

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«БРАТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Базовая кафедра менеджмента и информационных технологий

УТВЕРЖДАЮ:

Проректор по учебной работе

_____ Е.И. Луковникова

« _____ » декабря 2018 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ АНАЛИЗА И ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Б1.В.08

НАПРАВЛЕНИЕ ПОДГОТОВКИ

38.03.02 Менеджмент

ПРОФИЛЬ ПОДГОТОВКИ

Информационный менеджмент

Программа прикладного бакалавриата

Квалификация (степень) выпускника: бакалавр

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	5
3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ	
3.1 Распределение объёма дисциплины по формам обучения.....	5
3.2 Распределение объёма дисциплины по видам учебных занятий и трудоемкости	5
4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	6
4.1 Распределение разделов дисциплины по видам учебных занятий	6
4.2 Содержание дисциплины, структурированное по разделам и темам	8
4.3 Лабораторные работы.....	20
4.4 Практические занятия.....	20
4.5 Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат.....	20
5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ	20
6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ	22
7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	22
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО – ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	22
9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....	23
9.1. Методические указания для обучающихся по выполнению лабораторных работ/ семинаров / практических работ	24
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	32
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	32
Приложение 1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	33
Приложение 2. Аннотация рабочей программы дисциплины	39
Приложение 3. Протокол о дополнениях и изменениях в рабочей программе	40

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Вид деятельности выпускника

Дисциплина охватывает круг вопросов, относящихся к организационно-управленческому и информационно-аналитическому видам деятельности выпускника в соответствии с компетенциями и видами деятельности, указанными в учебном плане.

Цель дисциплины

овладение обучающимися основными теоретическими положениями в области проектирования автоматизированных информационных систем, знаниями этапов создания и видов создаваемых проектов, а также получение обучающимися практических навыков использования современных подходов в проектировании информационных автоматизированных систем.

Основными задачами дисциплины являются:

- знакомство с концептуальными основами CASE-технологий, с эволюцией развития и классификацией CASE-средств, понятием и основными принципами функционального моделирования, с основными методологиями;
- закрепление навыков работы с инструментами структурного анализа и проектирования SADT (Structure Analysis and Design Technique);
- получение практических навыков построения структурных карт, диаграмм переходов состояний;
- получение практических навыков использования процессного подхода к моделированию деятельности.

Код компетенции 1	Содержание компетенций 2	Перечень планируемых результатов обучения по дисциплине 3
ПК-1	владение навыками использования основных теорий мотивации, лидерства и власти для решения стратегических и оперативных управленческих задач, а также для организации групповой работы на основе знания процессов групповой динамики и принципов формирования команды, умение проводить аудит человеческих ресурсов и осуществлять диагностику организационной культуры	знать: - источники управленческой информации, методики отбора источников информации; уметь: - находить способы решения конфликта в организации и проектирования межличностных, групповых и организационных коммуникаций; владеть: - навыками выполнения заданий в составе команды и организации партнерского взаимодействия для решения стратегических и оперативных управленческих задач;
ПК-11	владение навыками анализа информации о функционировании системы внутреннего документооборота организации, ведения баз	знать: - основные методы, способы и средства получения, хранения, обработки и передачи информации; - виды, состав и структуру документов, отражающих систему внутреннего

	<p>данных по различным показателям и формирования информационного обеспечения участников организационных проектов</p>	<p>документооборота организации;</p> <ul style="list-style-type: none"> - методы и методику ведения баз данных по различным показателям и формирования информационного обеспечения участников организационных проектов; <p>уметь:</p> <ul style="list-style-type: none"> - осуществлять выбор информационных средств для их обработки в соответствии с поставленной задачей; <p>владеть:</p> <ul style="list-style-type: none"> - навыками анализа информации о функционировании системы внутреннего документооборота, проектирования и ведения баз данных; <p>.</p>
ПК-13	<p>Умение моделировать бизнес-процессы и использовать методы реорганизации бизнес-процессов в практической деятельности организаций</p>	<p>знать:</p> <ul style="list-style-type: none"> - методы анализа и моделирования бизнес-процессов - основные стандарты, технологии и нотации моделирования бизнес-процессов - инструментальные системы, используемые для описания и анализа бизнес-процессов; <p>уметь:</p> <ul style="list-style-type: none"> - моделировать, анализировать и совершенствовать бизнес-процессы с использованием изученных стандартов, технологий и нотаций моделирования; - рецензировать модель бизнес-процесса; <p>владеть:</p> <ul style="list-style-type: none"> - практическими навыками моделирования, анализа и документирования бизнес-процессов с помощью инструментальных сред; - терминологией из области моделирования бизнес-процессов; - методами построения, анализа и документирования моделей бизнес-процессов.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Б1.В.08 Современные технологии анализа и проектирования информационных систем относится к вариативной.

Дисциплина Современные технологии анализа и проектирования информационных систем базируется на знаниях, полученных при изучении таких дисциплин, как: Информатика, Информационная система и организация.

Основываясь на изучении перечисленных дисциплин, Современные технологии анализа и проектирования информационных систем представляют основу для изучения дисциплины: Системная интеграция и управление приложениями корпоративных информационных систем, Управление IT-проектами, Инструментальные средства моделирования сложных систем.

Такое системное междисциплинарное изучение направлено на достижение требуемого ФГОС уровня подготовки по квалификации бакалавр.

3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ

3.1. Распределение объема дисциплины по формам обучения

Форма обучения	Курс	Семестр	Трудоёмкость дисциплины в часах						Курсовая работа (проект), контрольная работа, реферат, РГР	Вид промежуточной аттестации
			Всего часов (экс)	Аудиторных часов	Лекции	Лабораторные работы	Практические занятия	Самостоятельная работа		
1	2	3	4	5	6	7	8	9	10	11
Очная	3	6	144	54	18	-	36	63	кр	экзамен
Заочная	4	-	144	18	6	-	12	117	кр	экзамен
Заочная (ускоренное обучение)	-	-	-	-	-	-	-	-	-	-
Очно-заочная	-	-	-	-	-	-	-	-	-	-

3.2. Распределение объема дисциплины по видам учебных занятий и трудоёмкости

Вид учебных занятий	Трудоёмкость (час.)	в т.ч. в интерактивной, активной, инновационной формах, (час.)	Распределение по семестрам, час
			5
1	2	3	4
I. Контактная работа обучающихся с преподавателем (всего)	63	12	63
Лекции (Лк)	18	4	18
Практические занятия (ПЗ)	36	8	36

Групповые консультации	+	-	+
II. Самостоятельная работа обучающихся (СР)	63	-	63
Подготовка к практическим занятиям	27	-	27
Подготовка к экзамену в течение семестра	36	-	36
III. Промежуточная аттестация экзамен	27	-	27
Общая трудоемкость дисциплины. час. зач. ед.	144	-	144
	4	-	4

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Распределение разделов дисциплины по видам учебных занятий

- для очной формы обучения:

№ раздела и темы	Наименование раздела	Трудоемкость, (час.)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость; (час.)		
			учебные занятия		самостоятельная работа обучающихся
			лекции	Практические работы	
1	2	3	4	5	6
1.	Основы анализа и проектирования информационных систем	48	9	12	27
1.1	Основные понятия и терминология проектирования экономических информационных систем управления объектом	4	1	0	3
1.2	Информационные технологии	4	1	0	3
1.3	Жизненный цикл информационной системы	4	1	0	3
1.4	Модели жизненного цикла	4	1	0	3
1.5	Основы анализа и проектирования информационных систем	11	1	4	5
1.6	Технологии и подходы к анализу и проектированию информационных систем	11	2	4	5
1.7	Разработка функциональной модели	11	2	4	5
2.	Разработка информационной модели	40	6	18	16
2.1	Основы проектирования баз данных	4	1	2	1
2.2	Концептуальное проектирование с использованием методологии IDEF1X	13	2	6	5
2.3	Логическое проектирование с использованием методологии IDEF1X	13	2	6	5
2.4	Физическое проектирование с	10	1	4	5

	использованием методологии IDEF1X				
3.	Разработка поведенческой модели	29	3	6	20
3.1	Основы построения поведенческих моделей	7	1	-	6
3.2	Блок-схемы алгоритмов	11	1	4	6
3.3	Правила и рекомендации построения блок-схем	7	1	2	4
	ИТОГО	117	18	36	63

- для заочной формы обучения:

№ раздела и темы	Наименование раздела	Трудоемкость, (час.)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость; (час.)		
			учебные занятия		самостоятельная работа обучающихся
			лекции	Практические работы	
1	2	3	4	5	6
1.	Основы анализа и проектирования информационных систем	65	3	7	55
1.1	Основные понятия и терминология проектирования экономических информационных систем управления объектом	7,25	0,25	0	7
1.2	Информационные технологии	8,25	0,25	0	8
1.3	Жизненный цикл информационной системы	8,1	0,1	0	8
1.4	Модели жизненного цикла	8,25	0,25	0	8
1.5	Основы анализа и проектирования информационных систем	10,25	0,25	2	8
1.6	Технологии и подходы к анализу и проектированию информационных систем	10,25	0,25	2	8
1.7	Разработка функциональной модели	11,15	0,15	3	8
2.	Разработка информационной модели	46	2	4	40
2.1	Основы проектирования баз данных	5,5	0,5	0	5
2.2	Концептуальное проектирование с использованием методологии IDEF1X	18	1	2	15
2.3	Логическое проектирование с использованием методологии IDEF1X	17	1	1	15
2.4	Физическое проектирование с использованием методологии IDEF1X	6,5	0,5	1	5
3.	Разработка поведенческой модели	24	1	1	22
3.1	Основы построения	9	0	0	9

	поведенческих моделей				
3.2	Блок-схемы алгоритмов	8	0,5	0,5	7
3.3	Правила и рекомендации построения блок-схем	7	0,5	0,5	6
	ИТОГО	135	6	12	117

4.2. Содержание дисциплины, структурированное по разделам и темам

Раздел 1. Основы анализа и проектирования информационных систем

1.1 ОСНОВНЫЕ ПОНЯТИЯ И ТЕРМИНОЛОГИЯ ПРОЕКТИРОВАНИЯ ЭКОНОМИЧЕСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ УПРАВЛЕНИЯ ОБЪЕКТОМ (ЭИС);

ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ; СОСТАВ И СОДЕРЖАНИЕ ЭТАПОВ ПРОЕКТИРОВАНИЯ

Под *экономической системой* (ЭС) будем понимать сложную систему, охватывающую процессы производства, обмена, распределения и потребления материальных и других благ.

Система – это совокупность элементов и связей между ними, обладающая определенной целостностью.

Элемент системы – это часть системы, выполняющая определенную функцию. Элемент системы также может быть системой, в этом случае его называют *подсистемой*.

Любая система имеет *организацию* – внутреннюю упорядоченность и согласованность взаимодействия элементов, и *структуру* – совокупность устойчивых внутренних связей между элементами, определяющую основные свойства системы.

Любая система обладает свойством *целостности* – принципиальной несводимости свойств системы к сумме свойств ее элементов.

Классификация систем:

1. В общем плане:

- материальные системы;
- абстрактные системы.

Материальные системы – это совокупность материальных элементов. Материальные системы подразделяются на неорганические (технические, химические, ...), органические (биологические, ...) и смешанные. *Абстрактные* системы – это совокупность знаковых элементов, являющихся продуктом человеческого мышления (знания, теории, гипотезы, языки).

2. По временной зависимости:

- статические системы;
- динамические системы.

Состояние *статической* системы с течением времени не изменяется; состояние *динамической* – изменяется.

3. По предопределенности развития (для динамических систем):

- детерминированные системы;
- стохастические (вероятностные) системы.

Поведение *детерминированной* системы всегда можно предсказать, т.к. ее состояние в любой момент времени полностью определяется состоянием в предшествующий момент. Поведение *вероятностной* системы предсказать невозможно.

4. По взаимодействию с внешней средой:

- закрытые системы;
- открытые системы.

Закрытые системы не взаимодействуют с внешней средой, все процессы, кроме энергетических, замыкаются внутри системы. *Открытые* системы активно взаимодействуют с внешней средой.

5. По сложности:

- простые системы;
- сложные системы.

Простые системы не имеют развитой иерархической структуры; *сложные* состоят из элементов-подсистем.

По приведенной классификации ЭС – смешанная материальная, динамическая, вероятностная, открытая, сложная система. Кроме этого, ЭС относится к классу *кибернетических* систем, т.е. систем с управлением.

Любая кибернетическая система, как система с управлением, включает две подсистемы:

- управляющую подсистему (управляющую систему – УС);
- управляемую подсистему (объект управления – ОУ).

В ЭС объект управления – это структурные подразделения, реализующие основную хозяйственную деятельность, а управляющая система – это подразделения, осуществляющие основные функции управления.

Информационные потоки:

I_1 – информационный поток из внешней среды в УС: информация о цели управления, нормативная информация, информация о конъюнктуре рынка, о конкурентах, о поставщиках, о потребителях и др.

I_2 – информационный поток из УС к ОУ (прямая кибернетическая связь) – совокупность плановой, распорядительной, нормативной внутренней информации, необходимой для осуществления хозяйственной деятельности ЭС.

I_3 – информационный поток от ОУ в УС (обратная кибернетическая связь) – сведения о текущем состоянии ОУ.

I_4 – информационный поток из УС во внешнюю среду: в государственные органы, инвесторам, кредиторам, потребителям и др.

