассматриваемого объекта, отличающих его от всех других видов объектов, и игнорирование менее важных, определяя

тем самым его концептуальные границы с точки зрения наблюдателя. Основными абстракциями предметной области являются объекты и классы;

– **принцип инкапсуляции** декларирует физическое отделение друг от друга элементов объекта, определяющих его устройство и поведение. Инкапсуляция призвана изолировать интерфейс объекта, отражающий его внешнее поведение, от внутренней реализации объекта. Абстрагирование и инкапсуляция являются взаимодополняющими: абстрагирование фокусирует внимание на внешних особенностях объекта, а инкапсуляция (или иначе ограничение доступа) не позволяет объектам-пользователям различать внутреннее устройство объекта;

– **принцип модульности** предполагает такую структуру систем, согласно которой они могут быть разложены на ряд внутренне сильно связанных подсистем, которые между собой связаны слабо. Принцип модульности дополняет принципы абстрагирования и инкапсуляции: объект логически определяет границы определенной абстракции, а инкапсуляция и модульность делают его относительно независимым и самостоятельным;

– **принцип иерархии** декларирует некоторое ранжированное упорядочение абстракций с расположением их по уровням. Основными видами иерархических структур применительно к сложным системам являются структура классов и структура объектов.

Объектный подход предполагает рассмотрение системы с различных точек зрения, которые являются взаимосвязанными. **Модель системы с определенной точки зрения называется ее представлением.** Каждое представление описывает важные аспекты системы только со своей точки зрения, таким образом, для получения полного описания системы необходимо использовать все ее представления. В объектно-ориентированном подходе выделяются следующие типы моделей (представления):

1. **Модель классов** раскрывает статическую структуру объектов системы: их индивидуальность, отношения с другими объектами, атрибуты и операции. Эта модель создает контекст для остальных представлений. Модель классов представляется в виде диаграммы классов.

2. **Модель состояний** описывает процесс изменения состояний объектов с течением времени. Данное представление охватывает вопросы управления - аспект системы, описывающий порядок осуществления операций без учета их фактического значения, участников и реализации. Модель состояний изображается в виде диаграммы состояний, действия и события которой становятся операциями объектов из модели классов.

3. **Модель взаимодействия** показывает, как объекты взаимодействуют между собой, то есть раскрывает кооперацию объектов для обеспечения необходимого поведения системы как единого целого. Данное представление изображается в виде вариантов использования на диаграммах последовательности и деятельности. Варианты использования показывают взаимодействие системы с внешними объектами.

Диаграммы последовательности направлены на отображение временной последовательности взаимодействия объектов. Диаграммы

деятельности показывают поток управления между последовательными этапами вычислений.

3.3 Реализация объектно-ориентированного подхода с помощью CASE-средства

Объектно-ориентированный анализ и проектирование (ООАП, Object-Oriented Analysis/Design) - технология разработки программных систем, в основу которых положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов.

Методология ООАП тесно связана с концепцией автоматизированной разработки программного обеспечения (Computer Aided Software Engineering, CASE). К первым CASE-средствам отнеслись с определенной настороженностью. Со временем появились как восторженные отзывы об их применении, так и критические оценки их возможностей. Причин для столь противоречивых мнений было несколько. **Первая** из них заключается в том, что ранние CASE-средства были простой надстройкой над системой управления базами данных (СУБД). Визуализация процесса разработки концептуальной схемы БД имеет немаловажное значение, тем не менее, она не решает проблем создания приложений других типов.

Вторая причина связана с графической нотацией, реализованной в CASE-средстве. Если языки программирования имеют строгий синтаксис, то попытки предложить подходящий синтаксис для визуального представления концептуальных схем БД, были восприняты далеко не однозначно. На этом фоне разработка и стандартизация унифицированного языка моделирования UML вызвала воодушевление у всего сообщества корпоративных программистов.

В рамках ООАП исторически рассматривались три графических нотации:

- диаграммы **«сущность-связь»** (Entity-Relationship Diagrams, ERD);
- диаграммы **функционального моделирования** (Structured Analysis and Design Technique, SADT);
- диаграммы **потоков данных** (Data Flow Diagrams, DFD).

Диаграммы «сущность-связь» (ERD) предназначены для графического представления моделей данных разрабатываемой программной системы и предлагают набор стандартных обозначений для определения данных и отношений между ними. С помощью этого вида диаграмм можно описать отдельные компоненты концептуальной модели данных и совокупность взаимосвязей между ними.

Основными понятиями данной нотации являются понятия сущности и связи. При этом под сущностью (entity) понимается произвольное множество реальных или абстрактных объектов, каждый из которых обладает одинаковыми свойствами и характеристиками. В этом случае любой

рассматриваемый объект может быть экземпляром одной и только одной сущности, должен иметь уникальное имя или идентификатор, а также отличаться от других экземпляров данной сущности.