Управление – это процесс целенаправленного воздействия на объект, осуществляемого для организации его функционирования в соответствии с заданной программой или целью.

Цикл (петля) управления:

Прогнозирование и планирование – это функции, определяющие цель функционирования ЭС на различные периоды времени.

Оперативное управление – это функция, осуществляющая регулирование всех хозяйственных процессов в соответствии с прогнозами и планами.

Учет и контроль – это функции, определяющие текущее состояние ОУ и отклонение учетных данных от плановых целей и нормативов.

Анализ – это функция, определяющая необходимость изменений:

А – отклонения отсутствуют → учет и контроль;

Б – существуют незначительные отклонения, нужно удерживать состояние ОУ → оперативное управление;

В – значительные отклонения, нужно изменить состояние ОУ → прогнозирование и планирование.

Управление всегда носит информационный характер, т.е. сопровождается целенаправленной переработкой информации. В этом аспекте функции управления можно объединить в следующие группы:

- **функции принятия решений** – преобразуют содержание информации, являются главными, т.к. именно они обеспечивают выработку управляющих воздействий на ОУ;
- **рутинные функции обработки информации** – не изменяют смыслового содержания информации (накопление, хранение, поиск, отображение в разных формах, тиражирование и др.);
- **функции обмена информацией** – обеспечивают передачу информации по различным каналам связи.

Цель автоматизации управления

Существует два основных подхода к совершенствованию управления:

- сокращение длительности цикла управления;
- повышение качества управляющих воздействий.

Эти подходы носят **противоречивый характер**: при постоянной производительности УС сокращение длительности цикла управления возможно лишь при уменьшении объемов перерабатываемой информации, что негативно сказывается на качестве принимаемых управленческих решений. Решением этой проблемы может стать повышение производительности УС по переработке и передаче информации.

Основные направления:

1. Оптимизация численности управленческого персонала.
2. Использование новых способов организации работы УС.
3. Изменение структуры УС.
4. Перераспределение функций и задач в УС.
5. Автоматизация управленческого труда.

Автоматизация – это комплекс мероприятий (организационных, технических, экономических), который позволяет снизить степень участия или полностью исключить непосредственное участие человека в осуществлении той или иной функции производственного процесса, процесса управления.

Автоматизация реализуется с помощью различных автоматических и автоматизированных систем. В управлении ЭС используют автоматизированные системы, т.к. всегда присутствует лицо, принимающее решение (ЛПР).

Т.о., цель автоматизации управленческого труда – это повышение эффективности управления за счет повышения оперативности управления и научной обоснованности принимаемых решений. Это обуславливает необходимость внедрения в управление новых информационных технологий (ИТ), создания и использования различных информационных систем (ИС).

1.2 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

1.2.1. Основные понятия

Согласно Закону РФ «Об информации, информатизации и защите информации» **информация** (лат. informatio – разъяснение, изложение) – это сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления. В литературе, посвященной компьютерной технике и программным средствам, часто употребляется термин «данные». Как правило, эти понятия взаимозаменяемы.

В то же время необходимо отметить одно принципиальное отличие между ними. **Данные** – это информация, представленная в виде, пригодном для обработки автоматическими средствами [1].

На протяжении последних десятилетий наблюдается неизменное ускорение темпов роста объема информации. Кратко и емко данная ситуация была сформулирована в виде тезиса: «Информация — это единственный неубывающий ресурс общества». Объем информации к 1800 г. удваивался каждые 50 лет, к 1950 г. – каждые 10 лет, к 1970 г. – каждые 5 лет, а к 1990 г. – ежегодно. Следствием такого положения дел стал количественный барьер в процессах обработки информации. Иногда информацию нет смысла собирать и хранить в связи с отсутствием возможности ее обработки и рационального использования. Увеличение информации и растущий спрос на нее обусловили появление отрасли, связанной с автоматизацией обработки информации – информатикой.

Новые информационные технологии призваны обеспечить повсеместную информатизацию всех сфер человеческой деятельности. В [2] **информатизация** определяется как «организационный, социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления, организаций, общественных объединений на основе формирования и использования информационных ресурсов». В этом определении важно отметить, что информатизация – это процесс, а не простой перечень мер.

Информатизация общества немыслима без использования автоматизированных информационных систем. **Автоматизированная информационная система** – это совокупность методического обеспечения, технических (аппаратных) и программных средств, а также работающих с ними пользователей (персонала), обеспечивающая процессы получения, передачи, обработки, хранения и представления информации. **Методическое обеспечение** играет первостепенную и направляющую роль в процессе эксплуатации информационной системы (ИС). Оно определяется [3] как совокупность документов, описывающих технологию функционирования информационной системы, методы выбора и применения пользователями технологических приемов для получения конкретных результатов.

1.2.2 Комплексы стандартов на информационные системы

При разработке промышленных информационных систем одним из ключевых требований является соответствие разрабатываемых систем и их компонентов, а также документации действующим государственным и международным стандартам.

Согласно Закону Российской Федерации «О техническом регулировании» **стандарт** – документ, в котором в целях добровольного многократного использования устанавливаются характеристики продукции, правила осуществления и характеристики процессов производства, эксплуатации, хранения, перевозки, реализации и утилизации, выполнения работ или оказания услуг. Стандарт также может содержать требования к терминологии, символике, упаковке, маркировке и правилам ее нанесения.

Деятельность по установлению правил и характеристик, направленная на достижение упорядоченности в сферах производства и обращения продукции, а также повышение ее конкурентоспособности, работ или услуг, называется *стандартизацией*.

Одним из принципов стандартизации, провозглашенных в статье 12 упомянутого закона, является принцип применения международного стандарта как основы для разработки национального, за исключением некоторых случаев (когда применение стандарта невозможно из-за специфики климатических и географических особенностей РФ, противоречит законодательству РФ и т. п.).

1.3. ЖИЗНЕННЫЙ ЦИКЛ ИНФОРМАЦИОННОЙ СИСТЕМЫ

1.3.1. Структура жизненного цикла информационной системы

Понятие жизненного цикла (ЖЦ) является одним из ключевых понятий методологии проектирования информационных систем. *Жизненный цикл информационной системы* – это непрерывный процесс, начинающийся с момента принятия решения о создании информационной системы и заканчивающийся в момент полного изъятия ее из эксплуатации [4].

Основным стандартом, определяющим структуру жизненного цикла, является ГОСТ Р ИСО/МЭК 12207-02 [5]. Согласно стандарту структура жизненного цикла основывается на трех группах процессов:

- *основные процессы* (заказ, поставка, разработка, эксплуатация, сопровождение);
- *вспомогательные процессы* (обеспечивают выполнение основных процессов):
 - *документирование* – работы по разработке, выпуску, редактированию, распространению и сопровождению документов, в которых нуждаются все заинтересованные лица;
 - *управление конфигурацией* (конфигурационное управление) включает работы: определение и установление состояния программных объектов в системе; управление изменениями и выпуском объектов; обеспечение полноты, совместимости и правильности объектов; управление хранением, обращением и поставкой объектов;
 - *обеспечение качества* – работы по обеспечению соответствия создаваемой системы и реализуемых процессов жизненного цикла установленным требованиям и утвержденным планам;
 - *верификация* – работы соответствующего субъекта (заказчика, поставщика или независимой стороны) по проверке соответствия создаваемых промежуточных результатов установленным требованиям по мере реализации проекта. Различают верификацию договора, процесса, требований, проекта, системы, сборки системы и документации;
 - *аттестация* – работы соответствующего субъекта по проверке полного соответствия требований и конечного продукта функциональному назначению системы;
 - *совместный анализ* – работы по оценке состояния или результатов какой-либо работы (системы);
 - *аудит* – работы независимых (по отношению к проекту) экспертов по определению соответствия деятельности субъекта принятым требованиям, планам и условиям договора;
 - *разрешение проблем* – работы по анализу и устранению проблем, обнаруженных при реализации проекта;
 - *организационные*:
 - *управление проектами* – работы по планированию и управлению процессами, включая контроль, проверку и оценку выполненных работ с формированием отчетности;
 - *создание инфраструктуры проекта* – работы по установлению и обеспечению инфраструктуры, необходимой для любого другого процесса. Инфраструктура может содержать технические и программные средства, инструментальные средства, методики, стандарты и условия для разработки, эксплуатации или сопровождения системы;
 - *усовершенствование* – работы по оценке, контролю и улучшению процессов жизненного цикла;
 - *обучение* – работы по планированию и проведению обучения персонала, включая разработку учебных материалов.

1.3.2. Основные процессы жизненного цикла

В табл. 2.1 предпринята попытка сопоставления стадий классического жизненного цикла (автор Уинстон Ройс, 1970 г.), стандарта [ИСО/МЭК 12207-02](#), [ГОСТ 34.601-90](#) и [ОРММ ИСЖТ 5.03-00](#).

1.3.3. Распределение обязанностей между участниками проекта

В процессе разработки и эксплуатации системы участвует определенный круг лиц (представители заказчика и разработчика), заинтересованных в успешной реализации проекта. В этом процессе между ними распределяются роли, за каждой из которых закрепляется определенный набор функций (обязанностей). При этом один и тот же человек может выступать в разных ролях (качествах). Так, например, один и тот же человек может быть проектировщиком и программистом, в то же время в проекте может принимать участие несколько экспертов, проектировщиков или программистов. В табл. 2.2 приведен типичный список ролей и их функций.

У проекта должен быть один руководитель и, как правило, один системный аналитик. За остальные роли в крупных проектах отвечает обычно по несколько человек. В табл. 2.2 роли эксперта-технолога и пользователя выполняют представители заказчика, остальные роли – представители разработчика. Эксперты-технологи могут быть приглашены из сторонней организации. По мере необходимости в проекте могут принимать участие координатор работ (ответственный) со стороны заказчика, аудиторы и т. д.

1.4. МОДЕЛИ ЖИЗНЕННОГО ЦИКЛА

1.4.1. Классификация моделей жизненного цикла

К настоящему времени наибольшее распространение получили следующие модели (стратегии) жизненного цикла:

- каскадная;
- инкрементная;
- спиральная.

Дальнейшее рассмотрение моделей жизненного цикла ведется с использованием терминологии классического жизненного цикла.

1.4.2. Каскадная стратегия

Каскадная стратегия (однократный проход, водопадная или классическая модель) подразумевает линейную последовательность прохождения стадий создания информационной системы (рис.3.1). Другими словами, переход с одной стадии на следующую происходит только после того, как будет полностью завершена работа на текущей.

Данная модель применяется при разработке информационных систем, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования.

Достоинства модели:

- на каждой стадии формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;

- выполняемые в четкой последовательности стадии позволяют уверенно планировать сроки выполнения работ и соответствующие ресурсы (денежные, материальные и людские).

Недостатки модели:

- реальный процесс разработки информационной системы редко полностью укладывается в такую жесткую схему. Особенно это относится к разработке нетиповых и новаторских систем;
- жизненный цикл основан на точной формулировке исходных требований к информационной системе. Реально в начале проекта требования заказчика определены лишь частично;
- основной недостаток – результаты разработки доступны заказчику только в конце проекта. В случае неточного изложения требований или их изменения в течение длительного периода создания ИС заказчик получает систему, не удовлетворяющую его потребностям.

1.4.3. Инкрементная стратегия

Инкрементная стратегия (англ. increment – увеличение, приращение) подразумевает разработку информационной системы с линейной последовательностью стадий, но в несколько инкрементов (версий), т. е. с запланированным улучшением продукта (рис.3.2).

В начале работы над проектом определяются все основные требования к системе, после чего выполняется ее разработка в виде последовательности версий. При этом каждая версия является законченным и работоспособным продуктом. Первая версия реализует часть запланированных возможностей, следующая версия реализует дополнительные возможности и т. д., пока не будет получена полная система.

Данная модель жизненного цикла характерна при разработке сложных и комплексных систем, для которых имеется четкое видение (как со стороны заказчика, так и со стороны разработчика) того, что собой должен представлять конечный результат (информационная система). Разработка версиями ведется в силу разного рода причин:

- отсутствия у заказчика возможности сразу профинансировать весь дорогостоящий проект;
- отсутствия у разработчика необходимых ресурсов для реализации сложного проекта в сжатые сроки;
- требований поэтапного внедрения и освоения продукта конечными пользователями. Внедрение всей системы сразу может вызвать у ее пользователей неприятие и только «затормозить» процесс перехода на новые технологии. Образно говоря, они могут просто «не переварить большой кусок, поэтому его надо измельчить и давать по частям».

1.4.4. Спиральная стратегия

Спиральная стратегия (эволюционная или итерационная модель, автор Барри Бозм, 1988 г.) подразумевает разработку в виде последовательности версий, но в начале проекта определены не все требования. Требования уточняются в результате разработки версий (рис. 3.3).

Данная модель жизненного цикла характерна при разработке новаторских (нетиповых) систем. В начале работы над проектом у заказчика и разработчика нет четкого видения итогового продукта (требования не могут быть четко определены) или стопроцентной уверенности в успешной реализации проекта (риски очень велики). В связи с этим принимается решение разработки системы по частям с возможностью изменения требований или отказа от ее дальнейшего развития. Как видно из рис.3, развитие проекта может быть завершено не только после стадии внедрения, но и после стадии анализа риска.

Достоинства модели:

- позволяет быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований;
- допускает изменение требований при разработке информационной системы, что характерно для большинства разработок, в том числе и типовых;
- обеспечивает большую гибкость в управлении проектом;
- позволяет получить более надежную и устойчивую систему. По мере развития системы ошибки и слабые места обнаруживаются и исправляются на каждой итерации;
- позволяет совершенствовать процесс разработки – анализ, проводимый в каждой итерации, позволяет проводить оценку того, что должно быть изменено в организации разработки, и улучшить ее на следующей итерации;
- уменьшаются риски заказчика. Заказчик может с минимальными для себя финансовыми потерями завершить развитие перспективного проекта.

Недостатки модели:

- увеличивается неопределенность у разработчика в перспективах развития проекта. Этот недостаток вытекает из предыдущего достоинства модели;
- затруднены операции временного и ресурсного планирования всего проекта в целом. Для решения этой проблемы необходимо ввести временные ограничения на каждую из стадий жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа выполнена. План составляется на основе статистических данных, полученных в предыдущих проектах и личного опыта разработчиков.

При разработке системы под итоговым продуктом и промежуточным программным обеспечением согласно [12] следует понимать:

- *ревизию* (исправительную или опытную) – любые оперативные изменения программного и информационного обеспечения, а также БД, необязательные в данный момент к передаче на объекты внедрения и связанные с устранением ошибок и усовершенствованием;
- *модификацию* – любые оперативные изменения программного и информационного обеспечения, а также БД, обязательные для передачи на объекты внедрения и обуславливающие изменение эксплуатационных характеристик без изменения функций (предусмотренных ТЗ), а также изменения, связанные с устранением ошибок, усовершенствованием;
- *версию* – любые изменения программного и информационного обеспечения, а также БД, обязательные для передачи на объекты внедрения, позволяющие выполнять заявленные или дополнительные функции, а также обеспечивающие переход на новые операционные системы и информационную среду;
- *развитие (очередь)* – плановые изменения информационной системы, связанные с введением новых функций и улучшением эксплуатационных характеристик, переходом на новую информационную среду, внедрением новых комплексов технических средств, новых информационных технологий и пр.

1.4.6. Методологии, поддерживающие спиральную модель

В настоящее время имеется несколько методологий¹ разработки программного обеспечения, которые можно рекомендовать при использовании спиральной модели жизненного цикла. Наиболее известными из них являются методология быстрой разработки приложений (Rapid Application Development, RAD) и экстремальное программирование (eXtreme Programming, XP – автор Кент Бек, 1999).

Методология RAD [4]. На начальном этапе существования компьютерных информационных систем их разработка велась на традиционных языках программирования и подразумевала, как правило, ручной набор текстов программ. Однако по мере возрастания сложности разрабатываемых систем и увеличения запросов пользователей потребовались новые средства, обеспечивающие значительное сокращение сроков разработки. Это послужило предпосылкой к созданию инструментальных средств для быстрой разработки приложений. Развитие этого направления привело к появлению на рынке разработки программного обеспечения средств автоматизации практически всех стадий жизненного цикла.

Под RAD-разработкой обычно понимается процесс разработки, содержащий 3 элемента:

- небольшую команду программистов (до 10 человек);
- короткий, но тщательно проработанный производственный график (от 2 до 6 месяцев);
- повторяющийся цикл, при котором разработчики по мере того, как приложение начинает обретать форму, реализуют в продукте требования, полученные через взаимодействие с заказчиком.

1.5. ОСНОВЫ АНАЛИЗА И ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

1.5.1. Особенности анализа и проектирования крупных систем

Тенденции развития современных информационных технологий приводят к постоянному возрастанию сложности разрабатываемых информационных систем. Для них характерны следующие особенности [14]:

- сложность описания (достаточно большое количество функций, процессов, элементов данных и сложные взаимосвязи между ними) требует тщательного моделирования и анализа данных и процессов;
- наличие совокупности тесно взаимодействующих компонентов (подсистем);
- отсутствие прямых аналогов, ограничивающее возможность использования каких-либо типовых проектных решений и прикладных систем;
- необходимость интеграции существующих и вновь разрабатываемых подсистем;
- функционирование в неоднородной среде на разных аппаратных и операционных платформах;
- разобщенность и разнородность отдельных групп разработчиков по уровню квалификации и сложившимся традициям использования тех или иных инструментальных средств;
- существенная временная протяженность проекта, обусловленная ограниченными возможностями коллектива разработчиков, масштабами организации-заказчика, различной степенью готовности отдельных ее подразделений к внедрению информационных систем и т. д.;
- изменение или уточнение потребностей пользователей в процессе разработки и эксплуатации системы.

1.5.2. Документы, содержащие требования на разработку системы

Основными документами, содержащими требования на разработку информационной системы, являются календарный план выполнения работ и техническое задание. Первый из них регламентирует состав, сроки и финансирование работ, а второй – основные требования к системе.

Пример календарного плана, являющегося обязательным приложением к договору.

Как было отмечено выше, техническое задание (ТЗ) является основным документом, определяющим требования и порядок создания (развития или модернизации) системы. Несмотря на то, что согласно [9, 13] ТЗ составляется после заключения договора, нередко оно должно быть подготовлено разработчиком еще до его подписания.

Состав, содержание, правила оформления этого документа устанавливаются ГОСТ 34.602–89 «Техническое задание на создание автоматизированной системы» [3]. ТЗ, как правило, содержит следующие разделы:

- общие сведения (наименование системы; наименование предприятий разработчика и заказчика с их реквизитами; перечень документов, на основании которых создается система, плановые сроки начала и окончания работы и т. д.);
- назначение и цели создания (развития) системы;
- характеристика объектов автоматизации;
- требования к системе в целом, к функциям и обеспечению. При этом выделяют следующие виды обеспечения:
 - *математическое* – совокупность математических методов, моделей и алгоритмов, применяемых в информационной системе;
 - *информационное* – совокупность форм документов, классификаторов, нормативной базы и реализованных решений по объемам, размещению и формам существования информации;

1.5.3. Основные принципы проектирования

Процесс перехода от первичного описания системы в виде технического задания к ее описанию в виде набора стандартных документов (проектной документации), достаточных для создания системы, называется *проектированием*.

Все наиболее распространенные методологии анализа и проектирования информационных систем при построении моделей базируются на ряде общих принципов [14, 16, 17].

1. Принцип **декомпозиции** ("разделяй и властвуй") – принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения. Применительно к проектированию информационных систем, данный принцип подразумевает разбиение на модули (модели или их элементы).

2. Принцип **иерархического упорядочения** – принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне. Этот принцип предписывает рассматривать процесс построения модели системы на разных уровнях абстрагирования и детализации в рамках фиксированных представлений. Таким образом, проектирование можно представить как поуровневый спуск от наиболее общих и абстрактных моделей системы к более частным и детальным. При разработке программного обеспечения с помощью объектно-ориентированного подхода данный принцип получил название «наследование» – принцип, в соответствии с которым знание об общей категории разрешается применять для более узкой.

3. Принцип **концептуальной общности** заключается в следовании единой философии на всех стадиях жизненного цикла (например, структурный анализ ® структурное проектирование ® структурное программирование ® структурное тестирование).

4. Принцип **абстрагирования** заключается в выделении существенных элементов системы и отвлечении от несущественных. Другими словами, этот принцип предписывает включать в модель только те элементы проектируемой системы, которые имеют непосредственное отношение к выполнению системой своих функций.

1.5.4. Классификация моделей информационной системы

Как было отмечено выше, при анализе и особенно при проектировании системы должны быть построены ее полные и непротиворечивые модели. При этом под *моделью* понимается совокупность взаимосвязанных абстрактных элементов с возможным указанием их свойств, поведения и связей между ними.

Классифицировать модели можно по следующим признакам.

1. По строгости описания:

- *неформальные* – представлены в неструктурированном виде и дают общее представление о моделируемой системе. Недостаточно наглядны (особенно при сложном взаимодействии между объектами) и неприемлемы для какого-либо количественного анализа и обработки автоматическими средствами;
- *формальные*:
 - *описательные* – модели, где сведения представлены с помощью специальных документов (бланки, формы, анкеты, таблицы и т. п.);
 - *графические* – модели представляют собой схемы, чертежи, графы, диаграммы и т. д. Наиболее наглядны и получили широкое распространение при проектировании с помощью CASE-средств;
 - *математические* – представляют модель на языке математических отношений в виде функциональных зависимостей, систем алгебраических или дифференциальных уравнений, логических выражений и т. д.

2. По степени физической реализации (логической независимости):

- *логические* – описывают состав, структуру, состояние или поведение элементов системы без привязки к конкретным языкам или средам программирования, СУБД, техническим средствам и т. д. При разработке системы это обеспечивает гибкость в выборе и быстрый переход с одной программно-аппаратной платформы на другую;
- *физические* – описывают элементы системы в соответствии с принятой физической реализацией этих элементов (языками программирования, СУБД, устройствами, и т. д.);

3. По степени отображения динамики происходящих процессов:

- *статические* – описывают состав и структуру системы;
- *динамические* – описывают поведение системы и/или ее отдельных элементов. Как правило, такие модели описывают порядок действий или состояния системы и переходы между ними. Другими словами, в этих моделях явно или не явно присутствует понятие времени;

4. По отображаемому аспекту:

- *функциональные* – описывают функции системы, возможные варианты ее использования; могут содержать сведения о циркулирующей в системе информации, объектах и субъектах, взаимодействующих с системой; могут быть как динамическими, так и статическими моделями;
- *информационные* – описывают состав и структуру данных (реляционных БД, классов и др.). Относятся к статическим моделям;
- *поведенческие* – описывают состояния системы и/или ее отдельных элементов и переходы между ними, взаимодействие элементов, алгоритмы обработки информации. Относятся к динамическим моделям;
- *компонентные* – описывают состав и структуру программных и аппаратных средств. Относятся к статическим моделям;
- *смешанные* – характеризуют сразу несколько аспектов системы (например, диаграммы потоков данных отображают работы, накопители данных, подсистемы) и т. д.

1.6. ТЕХНОЛОГИИ И ПОДХОДЫ К АНАЛИЗУ И ПРОЕКТИРОВАНИЮ ИНФОРМАЦИОННЫХ СИСТЕМ

1.6.1. CASE-технологии анализа и проектирования

Максимально упростить и формализовать процессы формирования требований и проектирования системы позволяют современные CASE-средства.

Основная цель использования CASE-технологий заключается в максимальной автоматизации стадий анализа и проектирования систем с целью построения формальных и непротиворечивых моделей системы.

Другая, не менее важная, цель использования CASE-технологий – вынесение части деятельности (чем больше, тем лучше) из стадии кодирования в стадию проектирования.

Большинство современных CASE-средств поддерживает методологии *структурного* и/или *объектно-ориентированного* анализа и проектирования информационных систем. Выбор того или иного подхода (парадигмы²) подразумевает следование ему и на стадии кодирования (согласно принципу концептуальной общности). *Их отличие друг от друга заключается в выборе способа декомпозиции системы (задачи). Если за основу принимается функциональная (алгоритмическая) декомпозиция, то речь идет о структурном подходе, если объектная – об объектно-ориентированном.*

Выбор того или иного подхода зависит от специфики решаемой задачи. Как правило, структурный подход применяется для автоматизации задач, оперирующих большими объемами «пассивных» данных и ориентированных на использование реляционных баз данных (например, учет, сбор статистики, математические и инженерные расчеты, анализ данных). Объектно-ориентированный подход в основном ориентирован на решение задач, в которых четко прослеживается деление системы на взаимодействующие между собой сущности (например, имитационное моделирование, управление техническими объектами или технологическими процессами, мониторинг³). Наиболее характерна эта особенность для распределенных систем.

1.6.2. Сущность структурного анализа и проектирования

Программы, написанные на первых языках программирования, как и первая программа, написанная лично, представляли собой простое перечисление команд, выполняемых от начала до конца. Естественно, в этих программах уже присутствовали управляющие конструкции типа циклов, условных и безусловных переходов и т. д. Но по мере усложнения программ (увеличения кода) их разработка и сопровождение при таком подходе становились все более затруднительными. Тогда в языках стали появляться дополнительные конструкции (функции и процедуры), возможность модульной разработки программы и последующая ее сборка из разных файлов. Программа стала представлять собой не просто единое целое с трудным различимым внутренним устройством (типа «черного ящика»), а структуру, состоящую из четко выраженных модулей, связанных между собой определенными отношениями (интерфейсами). То есть *программа приобрела структуру иерархической многоуровневой модульной системы*. Каждый уровень такой системы является законченным модулем, поддерживаемым и контролируемым модулем, находящимся над ним.

В начале разрабатывается функциональная модель, с помощью которой определяются, анализируются и фиксируются требования к составу и структуре функций системы, т. е. определяется, для каких целей разрабатывается система, какие функции она будет выполнять. На этой же модели указываются исходная информация, промежуточные и итоговые результаты работы системы. На основе информационных потоков определяется состав и структура необходимых данных, хранимых в системе (строится информационная модель). Далее, с учетом разработанных моделей, создаются процедуры реализации функции, т. е. алгоритмы обработки данных и поведения элементов системы. На заключительной стадии устанавливается распределение функций по подсистемам (компонентам), необходимое техническое обеспечение и строится модель их распределения по узлам системы.