Связь (relationship) определяется как отношение или ассоциация между отдельными сущностями. Примерами связей могут являться родственные отношения, в частности "отец-сын" или производственные - "начальник-подчиненный". Другой тип связей задается отношениями "иметь в собственности" или "обладать свойством". Различные типы связей графически изображаются в форме ромба с соответствующим именем данной связи.

Графическая модель данных строится таким образом, чтобы связи между отдельными сущностями отражали не только семантический характер соответствующего отношения, но и дополнительные аспекты обязательности связей, а также кратность участвующих в данных отношениях экземпляров сущностей. Нотация диаграмм (ERD) реализована в различных программных средствах. **Пример диаграммы ERD**, разработанной с помощью средства моделирования бизнес-процессов ARIS изображен на рисунке 1.

Ограниченность диаграмм ERD проявляется при конкретизации концептуальной модели в более детальное представление моделируемой программной системы, которое кроме статических связей должно содержать информацию о поведении или функционировании отдельных ее компонентов.

В рамках диаграмм функционального моделирования было разработано несколько графических языков моделирования, которые получили следующие названия:

1. Нотация IDEF0 - для документирования процессов производства и отображения информации об использовании ресурсов на каждом из этапов проектирования систем.

2. Нотация IDEF1 - для документирования информации о производственном окружении систем.

3. Нотация IDEF2 - для документирования поведения системы во времени.

Нотация IDEF2 никогда не была полностью реализована. Нотация IDEF1 в 1985 году была расширена и переименована в IDEF1X. Методология IDEF нашла применение в правительственных и коммерческих организациях, поскольку в 1993 году появился стандарт FIPS правительства США для двух технологий IDEF0 и IDEF1X. В течение последующих лет этот стандарт продолжал активно развиваться и послужил основой для реализации в некоторых CASE-средствах, наиболее известным из которых является AllFusion Process Modeler (новое название BPwin) компании Computer Associates.

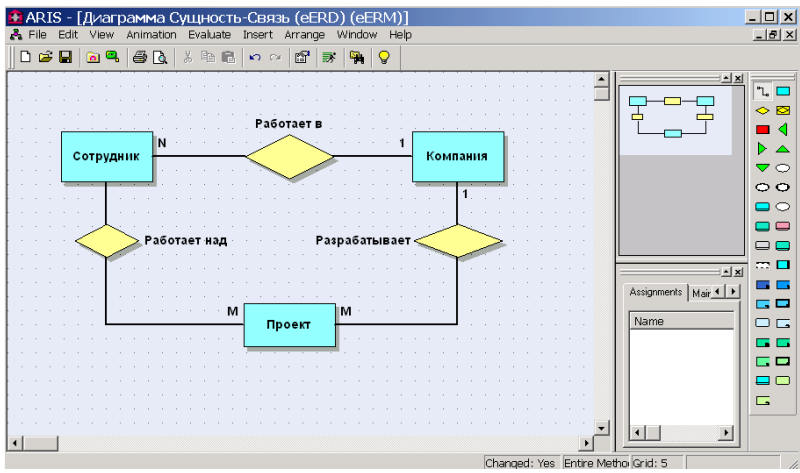


Рисунок 1 - Диаграмма «сущность-связь» для примера сотрудников компании, работающих над различными проектами

Процесс моделирования IDEF представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели системы какой-либо предметной области. Функциональная модель IDEF отображает структуру процессов функционирования системы и ее отдельных подсистем, то есть, выполняемые ими действия и связи между этими действиями. Для этой цели строятся специальные модели, которые позволяют в наглядной форме представить последовательность определенных действий. Исходными строительными блоками любой модели нотации IDEF0 процесса являются деятельность (activity) и стрелки (arrows).

Одна из наиболее важных особенностей нотации IDEF0 - постепенное введение все более детальных представлений модели системы по мере разработки отдельных диаграмм. Построение модели IDEF0 начинается с представления всей системы в виде простейшей диаграммы, состоящей из одного блока процесса и стрелок ICOM, служащих для изображения основных видов взаимодействия субъектами вне системы. Поскольку исходный процесс представляет всю систему как единое целое, данное представление является наиболее общим и подлежит дальнейшей декомпозиции. Пример представления общей модели процесса оформления кредита в банке, разработанной с помощью CASE-средства AllFusion Process Modeler, изображен на рисунке 2.