1.7. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ МОДЕЛИ

1.7.1. Основы функционального анализа и проектирования систем

Перед разработкой системы (лекция 4) заказчик и разработчик должны ясно представлять, какие функциональные возможности будут заложены в систему и как будет организовано функциональное взаимодействие внутри системы.

При разработке функциональной модели (определении функциональных требований) может возникнуть множество проблем:

- заказчик не может точно выразить, решение каких задач возлагается на информационную систему. Зачастую заказчик даже не знает, что такое требование и как его формулировать;
- представители заказчика (начальники разных уровней, эксперты-технологи, рядовые пользователи) по-своему видят работу будущей системы и часто их требования к системе носят взаимоисключающий характер. Особенно характерна такая ситуация, когда разрабатываемая система будет внедряться на нескольких объектах автоматизации;
- заказчик зачастую не знает возможностей современных вычислительных систем и стремится рассматривать процесс автоматизации как простой перенос элементарных видов деятельности, выполняемых вручную, на компьютеры.

1.7.2. Назначение и состав методологии SADT (IDEF0)

Методология SADT (Structured Analysis and Design Technique – методология структурного анализа и проектирования) представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели системы.

Данная методология при описании функционального аспекта информационной системы конкурирует с методами, ориентированными на потоки данных (DFD). В отличие от них IDEF0 позволяет:

- описывать любые системы, а не только информационные (DFD предназначена для описания программного обеспечения);
- создать описание системы и ее внешнего окружения до определения окончательных требований к ней. Иными словами, с помощью данной методологии можно постепенно выстраивать и анализировать систему даже тогда, когда трудно еще представить ее воплощение.

Таким образом, IDEF0 может применяться на ранних этапах создания широкого круга систем. В то же время она может быть использована для анализа функций существующих систем и выработки решений по их улучшению.

Основу методологии IDEF0 составляет графический язык описания процессов. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе.

Модель (AS-IS, TO-BE или SHOULD-BE) может содержать 4 типа диаграмм]:

- контекстную диаграмму;
- диаграммы декомпозиции;
- диаграммы дерева узлов;
- диаграммы только для экспозиции (for exposition only, FEO).

Контекстная диаграмма (диаграмма верхнего уровня), являясь вершиной древовидной структуры диаграмм, показывает назначение системы (основную функцию) и ее взаимодействие с внешней средой. В каждой модели может быть только одна контекстная диаграмма. После описания основной функции выполняется функциональная декомпозиция, т. е. определяются функции, из которых состоит основная.

Далее функции делятся на подфункции и так до достижения требуемого уровня детализации исследуемой системы. Диаграммы, которые описывают каждый такой фрагмент системы, называются *диаграммами декомпозиции*. После каждого сеанса декомпозиции проводятся сеансы экспертизы – эксперты предметной области указывают на соответствие реальных процессов созданным диаграммам. Найденные несоответствия устраняются, после чего приступают к дальнейшей детализации процессов.

Диаграмма дерева узлов показывает иерархическую зависимость функций (работ), но не связи между ними. Их может быть сколько угодно, поскольку дерево можно построить на произвольную глубину и с произвольного узла.

Диаграммы для экспозиции строятся для иллюстрации отдельных фрагментов модели с целью отображения альтернативной точки зрения на происходящие в системе процессы (например, с точки зрения руководства организации).

1.7.3. Элементы графической нотации IDEF0

Методология IDEF0 нашла широкое признание и применение, в первую очередь, благодаря простой графической нотации, используемой для построения модели. Главными компонентами модели являются диаграммы. На них отображаются функции системы в виде прямоугольников, а также связи между ними и внешней средой посредством стрелок. Использование всего лишь двух графических примитивов (прямоугольник и стрелка) позволяют быстро объяснить правила и принципы построения диаграмм IDEF0 людям, незнакомым с данной методологией. Это достоинство позволяет подключить и активизировать деятельность заказчика по описанию бизнес-процессов с использованием формального и наглядного графического языка.

Прямоугольник представляет собой *работу* (*процесс, деятельность, функцию или задачу*), которая имеет фиксированную цель и приводит к некоторому конечному результату. Имя работы должно выражать действие (например, «Изготовление детали», «Расчет допускаемых скоростей», «Формирование ведомости ЦДЛ № 3»).

1.7.4. Типы связей между работами

После определения состава функций и взаимосвязей между ними, возникает вопрос о правильной их композиции (объединении) в модули (подсистемы). При этом подразумевается, что каждая отдельная функция должна решать одну, строго определенную задачу. В противном случае необходима дальнейшая декомпозиция или разделение функций.

При объединении функций в подсистемы необходимо стремиться, чтобы внутренняя связность (между функциями внутри модуля) была как можно сильнее, а внешняя (между функциями, входящими в разные модули), как можно слабее. Опираясь на семантику связей методологии IDEF0, введем классификацию связей между функциями (работами). Данная классификация является расширением [14]. Типы связей приводятся в порядке уменьшения их значимости (силы связывания). В приводимых примерах утолщенными линиями выделяются функции, между которыми имеется рассматриваемый тип связи.

1. *Иерархическая связь* (связь «часть» – «целое») имеет место между функцией и подфункциями, из которых она состоит

2. *Регламентирующая (управляющая, подчиненная) связь* отражает зависимость одной функции от другой, когда выход одной работы направляется на управление другой. Функцию, из которой выходит управление, следует считать регламентирующей или управляющей, а в которую входит – подчиненной. Различают *прямую связь по управлению*, когда

управление передается с вышестоящей работы на нижестоящую (рис. 6.3), и *обратную связь по управлению*, когда управление передается от нижестоящей к

3. *Функциональная (технологическая) связь* имеет место, когда выход одной функции служит входными данными для следующей функции. С точки зрения потока материальных объектов данная связь показывает технологию (последовательность работ) обработки этих объектов. Различают *прямую связь по входу*, когда выход передается с вышестоящей работы на нижестоящую (рис. 6.5), и *обратную связь по входу*, когда выход передается с нижестоящей к вышестоящей

4. *Потребительская связь* имеет место, когда выход одной функции служит механизмом для следующей функции. Таким образом, одна функция потребляет ресурсы, вырабатываемые другой

Рис. 6.7. Потребительская связь

5. *Логическая связь* наблюдается между логически однородными функциями (рис. 6.8). Такие функции, как правило, выполняют одну и ту же работу, но разными (альтернативными) способами или, используя разные исходные данные (материалы).

6. *Коллегиальная (методическая) связь* имеет место между функциями, алгоритм работы которых определяется одним и тем же управлением (рис. 6.9). Аналогом такой связи является совместная работа сотрудников одного отдела (коллег), подчиняющихся начальнику, который отдает указания и приказы (управляющие сигналы). Такая связь также возникает, когда алгоритмы работы этих функций определяются одним и тем же методическим обеспечением (СНИП, ГОСТ, официальные нормативными материалами и т. д.), служащим в качестве управления.

7. *Ресурсная связь* возникает между функциями, используемыми для своей работы одни и те же ресурсы (рис. 6.10). Ресурсно-зависимые функции, как правило, не могут выполняться одновременно.

1.7.5. Правила и рекомендации построения диаграмм IDEF0

В IDEF0 существуют соглашения (правила и рекомендации) по созданию диаграмм, которые призваны облегчить чтение и экспертизу модели [14, 15, 18, 19]. Некоторые из этих правил CASE-средства поддерживают автоматически, выполнение других следует обеспечить вручную.

1. Перед построением модели необходимо определиться, какая модель (модели) системы будет построена. Это подразумевает определение ее типа AS-IS, TO-BE или SHOULD-BE, а также определения позиции, с точки зрения которой строится модель. «Точку зрения» лучше всего представлять себе как место (позицию) человека или объекта, в которое надо встать, чтобы увидеть систему в действии. Например, при построении модели работы продуктового магазина можно среди возможных претендентов, с точки зрения которых рассматривается система, выбрать продавца, кассира, бухгалтера или директора. Обычно выбирается одна точка зрения, наиболее полно охватывающая все нюансы работы системы, и при необходимости для некоторых диаграмм декомпозиции строятся диаграммы FEO, отображающие альтернативную точку зрения.

2. На контекстной диаграмме отображается один блок, показывающий назначение системы. Для него рекомендуется отображать по

2–4 стрелки, входящие и выходящие с каждой стороны.

3. Количество блоков на диаграммах декомпозиции рекомендуется в пределах 3–6. Если на диаграмме декомпозиции два блока, то она, как правило, не имеет смысла. При наличии большого количества блоков диаграмма становится перенасыщенной и трудно читаемой.

4. Блоки на диаграмме декомпозиции следует располагать слева направо и сверху вниз. Такое расположение позволяет более четко отразить логику и последовательность выполнения работ. Кроме этого маршруты стрелок будут менее запутанными и иметь минимальное количество пересечений.

5. Отсутствие у функции одновременно стрелок управления и входа не допускается. Это означает, что запуск данной функции не контролируется и может произойти в любой произвольный момент времени либо вообще никогда (рис. 6.14).

Блок с наличием только управления можно рассматривать как вызов в программе функции (процедуры) без параметров. Если у блока имеется вход, то он эквивалентен вызову в программе функции с параметрами. Таким образом, блок без управления и входа эквивалентен функции, которая в программе ни разу не вызывается на исполнение.

На рис. 6.7–6.12, отображающих фрагменты диаграмм IDEF0, встречаются блоки без входа и управления. Это не стоит рассматривать как ошибку, так как подразумевается, что одна из этих стрелок должна быть.

6. У каждого блока должен быть как минимум один выход (рис. 6.15).

Работы без результата не имеют смысла и не должны моделироваться. Исключение составляют работы, отображаемые в модели AS-IS. Их наличие свидетельствует о неэффективности и несовершенстве технологических процессов. В модели TO-BE эти работы должны отсутствовать.

7. При построении диаграмм следует минимизировать число пересечений, петель и поворотов стрелок.

8. Обратные связи и итерации (циклические действия) могут быть изображены с помощью обратных дуг. Обратные связи по входу рисуются «нижней» петлей, обратная связь по управлению – «верхней» (см. рис. 6.4 и 6.6).

9. Каждый блок и каждая стрелка на диаграммах должны обязательно иметь имя. Допускается использовать ветвление (декомпозицию) или слияние (композицию) стрелок (рис. 6.16). Это связано с тем, что одни и те же данные или объекты, порожденные одной работой, могут использоваться сразу в нескольких других работах. И наоборот, одинаковые или однородные данные и объекты, порожденные разными работами, могут использоваться в одном месте.

1.7.6. Пример построения модели IDEF0 для системы определения допускаемых скоростей

Расчет допускаемых скоростей движения поездов является трудоемкой инженерной задачей. При проходе поездом какого-либо участка фактическая скорость движения поезда не должна превышать предельно допускаемую. Эта предельно допускаемая скорость устанавливается исходя из опыта эксплуатации и специально проводимых испытаний по динамике движения и воздействию на путь подвижного состава. Непревышение этой скорости гарантирует безопасность движения поездов, комфортабельные условия езды пассажиров и т. п. Они определяются в зависимости от типа подвижного состава (марки локомотива и типа вагонов), параметров верхнего строения пути (типа рельсов, балласта, эпюры шпал) и плана (радиуса кривых, переходных кривых, возвышения наружного рельса и т. д.). Как правило, для установления допускаемых скоростей необходимо определить не менее двух (на прямых) и пяти (в кривых) скоростей, из которых и выбирается окончательная допускаемая скорость, как наименьшая из всех рассчитанных. Расчет этих скоростей регламентируется Приказом МПС России № 41 от 12 ноября 2001 г. «Нормы допускаемых скоростей движения подвижного состава по железнодорожным путям колеи 1520 (1524) мм Федерального железнодорожного транспорта».

Как было отмечено, построение модели IDEF0 начинается с представления всей системы в виде простейшей компоненты (контекстной диаграммы). Данная диаграмма отображает назначение (основную функцию) системы и необходимые входные и выходные данные, управляющую и регламентирующую информацию, а также механизмы.

Контекстная диаграмма для задачи определения допускаемых скоростей показана на рис.6.21. Для построения модели использовался продукт VPwin 4.0 фирмы Computer Associates.

В качестве *исходной информации*, на основе которой выполняется определение допускаемых скоростей, используются:

- данные проекта новой линии или проекта реконструкции (содержат всю необходимую информацию для реализации проекта, а именно километраж, оси отдельных пунктов, план линии и др.);
- подробный продольный профиль (содержит информацию, аналогичную рассмотренной выше);
- паспорт дистанции пути (содержит информацию, аналогичную рассмотренной выше, а также сведения о верхнем строении пути (ВСП));
- данные о результатах съемки плана пути вагоном-путеизмерителем;
- ведомость возвышений наружного рельса в кривых (содержит информацию о плане пути).

Часть исходной информации может быть взята из разных источников. В частности сведения о плане (параметрах кривых) могут быть взяты из проекта новой линии или проекта реконструкции, подробного продольного профиля, паспорта дистанции пути и т. д.

1.7.7. ICOM-коды

Стрелки, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются теми же самыми, что и стрелки, входящие в диаграмму нижнего уровня и выходящие из нее, потому что блок и диаграмма представляют одну и ту же часть системы (см. рис. 6.21 и 6.22). Как следствие этого, границы функции верхнего уровня – это то же самое, что и границы диаграммы декомпозиции.

ICOM-коды (аббревиатура от Input, Control, Output и Mechanism) предназначены для идентификации граничных стрелок. ICOM-код содержит префикс, соответствующий типу стрелки (I, C, O или M), и порядковый номер (см. рис. 6.22).

1.7.8. Назначение и состав DFD

При построении функциональной модели системы альтернативой методологии SADT (IDEF0) является методология *диаграмм потоков данных* (Data Flow Diagrams, DFD). В отличие от IDEF0, предназначенной для проектирования систем вообще, DFD предназначена для проектирования информационных систем. Ориентированность этой методологии на проектирование автоматизированных систем делает ее удобным и более выгодным инструментом при построении функциональной модели ТО-ВЕ.

Как и в IDEF0 основу методологии DFD составляет графический язык описания процессов. Авторами одной из первых графических нотаций DFD (1979 г.) стали Эд Йордан (Yourdon) и Том де Марко (DeMarko).

В настоящее время наиболее распространенной является нотация Гейна-Сарсона (Gane-Sarson).

Модель системы в нотации DFD представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе. Модель системы содержит контекстную диаграмму и диаграммы декомпозиции.

Принципы построения функциональной модели с помощью DFD аналогичны принципам методологии IDEF0. Вначале строится контекстная диаграмма, где отображаются связи системы с внешним окружением. В дальнейшем выполняется декомпозиция основных процессов и подсистем с построением иерархии диаграмм.

1.7.9. Элементы графической нотации DFD

Согласно DFD источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам. Те в свою очередь преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных или внешним сущностям – потребителям

Поток данных определяет информацию (материальный объект), передаваемую через некоторое соединение от источника к приемнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой и т. д.

Каждый поток данных имеет имя, отражающее его содержание. Направление стрелки показывает направление потока данных. Иногда информация может двигаться в одном направлении, обрабатываться и возвращаться назад в ее источник. Такая ситуация может моделироваться либо двумя различными потоками, либо одним – двунаправленным.

На диаграммах IDEF0 потоки данных соответствуют входам и выходам, но в отличие от IDEF0 стрелки потоков на DFD могут отображаться входящими и выходящими из любой грани внешней сущности, процесса или накопителя данных.

Процесс (в IDEF0 – функция, работа) представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом.

1.7.10. Правила и рекомендации построения DFD

Правила и рекомендации построения модели DFD в основном совпадают с принятыми в IDEF0. Часть из них приведена в подразд. 6.9.

По аналогии с IDEF0 у каждого процесса (подсистемы) на диаграмме потоков данных должен быть как минимум один входящий и один выходящий поток. Процесс должен запускаться на выполнение либо через обрабатываемый, либо через управляющий поток данных. Работа каждого процесса должна завершаться конкретным результатом.

Каждый накопитель данных должен иметь как минимум один входящий и один выходящий поток. Наличие только входящих потоков в накопитель означает, что информация накапливается, но не используется.

Наличие только выходящих потоков из накопителя также является ошибкой. Прежде чем использовать данные из накопителя, они должны там появиться в результате работы какого-либо процесса (подсистемы, внешней сущности). Исключением из правил считается случай, когда накопитель является внешней сущностью. Тогда допускается наличие либо только входящих стрелок, либо только выходящих стрелок (см. рис. 6.23, накопитель «БД АРМ-П или СБД-П»).

1.7.11. Пример построения модели DFD для системы определения допускаемых скоростей

Описание задачи приведено в подразд. 6.6.

Построение функциональной модели DFD начинается как и в IDEF0 с разработки контекстной диаграммы. На ней отображается основной процесс (сама система в целом) и ее связи с внешней средой (внешними сущностями). Это взаимодействие показывается через потоки данных. Допускается на контекстной диаграмме отображать сразу несколько основных процессов или подсистем.

На этой диаграмме видно, что в качестве источника исходных данных для работы системы могут использоваться базы данных АРМ-П (АРМ службы пути) или СБД-П (Сводная БД – Путьский фрагмент), содержащие практически всю необходимую информацию по участкам дороги.

В то же время в системе оставлена возможность ее ручного ввода и корректировки. Несмотря на то, что БД АРМ-П или СБД-П по отношению к системе являются внешними сущностями, они, в целях лучшего восприятия, показаны в виде накопителя данных.

Дальнейший процесс проектирования состоит в построении диаграмм декомпозиции, которые строятся (показывают устройство) *только для процессов или подсистем (систем)*.

Диаграмма декомпозиции первого уровня проектируемой системы приведена

1.7.12. Расширения DFD для систем реального времени

Системы реального времени построены, как правило, на взаимодействии средств вычислительной техники и различных физических устройств съема информации (датчиков, камер, микрофонов и т. д.). Первые являются дискретными преобразователями информации, вторые в основном – аналоговыми, т. е. генерирующими информацию в виде непрерывного потока. Другой особенностью таких систем является значительный уклон в сторону управления объектами. Для моделирования особенностей поведения систем реального времени П. Вард и С. Меллор предложили использовать на DFD дополнительные элементы.

Квазинепрерывный поток (лат. quasi – как будто, якобы) – поток данных, непрерывный во времени. Отображается линией с двумя стрелками на

Управляющий процесс – процесс, формирующий сигналы управления на

Управляющий поток – управляющая информация, запускающая процесс (подсистему) или изменяющая ход его выполнения

Использование управляющих потоков позволяет отделить управляющую информацию от обрабатываемой, как это делается на диаграммах IDEF0.

Накопитель управлений – накопитель управляющих потоков

Раздел 2. Разработка информационной модели (12 час. компьютерная презентация)

2.1. Основы проектирования баз данных

Разработанная функциональная модель системы отвечает на вопросы «Что должна делать система?» и «За счет каких действий может быть достигнут требуемый результат?». Эта модель также позволяет концептуально определить наборы данных, используемых в системе.

В то же время она не отвечает на вопрос «Каким образом организованы данные в системе?». Для ответа на него необходимо построить информационную модель (запроектировать БД).

Традиционно процедуру проектирования базы данных разбивают на три этапа, каждый из которых завершается созданием соответствующей информационной модели [1, 20, 21].

Этап 1-й. Концептуальное проектирование – создание представления (схемы, модели) БД, включающего определение важнейших сущностей (таблиц) и связей между ними, но не зависящего от модели БД (иерархической, сетевой, реляционной и т. д.) и физической реализации (целевой СУБД).

Этап 2-й. Логическое проектирование – развитие концептуального представления БД с учетом принимаемой модели (иерархической, сетевой, реляционной и т. д.).

Этап 3-й. Физическое проектирование – развитие логической модели БД с учетом выбранной целевой СУБД.

Концептуальное и логическое проектирование вместе называют также *инфологическим* или *семантическим проектированием*.

В настоящее время для проектирования БД активно используются CASE-средства, в основном ориентированные на использование *ERD (Entity – Relationship Diagrams, диаграммы «сущность–связь»)*. С их помощью определяются важные для предметной области объекты (сущности), отношения друг с другом (связи) и их свойства (атрибуты). Следует отметить, что средства проектирования ERD в основном ориентированы на реляционные базы данных (РБД), и если существует необходимость проектирования другой системы, скажем объектно-ориентированной, то лучше избрать другие методы проектирования.

2.2. Концептуальное проектирование с использованием методологии IDEF1X

Цель концептуального проектирования – создание концептуальной модели данных на основе представлений о предметной области каждого отдельного типа пользователей. *Концептуальная модель* представляет собой описание основных сущностей (таблиц) и связей между ними без учета принятой модели БД и синтаксиса целевой СУБД. Часто на такой модели отображаются только имена сущностей (таблиц) без указания их атрибутов. *Представление пользователя* включает в себя данные, необходимые конкретному пользователю для принятия решений или выполнения некоторого задания.

Ниже рассматривается последовательность шагов при концептуальном проектировании. *Выделение сущностей*.

Первый шаг в построении концептуальной модели данных состоит в определении основных объектов (сущностей), которые могут интересовать пользователя и, следовательно, должны храниться в БД. При наличии функциональной модели IDEF0 прообразами таких объектов являются входы, управления и выходы. Еще лучше для этих целей использовать DFD. Прообразами объектов в этом случае будут накопители данных. Как было отмечено выше, накопитель данных является совокупностью таблиц (набором объектов) или непосредственно таблицей (объектом). Для более детального определения набора основных объектов необходимо также проанализировать потоки данных и весь методический материал, требуемый для решения задачи. Например, для задачи определения допускаемых скоростей основными объектами (наборами объектов) являются: нормативно-справочная информация, информация об участках дороги, задания на расчет, ведомости допускаемых скоростей и т. д. В ходе анализа и проектирования информационной модели наборы объектов должны быть детализированы.

3. Определение связей.

Наиболее характерными типами связей между сущностями являются:

- связи типа «часть–целое», определяемые обычно глаголами «состоит из», «включает» и т.п.;
- классифицирующие связи (например, «тип – подтип», «множество – элемент», «общее – частное» и т. п.);
- производственные связи (например, «начальник–подчиненный»);
- функциональные связи, определяемые обычно глаголами «производит», «влияет», «зависит от», «вычисляется по» и т. п.

Среди них выделяются только те связи, которые необходимы для удовлетворения требований к разработке БД.

Связь характеризуется следующим набором параметров:

- именем – указывается в виде глагола и определяет семантику (смысловую подоплеку) связи;
- кратностью (кардинальность, мощность): один-к-одному (1:1), один-ко-многим (1:N) и многие-ко-многим (N:M, $N = M$ или $N <> M$). Кратность показывает, какое количество экземпляров одной сущности определяется экземпляром другой. Например, на одном участке (описывается строкой таблицы «Участки») может быть один, два и более путей (каждый путь описывается отдельной строкой в таблице «Пути»). В данном случае связь 1:N. Другой пример: один путь проходит через несколько отдельных пунктов и через один отдельный пункт может проходить несколько путей – связь N:M;

- типом: идентифицирующая (атрибуты одной сущности, называемые внешним ключом, входят в состав дочерней и служат для идентификации ее экземпляров, т.е. входят в ее первичный ключ) и неидентифицирующая (внешний ключ имеется в дочерней сущности, но не входит в состав первичного ключа);

- обязательностью: обязательная (при вводе нового экземпляра в дочернюю сущность заполнение атрибутов внешнего ключа обязательно и для введенных значений должен существовать экземпляр в родительской сущности) и необязательная (заполнение атрибутов внешнего ключа в экземпляре дочерней сущности необязательно или введенным значениям не соответствует экземпляр в родительской сущности);

- степенью участия – количеством сущностей, участвующих в связи. В основном между сущностями существуют бинарные связи, т. е. ассоциации, связывающие две сущности (степень участия равна 2). Например, «Участок» состоит из «Путей». В то же время по степени участия возможны следующие типы связей:

- о унарная (рекурсивная) – сущность может быть связана сама с собой. Например, в таблице «Работники» могут быть записи и по подчиненным, и по их начальникам. Тогда возможна связь «начальник» – «подчиненный», определенная на одной таблице;

- о тернарная – связывает три сущности. Например, «Студент» на «Сессии» получил «Оценку по дисциплине»;

- о кватернарная и т.д.

В методологии IDEF1X степень участия может быть только унарной или бинарной. Связи большей степени приводятся к бинарному виду.

Внешний вид связи на диаграммах IDEF1X указывает на ее мощность, тип и обязательность (табл. 7.1).

Примечания:

1. Идентифицирующая связь отображается сплошной линией, неидентифицирующая – пунктирной.

2. Необязательность обозначается ромбиком.

4. *Определение суперклассов и подклассов.*

В тех случаях, когда две и более сущностей по набору атрибутов незначительно отличаются друг от друга, можно применять в модели конструкцию – иерархию наследования (категорий), включающую в себя суперклассы и подклассы.

Суперкласс – сущность, включающая в себя подклассы.

Иерархия наследования представляет собой особый тип объединения сущностей, которые разделяют общие характеристики. Например, в организации работают служащие, занятые полный рабочий день (постоянные служащие) и совместители. Из их общих свойств можно сформировать обобщенную сущность (родового предка) «Сотрудник» (рис. 7.6), чтобы представить информацию, общую для всех типов служащих. Специфическая для каждого типа информация может быть расположена в дополнительных сущностях (потомках) «Постоянный сотрудник» и «Совместитель».

Обычно иерархию наследования создают, когда несколько сущностей имеют общие по смыслу атрибуты, либо когда сущности имеют общие по смыслу связи (например, если бы «Постоянный сотрудник» и «Совместитель» имели бы сходную по смыслу связь «работает в» с сущностью «Организация»).

Для каждой категории требуется указать *дискриминатор* – атрибут родового предка, который показывает, как отличить одну сущность от другой. В приведенном примере дискриминатор – атрибут «Тип».

2.3. Логическое проектирование с использованием методологии IDEF1X

Цель логического проектирования – развить концептуальное представление БД с учетом принимаемой модели БД (иерархической, сетевой, реляционной и т. д.).

Примем в качестве модели реляционную БД в третьей нормальной форме (набор нормализованных отношений с кратностью связей 1:N). Поэтому необходимо будет проверить концептуальную модель с помощью методов нормализации и контроля выполнения транзакций [1, 20, 21]. *Транзакция* – одно действие или их последовательность, выполняемых как единое целое одним или несколькими пользователями (прикладными программами) с целью осуществления доступа к БД и изменению ее содержимого.