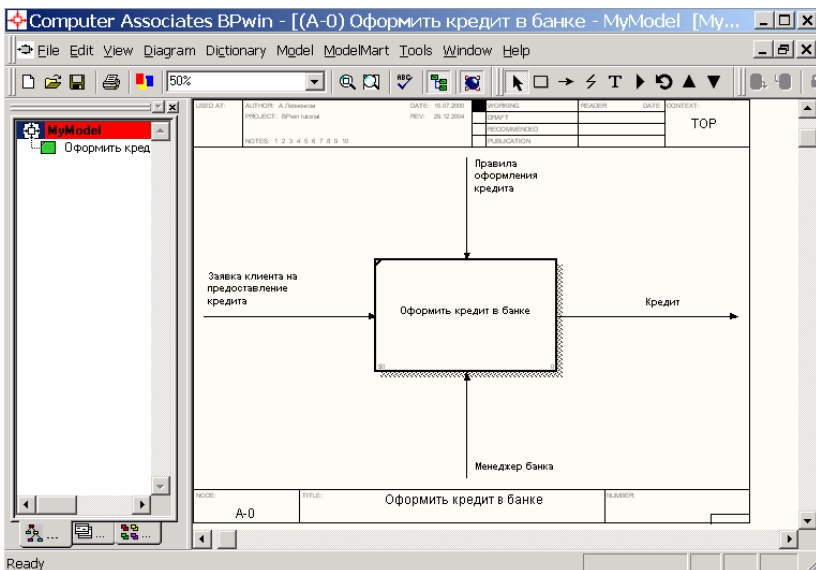


Рисунок 2 - Пример исходной диаграммы IDEF0 для процесса оформления кредита в банке

В конечном итоге модель IDEF0 представляет собой набор иерархически взаимосвязанных диаграмм с сопроводительной документацией, которая разбивает исходное представление сложной системы на отдельные составные части. Детали каждого основного процесса представляются в виде более подробных процессов на других диаграммах. В этом случае каждая диаграмма нижнего уровня является декомпозицией процесса из более общей диаграммы. Поэтому на каждом шаге декомпозиции более общая диаграмма конкретизируется на ряд детальных диаграмм.

Основной недостаток данной методологии связан с отсутствием явных средств для объектно-ориентированного представления моделей сложных систем. Некоторые аналитики отмечают важность знания и применения нотации IDEF0, однако отсутствие возможности реализации соответствующих графических моделей в объектно-ориентированном программном коде существенно сужают диапазон решаемых с ее помощью задач.

В основе графического моделирования информационных систем с помощью диаграмм потоков данных лежит специальная технология построения диаграмм потоков данных DFD. В разработке методологии DFD приняли участие многие аналитики, среди которых следует отметить Э. Йордона. Он автор одной из первых графических нотаций DFD.

Недостаток рассмотренных нотаций связан с отсутствием явных средств для объектно-ориентированного представления моделей сложных систем, а также сложных алгоритмов обработки данных. Поскольку на рассмотренных типах диаграмм не указываются характеристики времени выполнения отдельных процессов и передачи данных между процессами, то модели систем, реализующих синхронную обработку данных, не могут быть адекватно представлены в этих нотациях. Все эти особенности методов структурного системного анализа ограничили возможности широкого применения соответствующих нотаций и послужили основой для разработки унифицированного языка моделирования UML.

4. Практическое занятие №2

«Реализация объектно-ориентированного подхода к проектированию комплексных систем управления с помощью CASE-средства ARIS-EXPRESS»

Цель работы:

1. Изучить основные этапы объектно-ориентированного проектирования комплексных систем управления организацией.
2. Формировать навыки в проектировании комплексных систем управления с применением объектно-ориентированного подхода.

Время: 2 часа.

Место проведения: Лаборатория информационных и мультимедиа технологий.

Обеспечение занятия:

- 1 Конспект - лекций по дисциплине «Комплексные системы управления в структуре бизнеса»
- 2 ПЭВМ с установленной операционной системой Windows XP/7 и пакетом офисных программ.
- 3 Методические указания к выполнению практического занятия по дисциплине «Комплексные системы управления в структуре бизнеса».
- 4 CASE-средство моделирования ARIS Express.

Порядок проведения практического занятия

1. Изучить основные этапы объектно-ориентированного подхода при проектировании комплексных систем управления организацией с помощью CASE-средства (20 мин.).

Используя материалы, представленные в п.3.3 методической разработки студенты изучают процедуру объектно-ориентированного подхода и особенности его описания.

2. Осуществить установку CASE-средства моделирования ARIS Express

Студенты с сайта производителя осуществляют установку на рабочие станции академическую версию программного продукта Арис Экспресс (<http://www.ariscommunity.com/aris-express>).

3. Изучить возможности и особенности интерфейса CASE-средства моделирования ARIS Express (40 мин.).

Студенты с использованием контента портала разработчика (<http://www.ariscommunity.com/aris-express>) и встроенной в программный продукт справочной системы изучают возможности и особенности интерфейса CASE-средства моделирования.

4. Контроль по итогам занятия осуществляется в форме тестирования по теоретическому материалу, представленному п.3.3. методической разработки и знанию возможностей и интерфейса CASE-средства моделирования.

5. Литература

1. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления. Учебник для вузов. – СПб.: Питер, 2009. – 318 с.

2. В. В. Трофимов Информационные системы и технологии в экономике и управлении Издательство: Юрайт, Серия: Основы наук, 2011 г., 528 стр., ил.

3. О. Н. Граничин, В. И. Кияев Информационные технологии в управлении, Издательство: Интернет-университет информационных технологий, Бином. Лаборатория знаний Серия: Основы информационных технологий, 2011 г. , 336 стр., ил.

4. Портал производителя программного обеспечения
<http://www.ariscommunity.com>.