1. *Удаление и проверка элементов, не отвечающих принятой модели данных.*

1.1. *Удаление связей N:M.*

Если в концептуальной модели присутствуют связи N:M, то их следует устранить путем определения промежуточной сущности. Связь N:M заменяется двумя связями типа 1:M, устанавливаемыми со вновь созданной сущностью

1.2. *Удаление связей с атрибутами.*

Связи с атрибутами должны быть преобразованы в сущности.

В разработанной концептуальной модели существовала связь N:M («Раздельные пункты» : «Пути»), которая имела собственные атрибуты. После устранения связи N:M, ее атрибуты перешли в сущность «Раздельные пункты на пути» (см. рис. 7.9, неключевые атрибуты).

В графической нотации IDEF1X не предусмотрено отображение связей с атрибутами, хотя некоторые методологии ERD допускают их наличие и отображение.

1.3. *Удаление сложных связей (со степенью участия более 2).*

Сложную связь заменяют необходимым количеством бинарных связей 1:N со вновь созданной сущностью, которая и показывает эту связь.

1.4. *Удаление рекурсивных связей (со степенью участия 1).*

Рекурсивную связь заменяют, определив дополнительную сущность и необходимое количество связей (рис. 7.10). В данной схеме можно принять следующее правило: если в сущности «Сотрудник» атрибуты «Табельный номер» и «Табельный номер руков.» совпадают, то набор атрибутов в сущности «Сотрудник» характеризует руководителя.

1.5. *Удаление многозначных атрибутов (атрибутов имеющих несколько значений).*

Многозначность устраняется путем введения новой сущности и связи 1:

Рис. 7.11. Удаление многозначных атрибутов

Данное преобразование, помимо соответствия реляционной модели данных, также позволяет хранить любое количество телефонов по одному филиалу.

1.6. Удаление избыточных связей.

Связь является избыточной, если одна и та же информация может быть получена не только через нее, но и с помощью другой связи

В приведенном примере одну из связей «Руководит» можно смело удалить (лучше между «Руководителем филиала» и «Сотрудником»).

1.7. Перепроверка связей 1:1.

В процессе определения сущностей могли быть созданы сущности, которые на самом деле являются одной. В этом случае их следует объединить. Например, в приведенном выше примере (рис. 46) сущности «Филиал» и «Руководитель филиала» лучше объединить.

В то же время не всегда можно выполнить такое

Офисный пакет состоит из строго определенного набора компонентов, причем каждый из них характеризуется большим количеством атрибутов. Кроме этого, некоторые компоненты могут отсутствовать в офисном пакете (например, почтовый клиент) или они не входят в его состав (выступают в качестве самостоятельного продукта). В описанных случаях рекомендуется не объединять сущности.

2. Проверка модели с помощью правил нормализации.

Основная идея нормализации заключается в том, чтобы каждый факт хранился в одном месте, т. е. чтобы не было дублирования данных. Многие из требований нормализации, как правило, уже учитываются при выполнении предыдущих шагов проектирования.

Пример построения логической модели

На рис. 7.19 также показаны триггеры на действия, выполняемые как со стороны родительской сущности, так и со стороны дочерней. Триггеры показаны в следующем формате «Действие : Тип триггера». Действие может быть одного из трех типов: D (DELETE), I (INSERT) и U (UPDATE). Тип триггера обозначается: C (CASCADE) и R (RESTRICT).

Логическая модель построена с использованием ERwin 4.0.

2.4. Физическое проектирование с использованием методологии IDEF1X

Цель физического проектирования – преобразование логической модели с учетом синтаксиса, семантики и возможностей выбранной целевой СУБД.

В связи с тем, что методология физического проектирования существенно зависит от выбранной целевой СУБД, ограничимся лишь общими рекомендациями [20, 21].

1. Анализ необходимости введения контролируемой избыточности.

При реализации проекта часто для достижения большей эффективности системы требуется снизить требования к уровню нормализации отношений, т. е. внести некоторую избыточность данных. Процесс внесения таких изменений в БД называется *денормализацией*.

Раздел 3. Разработка поведенческой модели

3.1. Основы построения поведенческих моделей

После определения функций системы и разработки информационной основы следующим шагом является проектирование поведения системы. Поведенческая модель системы показывает, за счет чего достигается требуемая функциональность и какие данные используются для ее обеспечения. Таким образом, поведенческая модель напрямую базируется на функциональной и информационной моделях системы.

Для построения поведенческой модели обычно используются блок-схемы алгоритмов. Как правило, их строят для функций (процессов), показываемых на последних уровнях диаграмм декомпозиции IDEF0 и DFD. Для DFD блок-схемы алгоритмов являются более удачным решением, чем описание элементарных процессов в виде миниспецификаций (см. подразд. 6.11). В связи с этим далее рассматриваются основы построения блок-схем для моделирования поведения системы.

3.2. Блок-схемы алгоритмов

При описании алгоритмов давно и успешно используются блок-схемы (Basic Flowchart). Построение блок-схем алгоритмов регламентируется ГОСТ 19.701-90 (ИСО 5807-85) «Единая система программной документации. Схемы алгоритмов программ, данных и систем. Условные обозначения и правила выполнения» [22]. Данный государственный стандарт составлен на основе международного стандарта «ISO 5807-85. Information processing – Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts».

Согласно ГОСТ 19.701-90 под *схемой* понимается графическое представление определения, анализа или метода решения задачи. С помощью схем можно отобразить как статические, так и динамические аспекты системы. Символы, приведенные в государственном стандарте, могут использоваться в следующих *типах схем*:

- схемы данных – определяют последовательность обработки данных и их носители;
- схемы программ – отображают последовательность операций в программе (по сути, это и есть блок-схемы алгоритмов в традиционном понимании);
- схемы работы системы – отображают управление операциями и потоки данных в системе;
- схемы взаимодействия программ – отображают путь активации программ (модулей) и их взаимодействие с соответствующими данными;
- схемы ресурсов системы – отображают конфигурацию блоков данных и обрабатывающих блоков.

При построении поведенческой модели системы используются основные принципы структурного подхода – принципы декомпозиции и иерархического упорядочения. Поведенческая модель представляет собой набор взаимосвязанных схем (диаграмм) с разным уровнем детализации, причем с каждым новым уровнем детализации система приобретает все более законченные очертания.

На схемах могут присутствовать следующие *элементы графической нотации*:

- символы данных – указывают на наличие данных, вид носителя или способ ввода-вывода данных;
- символы процесса – указывают операции, которые следует выполнить над данными;
- символы линий – указывают потоки данных между процессами и/или носителями данных, а также потоки управления между процессами;
- специальные символы – используются для облегчения написания и чтения схем.

3.3. Правила и рекомендации построения блок-схем

Ниже приводятся некоторые правила и рекомендации построения схем [22].

1. Допускается зеркально отображать символы и поворачивать их вокруг оси. В частности, запоминающие устройства с прямым доступом (таблицы на жестком диске) на схемах, как правило, поворачивают на 90° против часовой стрелки.
2. Большинство символов допускают задание внутри них текстовых пояснений. Если текст не помещается внутри символа, то лучше его приводить, используя комментарии.
3. Количество пересечений линий следует минимизировать. При этом считается, что пересекающиеся линии не имеют логической связи друг с другом. Другими словами потоки данных или управления в местах пересечений не меняют своего направления.
4. Если две или более линий объединятся в одну, то место объединения должно быть
 5. Несколько выходов из символа решения следует показывать одним из следующих способов:
 - несколькими линиями от данного символа к другим символам;
 - одной линией от данного символа, которая затем разветвляется в соответствующее число
 В случае ветвления каждый выход из символа должен сопровождаться либо записью условия (например, условие «Сравнить А и В», 3 выхода: $A > B$, $A < B$ и $A = B$), либо результата проверки условия, записанного внутри символа (например, условие « $A = B$ », 2 выхода: Да и Нет).
 Как видно, на рис. 8.2 символ решения допускает более двух исходящих потоков.
 6. Вместо одного символа с соответствующим текстом могут быть использованы несколько символов с перекрытием изображения, каждый из которых может содержать дополнительный текст (например, запись или посылка нескольким получателям, подготовка нескольких копий документа и т. д.
 7. Если направление стрелки не указано, то направление потока считается сверху вниз, слева направо.

4.3. Лабораторные работы

Учебным планом не предусмотрено

4.4. Практические занятия

<i>№ п/п</i>	<i>Номер раздела дисципли ны</i>	<i>Наименование тем практических занятий</i>	<i>Объем (час.)</i>	<i>Вид занятия в интерактивн ой, активной, инновационно й формах, (час.)</i>
1	1.	Разработка функциональной модели (методология IDEF0)	8	-
2	1.	Разработка функциональной модели (методология DFD)	6	6 час. (работа в малых группах)
3	2.	Проектирование реляционной базы данных как компонента АИС на базе современных CASE-средств.	12	-
4	2.	Объектно-ориентированное проектирование пользовательского интерфейса	6	-
5	3.	Разработка блок-схем	4	-
ИТОГО			36	6

4.5. Контрольная работа

Цель: формирование знаний о современных технологиях анализа и проектирования информационных систем, приобретение навыков работы с учебной, научной и справочной литературой, и закрепление практических знаний по дисциплине

Контрольная работа выполняется в виде пояснительной записки объемом 20-25 страниц, оформляется в строгом соответствии со стандартом ФГБОУ ВО «БрГУ».

Структура работы:

- титульный лист;
- содержание;
- введение;
- основные разделы работы;
- заключение;
- список использованных источников;
- приложения.

Основная тематика: контрольная работа выполняется по варианту, выдаваемому преподавателем.

Выдача задания, прием контрольной работы проводится в соответствии с календарным учебным графиком.

5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

<i>Компетенции</i> <i>№, наименование разделов дисциплины</i>	<i>Кол-во часов</i>	<i>Компетенции</i>			Σ <i>ком п.</i>	<i>t_{ср}, час</i>	<i>Вид учебной работы</i>	<i>Оценка результатов</i>
		<i>ПК</i>						
		<i>1</i>	<i>11</i>	<i>13</i>				
1	2	3	4	5	6	7	8	9
1. Основы анализа и проектирования информационных систем	48	+	+	+	3	16	Лекция, ПЗ, кр, СРС	экзамен
2. Разработка информационной модели	40	-	+	-	1	40	Лекция, ПЗ, кр, СРС	экзамен
3. Разработка поведенческой модели	29	+	-	-	1	29	Лекция, ПЗ, кр,СРС	экзамен
<i>всего часов</i>	117	45	56	16	3	39		

6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

1. Заботина Н.Н. Проектирование информационных систем : учебное пособие / Н. Н. Заботина. – М. : ИНФРА-М, 2013. – 331 с. – (Высшее образование: Бакалавриат).
2. Шелухин О. И. Моделирование информационных систем : учебное пособие / О. И. Шелухин. – 2-е изд., перераб. И доп. – Москва : Горячая линия- Телеком, 2011. – 536 с.
3. Заботина Н. Н. Проектирование информационных систем. В 2 ч. Ч. 1-2 / Н. Н. Заботина. – Братск : БрГУ, 2005. Ч.1 : Методология функционального моделирования. – 146 с.

7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№	Наименование издания (автор, заглавие, выходные данные)	Вид занятия (Лк, ПЗ, КР, СР)	Количество экземпляров в библиотеке, шт.	Обеспеченность, (экз./чел.)
1	2	3	4	5
1. Основная литература				
1	Балдин, К.В. Информационные системы в экономике : учебник / К.В. Балдин, В.Б. Уткин. – 7-е изд. – Москва : Издательско-торговая корпорация «Дашков и К°», 2017. – 395 с. : ил. – Библиогр. В кн. – ISBN 978-5-394-01449-9 ; То же [Электронный ресурс]. – URL: http://biblioclub.ru/index.php?page=book&id=454036	Лк, ПЗ, СР	1(ЭР)	1
2	Белов В.В. Проектирование информационных систем : учебник / В. В. Белов, В. И. Чистякова. – Москва : Академия, 2013. – 352 с. – (Бакалавриат). -ISBN 978-5-	Лк, ПЗ, СР	15	1
2. Дополнительная литература				
3	Грекул, В. И. Проектирование информационных систем : учеб. Пособие для вузов / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – 2-е изд., испр. – Москва : ИНТУИТ.РУ, 2007. – 326 с.	Лк, ПЗ, СР	50	1
4	Вендров, А. М. Практикум по проектированию программного обеспечения экономических информационных систем : учебное пособие / А. М. Вендров. – 2-е изд., перераб. И доп. – Москва : Финансы и статистика, 2006. – 192 с.	Лк, ПЗ, СР	28	1

8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронный каталог библиотеки БрГУ: http://irbis.brstu.ru/CGI/irbis64r_15/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=
2. Электронная библиотека БрГУ <http://ecat.brstu.ru/catalog>
3. Федеральная университетская компьютерная сеть России // Электронный ресурс

[Режим доступа: свободный] <http://www.runnet.ru/>

4. Каталог учебников, оборудования, электронных ресурсов // Электронный ресурс [Режим доступа: свободный] <http://ndce.edu.ru/>

5. Электронно-библиотечная система издательства «Лань» // Электронный ресурс <http://e.lanbook.com/> ,

6. Библиотека «Книгосайт» // Электронный ресурс [Режим доступа: свободный] <http://knigosite.ru/>

7. Электронная библиотека книг на тему бизнеса, финансов, экономики и смежным темам // Электронный ресурс [Режим доступа: свободный] <http://www.finbook.biz/>

8. ЭБС «Университетская библиотека online» // Электронный ресурс <http://biblioclub.ru/> ,

9. Научная электронная библиотека «КИБЕРЛЕНИНКА» // Электронный ресурс [Режим доступа: свободный] <http://cyberleninka.ru/>

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Вид учебных занятий	Организация деятельности обучающихся
Лекции	Написание конспекта лекций: кратко, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверка терминов с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, практическом занятии.
Практические занятия	Развитие интеллектуальных умений, подготовка ответов к контрольным вопросам, работа с основной и дополнительной литературой, необходимой для освоения дисциплины, выполнение заданий, активное участие в интерактивной, активной, инновационной формах обучения, составление и оформление отчетов по практическим заданиям.
Контрольная работа	Работа с основной и дополнительной литературой, необходимой для выполнения контрольной работы, углубление и конкретизация необходимого в соответствии с темой материала из литературных источников и полученных теоретических знаний, выработка способности и готовности их использования в практической исследовательской работе. Развитие интеллектуальных умений изложения материала, представления с элементами визуализации (схемы, графики, рисунки, таблицы, формулы) и оформления в соответствии с требованиями ГОСТ.
Самостоятельная работа обучающихся	<i>Подготовка к практическим занятиям.</i> Проработка основной и дополнительной литературы, терминов, сведений, требующихся для запоминания и являющихся основополагающими в теме/разделе. Конспектирование прочитанных литературных источников. Проработка материалов по изучаемому вопросу, с использованием на рекомендуемых ресурсах информационно-телекоммуникационной сети «Интернет». Выполнение заданий преподавателя, необходимых для подготовки к участию в интерактивной, активной, инновационных формах обучения по изучаемой теме. <i>Подготовка к экзамену.</i> При подготовке к экзамену необходимо ориентироваться на конспекты лекций, рекомендуемую литературу, использовать рекомендуемые ресурсы информационно-телекоммуникационной сети «Интернет».

9.1. Методические указания для обучающихся по выполнению практических работ

Практическое занятие № 1. Разработка функциональной модели (методология IDEF0)

Цель практического занятия. Освоение CASE-средства BPwin в целях разработки функциональной модели информационной системы с использованием методологии IDEF0.

Задание на выполнение практической работы

1) Изучить и закрепить основы разработки функциональных моделей с использованием методологии IDEF0 (Лекция. Разработка функциональной модели).

2) Освоить CASE-средство BPwin в части разработки функциональных моделей с использованием методологии IDEF0.

3) Построить функциональную модель информационной системы по индивидуальному заданию.

4) Оформить и защитить отчет. В отчете должны быть приведены:

- контекстная диаграмма;
- диаграмма декомпозиции 1-го уровня;
- две диаграммы декомпозиции 2-го уровня для двух наиболее интересных блоков с диаграммы декомпозиции 1-го уровня;
- диаграмма дерева узлов.

Назначение BPwin. CASE-средство BPwin предназначено для построения функциональных моделей с использованием методологий:

- IDEF0 - функциональные модели любых систем;
- IDEF3 - функциональные модели технологических процессов;
- DFD - функциональные модели информационных систем.

Навигатор панели процессов предназначен для отображения и выбора диаграмм разрабатываемой функциональной модели.

Рабочая область предназначена для отображения и редактирования диаграммы модели, выбранной в панели процессов.

Создание новой модели

Для создания новой модели необходимо выбрать пункт меню "File / New" или нажать на соответствующую кнопку стандартной панели инструментов (см.рис.2). На экране появится диалоговое окно

В диалоговом окне необходимо выбрать радиокнопку "Create model", ввести имя модели в поле "Name" и выбрать методологию, нотация которой будет использоваться при построении модели (радиокнопки "Type").

Для указания общих параметров модели необходимо выбрать пункт меню "Model / Model Properties" и в появившемся диалоговом окне перейти на вкладку "General"

На вкладке задаются следующие параметры модели:

- имя модели (Model name);
- имя проекта (Project). Имя проекта, как правило, совпадает с именем разрабатываемой информационной системы;
- фамилия автора или наименование компании (Author);
- инициалы автора (Author initials);
- тип модели - AS-IS (как есть) или TO-BE (как будет). Подробнее см. раздел "Основы функционального анализа и проектирования систем".

Создание и редактирование диаграмм

После нажатия на кнопку "Ok" диалогового окна создания модели автоматически создается контекстная диаграмма. Указание параметров диаграммы, выбранной в текущий момент в панели процессов, осуществляется через диалоговое окно "Diagram Property", вызываемого через пункт меню "Diagram / Diagram Property"

На вкладке "Status" указываются статус, дата создания и дата последней редакции диаграммы

Типы статуса диаграммы имеют следующий смысл:

- рабочая (WORKING) – диаграмма находится в стадии разработки;
- черновик (DRAFT) – диаграмма прошла некоторые стадии рассмотрения с заказчиками, но это не окончательный вариант;
- рекомендованная (RECOMMENDED) – диаграмма прошла все стадии рассмотрения с заказчиками и отвечает формальным требованиям, но это не окончательный вариант;
- готовая или публикуемая (PUBLICATION) – окончательный вариант диаграммы.

На вкладке "Page Setup" указываются единицы измерения (Units), формат листов (Sheet Size), поля, необходимость отображения заголовка (Header) и нижнего колонтитула (Footer)

На вкладке "Header/Footer" возможно задание пользовательского (custom) вида заголовка (Header) и нижнего колонтитула (Footer) диаграммы

Для непосредственного создания элементов диаграммы и ускоренной навигации по модели используется панель инструментов "BPwin Toolbox" (отображение или скрытие панели выполняется через пункт меню "View"). На рис.9 приведено назначение элементов управления панель инструментов "BPwin Toolbox".

Для указания параметров функции необходимо щелкнуть по ней правой кнопкой мыши и в контекстном меню выбрать соответствующий пункт. В результате на экране появится диалоговое окно "Activity Properties"

На вкладке диалогового окна можно задать:

- имя блока (вкладка "Name");

- комментарий к блоку (вкладка "Definition");
- параметры шрифта надписи блока (вкладка "Font");
- цвет блока (вкладка "Color");
- графический примитив, используемый для отображения блока (вкладка "Box style").

Для указания аналогичных параметров стрелки используется диалоговое окно "Arrow Properties"

Если наименование стрелки расположено удаленно от самой стрелки или возникают трудности по сопоставлению наименования стрелки с самой стрелкой (в случае высокого насыщения диаграммы элементами) можно на диаграмме отобразить ассоциацию между ними. Для этого необходимо щелкнуть по стрелке правой кнопкой мыши и в контекстном меню выбрать пункт "Squiggle".

Для указания на диаграмме произвольного комментария непосредственно к элементу используются кнопки "Задание ассоциации" и "Добавление произвольного текста".

Для навигации по модели (переходу к диаграммам) используются последние четыре кнопки панели "BPwin Toolbox".

Если на диаграмме выбран блок, для которого не существует диаграммы декомпозиции, и нажата кнопка в панели инструментов ▼, то на экране появится диалоговое окно "Activity Box Count"

В этом диалоговом окне требуется выбрать методологию, в соответствии с которой будет строиться диаграмма декомпозиции, и предполагаемое количество блоков на диаграмме. BPwin создаст диаграмму с указанным количеством блоков и перенесет на нее все стрелки входящие и выходящие в родительский блок.

- перемещение блока или стрелки на диаграмме - выбрать в панели инструментов "BPwin Toolbox" кнопку ↑, подвести указатель мыши на диаграмме к соответствующему элементу диаграммы, нажать левую кнопку мыши и, не отпуская ее, задать новое положение элемента.

- создание диаграммы дерева узлов - выбрать в панели процессов или на диаграмме блок (корень дерева), начиная с которого будет строиться диаграмма дерева узлов, выбрать пункт меню "Diagram / Add Node Tree" и в появившемся диалоговом окне задать имя диаграммы дерева узлов и количество уровней дерева.

Контрольные вопросы для самопроверки

1. Case-технологии анализа и проектирования ИС. Назначение и основные возможности Case-средств.
2. Модели ИС.
3. Принципы построения моделей.
4. Сущность структурного подхода к анализу и проектированию ИС.
5. Краткая характеристика методологий структурного анализа и проектирования ИС.
6. Методология IDEF0. Назначение.
7. Методология IDEF0. Виды диаграмм.
8. Методология IDEF0. Элементы диаграмм.
9. Методология IDEF0. ICOM-коды.
10. Методология IDEF0. Типы связей между работами.
11. Методология IDEF0. Модели AS-IS, TO-BE и SHOULD-BE.
12. Методология IDEF0. Правила и рекомендации построения диаграмм.

Основная литература

1.-2. из раздела 7.

Дополнительная литература

3.-4. из раздела 7.

Практическое занятие № 2. . Разработка функциональной модели (методология DFD)

Цель практического занятия: Освоение CASE-средства BPwin в целях разработки функциональной модели информационной системы с использованием методологии DFD.

Назначение, общие сведения об интерфейсе, создание новой модели и основы создания и редактирования диаграмм рассмотрены в предыдущем занятии (№ 1. Разработка функциональной модели (методология IDEF0)).

Ниже рассматриваются особенности разработки функциональной модели с использованием методологии DFD:

Панель инструментов "BPwin Toolbox" при редактировании диаграмм в нотации DFD имеет следующий

Для указания параметров процесса, потока данных, внешней сущности и накопителя данных используются диалоговые окно, аналогичные окнам "Activity Properties" и "Arrow Properties".

Задание особых видов потоков данных (двунаправленного, квазинепрерывного и управляющего) осуществляется в разделе "Type" на вкладке "Style" диалогового окна "Arrow Properties"

Задание на выполнение практической работы

1) Изучить и закрепить основы разработки функциональных моделей с использованием методологии DFD (материалы лекции 6. Разработка функциональной модели).

2) Освоить CASE-средство BPwin в части разработки функциональных моделей с использованием методологии DFD.

3) Построить функциональную модель информационной системы по индивидуальному заданию.

4) Оформить и защитить отчет. В отчете должны быть приведены:

- контекстная диаграмма;

- диаграмма декомпозиции 1-го уровня;

- две диаграммы декомпозиции 2-го уровня для двух наиболее интересных блоков с диаграммы декомпозиции 1-го уровня;

- диаграмма дерева узлов.

На диаграммах должны быть показаны не менее 3-4 внешних сущностей и 4-5 накопителей данных.

Контрольные вопросы для самопроверки

1. Диаграммы потоков данных. Назначение.

2. Диаграммы потоков данных. Элементы диаграмм.

3. Диаграммы потоков данных. Правила и рекомендации построения диаграмм.

Миниспецификации.

4. Диаграммы потоков данных. Расширение DFD для систем реального времени.

5. ERD. Назначение и основные элементы моделей.

6. Возможности современных CASE-средств моделирования данных.

Основная литература

1.-2. из раздела 7.

Дополнительная литература

3.-4. из раздела 7.

Практическое занятие № 3. Проектирование реляционной базы данных как компонента АИС на базе современных CASE-средств (работа в малых группах)

Задание на выполнение практической работы

1) Изучить и закрепить основы разработки информационных моделей с использованием методологии IDEF1X (материалы лекции 7. Разработка информационной модели).

2) Освоить CASE-средство ERwin в части разработки информационных моделей с использованием методологии IDEF1X.

3) Построить информационную модель системы по индивидуальному заданию.

4) Оформить и защитить отчет. В отчете должны быть приведены:

- концептуальная модель БД;

- логическая модель БД;

- физическая модель БД, включая типы данных для атрибутов и триггеры;

- DDL-скрипт генерации структуры БД.

Цель практического занятия

Освоение CASE-средства ERwin в целях разработки информационной модели с использованием методологии IDEF1X.

Назначение ERwin

CASE-средство ERwin предназначено для разработки информационных моделей с использованием методологий IDEF1X и IE. В ERwin реализованы основные функции, характерные для классических CASE-средств:

- прямое проектирование от создания концептуальной или логической модели БД до генерации **структуры БД** на диске или DDL-скрипта;

- обратное проектирование (реинжиниринг) создания физической модели БД на основе БД на диске или DDL-скрипта;

- синхронизацию моделей БД с самой БД на диске.

В качестве несомненных достоинств Erwin следует отметить:

- поддержку около 20 промышленных СУБД (ORACLE, Informix, DB2, MS SQL Server и др.) и 5 популярных настольных СУБД (Access, Foxpro, Paradox и др.);

- наличие функции проверки моделей БД требованиям полноты, целостности и нормализации.

Общие сведения об интерфейсе ERwin

Навигатор панели процессов предназначен для отображения и выбора общих доменов (Domains), правил проверки данных (Validation Rules), значений по умолчанию (Default Values), сущностей (Entities) и др., используемых при построении информационной модели.

Рабочая область предназначена для отображения и редактирования диаграммы модели.

На рис.2 приведено назначение элементов управления стандартной панели инструментов (Standard Toolbar).

С помощью режимов отображения можно задать степень детализации отображения сущностей на диаграммах. Список для выбора типа модели позволяет выполнить переключение между логической и физической моделями БД. Режим отображения "Entity" с типом "Logical" соответствует концептуальной модели БД (материалы лекции 7. Разработка информационной модели), режим "Attribute" с типом "Logical" - логической модели и режим "Attribute" с типом "Physical" - физической модели.

Создание новой модели

Для создания новой модели необходимо выбрать пункт меню "File / New" или нажать на соответствующую кнопку стандартной панели инструментов (см.рис.2). На экране появится диалоговое окно (

В диалоговом окне необходимо выбрать тип модели "New Model Type" (логическая, физическая или логическая/физическая), а также выбрать целевую СУБД "Database" и ее версию "Version".

Для указания общих параметры модели необходимо выбрать пункт меню "Model / Model Properties" и в появившемся диалоговом окне перейти на вкладку "General"

В разделе "Auto-Transform Logical Objects" имеется возможность указание следующих действий при переходе от логической модели к физической:

- преобразование связи многие-ко-многим в связующую таблицу, соединенную с исходными связями один-ко-многим (Many-to-Many Relationships with Association Table);
- автоматическое создание триггеров для суперклассов и подклассов (Supertype/Subtype with Identifying Relationships).

На вкладке "Notation" можно изменить нотацию, в соответствии с которой отображаются диаграммы

Создание и редактирование диаграмм

Перед созданием диаграммы необходимо в списке панели инструментов "Standard" выбрать ее тип: логическая или физическая.

Для создания элементов диаграммы используется панель инструментов "Toolbox" (отображение или скрытие панели выполняется через пункт меню "View"). На рис.6 приведено назначение элементов управления панель инструментов "Toolbox".

Для работы с сущностями предназначено контекстное меню, вызываемое при нажатии правой кнопки мыши на соответствующей сущности (рис.7).

Для указания общих параметров сущности используется диалоговое окно "Entities" (рис.8), вызываемое через пункт "Entity Properties" контекстного меню.

В диалоговом окне задаются имя и комментарий к сущности. В случае установки галочки напротив пункта "Logical Only" сущность будет отображаться только в логической модели.

Для указания набора атрибутов сущности и их параметров используется диалоговое окно "Attributies" (рис.9), вызываемое через пункт "Attributies" контекстного меню.

В левой части окна отображается список имеющихся атрибутов сущности с возможностями:

- добавления нового атрибута - кнопка "New";
- переименования выделенного атрибута - кнопка "Rename";
- удаления выделенного атрибута - кнопка "Delete";
- изменения положения выделенного атрибута в списке - кнопки "▲" и "▼".

При задании нового атрибута посредством нажатия кнопки "New" требуется указать его тип, имя атрибута в логической модели (Attribute Name) и имя атрибута в физической модели (Column Name)

В правой части диалогового окна "Attributies" для выделенного атрибута задается один из стандартных типов данных (на вкладке "General") и указывается признак вхождения атрибута в первичный ключ (Primary Key).

На вкладке "Datatype" диалогового окна "Attributies" задаются дополнительные параметры выделенного атрибута

На вкладке можно задать более конкретный тип данных атрибута (Datatype), правило проверки данных при изменении значения атрибута (Valid) и значение по умолчанию (Default).

Для создания связи между сущностями необходимо в панели инструментов "Standard" нажать на кнопку соответствующей связи, щелкнуть левой кнопкой мыши вначале по родительской, а затем по дочерней сущности. Если в родительской сущности имеются атрибуты, входящие в первичный ключ, то эти атрибуты будут добавлены в дочернюю сущность в качестве внешнего ключа. При этом, если связь идентифицирующая, то добавленные атрибуты автоматически будут входить в состав первичного ключа дочерней сущности.

Для работы со связями предназначено контекстное меню, вызываемое при нажатии правой кнопки мыши на соответствующей

Особенности работы с физической моделью

При работе с физической моделью в меню программы появляется пункт меню "Database". При выборе пунктов этого меню появляется возможность задания триггеров (Triggers), хранимых процедур (Stored Procedures) и SQL-скриптов (Pre & Post Scripts), выполняемых до и/или после создания БД или отдельной сущности. Триггеры, процедуры и скрипты могут быть заданы на уровне конкретных команд SQL.

Доступ к этим возможностям возможен также, но только на уровне сущности, через контекстное меню сущности, имеющее измененный вид

При выборе пункта "Columns" контекстного меню вместо вкладки "Datatype" появляется вкладка, соответствующая целевой СУБД (

На этой вкладке список типов данных ограничен набором типов, поддерживаемых целевой СУБД, и имеется возможность указание обязательности заполнения атрибута (NOT NULL).

В случае необходимости с помощью диалогового окна "Target Server", вызываемого через пункт меню "Database / Choose Database", можно сменить целевую СУБД физической модели

Генерация БД или DDL-скрипта

Для генерации БД или DDL-скрипта (при прямом проектировании) необходимо выбрать пункт меню "Tools / Forward Engineer". На экране появиться диалоговое окно "Schema Generation

В этом окне для компонентов БД (схем, таблиц, полей, индексов и т.д.) можно задать генерируемые элементы.

Для генерации БД на диске необходимо нажать кнопку "Generate", в появившемся диалоговом окне выбрать имя БД, задать имя пользователя и пароль, после чего нажать на кнопку "Ok".

Для генерации и просмотра DDL-скрипта необходимо нажать кнопку "Preview". На рис.19 приведено диалоговое окно, содержащее сгенерированный DDL-скрипт.

Для обратного проектирование (реинжиниринга) предназначен пункт меню "Tools / Reverse Engineer".

Для синхронизации модели и БД предназначен пункт меню "Tools / Complete Compare".

Контрольные вопросы для самопроверки

1. Методология IDEF1X. Элементы диаграмм.
2. Концептуальное проектирование БД. Стадии.
3. Концептуальное проектирование БД. Сущности.
4. Концептуальное проектирование БД. Связи.
5. Концептуальное проектирование БД. Атрибуты.
6. Концептуальное проектирование БД. Ключи.
7. Концептуальное проектирование БД. Суперклассы и подклассы.
8. Логическое проектирование БД. Стадии.
9. Логическое проектирование БД. Удаление элементов, не отвечающих реляционной модели данных.
10. Логическое проектирование БД. Нормализация.
11. Логическое проектирование БД. Определение требований поддержки целостности данных.
12. Физическое проектирование БД. Стадии.
13. Физическое проектирование БД. Денормализация.
14. Физическое проектирование БД. Разработка механизмов защиты.

Основная литература

1.-2. из раздела 7.

Дополнительная литература

3.-4. из раздела 7.

Практическое занятие № 4. Объектно-ориентированное проектирование пользовательского интерфейса

Цель— изучение структуры и инструментов человекоориентированного подхода к разработке интерфейсов, основных принципов и паттернов разработки пользовательских интерфейсов, формирование навыков анализа, подбора и применения необходимого инструментария для решения поставленных интерфейсных задач.

Задание.

1. Создать формы для ввода каждой из таблиц-справочников.
2. Создать сложную форму для таблиц, связанных отношением «1 ко многим».
3. Создать кнопочную форму, которая бы предоставляла доступ ко всем созданным формам и запросам.
4. Поместить в созданные формы кнопки навигации по записям и работы с формой (закрыть, напечатать, выйти из приложения и т.д.).
5. Создать модули для автоматической загрузки кнопочной формы при открытии базы данных.
6. Разработать руководство пользователя.

Отчет должен содержать:

- основную цель работы;
- описание последовательности выполнения задания;
- распечатки схем данных и создаваемых экранных форм и отчетов;
- распечатку разработанного руководства пользователя.

Терминология объектно-ориентированного программирования

Перед тем, как перейти к описанию преимуществ, которые дает ООП разработчикам программного обеспечения в процессе [проектирования](#), [кодирования](#) и [тестирования](#) программных продуктов необходимо познакомиться с наиболее часто встречающимися терминами в этой области.

Класс – тип данных, описывающий структуру и поведение объектов.

Объект – экземпляр класса.

Поле – элемент данных класса: переменная элементарного типа, структура или другой класс, являющийся частью класса.

Состояние объекта – набор текущих значений полей объекта.

Метод – процедура или функция, выполняющаяся в контексте объекта, для которого она вызывается. Методы могут изменять состояние текущего объекта или состояния объектов, передаваемых им в качестве параметров.

Свойство – специальный вид методов, предназначенный для модификации отдельных полей объекта. Имена свойств обычно совпадают с именами соответствующих полей. Внешне работа со свойствами выглядит точно так же, как работа с полями структуры или класса, но на самом деле перед тем, как вернуть или присвоить новое значение полю может быть выполнен программный код, осуществляющий разного рода проверки, к примеру, проверку на допустимость нового значения.

Член класса – поля, методы и свойства класса.

Модификатор доступа – дополнительная характеристика членов класса, определяющая, имеется ли к ним доступ из внешней программы, или же они используются исключительно в границах класса и скрыты от окружающего мира. Модификаторы доступа разделяют все элементы класса на детали реализации и открытый или частично открытый интерфейс.

Конструктор – специальный метод, выполняемый сразу же после создания экземпляра класса. Конструктор инициализирует поля объекта – приводит объект в начальное состояние. Конструкторы могут быть как с параметрами, так и без. Конструктор без параметров называют конструктором по умолчанию, который может быть только один. Имя метода конструктора, чаще всего, совпадает с именем самого класса.

Деструктор – специальный метод, вызываемый средой исполнения программы в момент, когда объект удаляется из оперативной памяти. Деструктор используется в тех случаях, когда в состав класса входят ресурсы, требующие явного освобождения (файлы, соединения с базами данных, сетевые соединения и т.п.)

Интерфейс – набор методов и свойств объекта, находящихся в открытом доступе и призванных решать определенный круг задач, к примеру, интерфейс формирования графического представления объекта на экране или интерфейс сохранения состояния объекта в файле или базе данных.

Статический член – любой элемент класса, который может быть использован без создания соответствующего объекта. К примеру, если метод класса не использует ни одного поля, а работает исключительно с переданными ему параметрами, то ничто не мешает его использовать в контексте всего класса, не создавая отдельных его экземпляров. Константы в контексте класса обычно всегда являются статическими его членами.

На этом с терминологией ООП далеко еще не все, но остальные понятия, связанные с этой парадигмой будут рассмотрены в следующем разделе.

Преимущества объектно-ориентированного программирования

Теперь поговорим о свойствах, которые приобретает программа при использовании объектно-ориентированного подхода к ее проектированию и кодированию. Как мне кажется, большинство этих свойств являются преимуществами ООП, но есть на этот счет и другие мнения...

- **Инкапсуляция** обозначает сокрытие деталей реализации классов средствами награднения отдельных его членов соответствующими модификаторами доступа. Таким образом, вся функциональность объекта, нацеленная на взаимодействие с другими объектами программы группируется в открытый интерфейс, а детали тщательно скрываются внутри, что избавляет основной код бизнес-логики [информационной системы](#) от ненужных ему вещей. Инкапсуляция повышает надежность работы программного кода, поскольку гарантирует, что определенные данные не могут быть изменены за пределами содержащего их класса.

- **Наследование.** Краеугольный камень ООП. В объектно-ориентированном программировании есть возможность наследовать структуру и поведение класса от другого класса. Класс, от которого наследуют, называется базовым или суперклассом, а класс, который получается вследствие наследования – производным или просто потомком. Любой класс может выступать как в роли суперкласса, так и в роли потомка. Связи наследования классов образуют иерархию классов. Множественным наследованием называют определение производного класса сразу от нескольких суперклассов. Не все объектно-ориентированные языки программирования поддерживают множественное наследование. Наследование – это эффективный способ выделения многократно используемых фрагментов кода, но у него есть и минусы, о которых будет рассказано далее.

- **Абстрагирование.** Возможность объединять классы в отдельные группы, выделяя общие, значимые для них всех характеристики (общие поля и общее поведение). Собственно, абстрагирование и есть следствие наследования: базовые классы не всегда имеют свою проекцию на объекты реального мира, а создаются исключительно с целью выделить общие черты целой группы объектов. К примеру, объект мебель – это базовое понятие для стола, стула и дивана, всех их объединяет то, что это движимое имущество, часть интерьера помещений, и они могут быть выполнены для дома или офиса, а также относиться к “эконом” или “премиум” классу. В ООП есть для этого отдельное понятие абстрактный класс – класс, объекты которого создавать запрещено, но можно использовать в качестве базового класса. Наследование и абстрагирование позволяют описывать структуры данных программы и связи между ними точно так же, как выглядят соответствующие им объекты в рассматриваемой [модели предметной области](#).

Контрольные вопросы для самопроверки

1. Класс. Объект. Состояние.
2. Метод Свойство Конструктор

3. Интерфейс
4. Полиморфизм
5. Преимущества объектно-ориентированного программирования
6. Недостатки объектно-ориентированного программирования.

Основная литература

1.-2. из раздела 7.

Дополнительная литература

3.-4. из раздела 7.

Практическое занятие № 5. Разработка блок-схем

Цель практического занятия

Освоение векторного редактора Visio в целях разработки блок-схем.

Задание на выполнение практической работы

- 1). Изучить ГОСТ 19.701–90 (ИСО 5807–85). Единая система программной документации. Схемы алгоритмов, программ данных и систем. Условные обозначения и правила выполнения.
- 2) Изучить и закрепить основы разработки блок-схем (материалы лекции 8. Разработка поведенческой модели).
- 3) Освоить векторный редактор Visio в части разработки блок-схем.
- 4) Построить блок-схему решения задачи по индивидуальному заданию.
- 5) Оформить и защитить отчет. В отчете должна быть приведена блок-схема решения задачи.

Варианты заданий:

1. Запросить число и вывести на экран все делители введенного числа, их количество, сумму и произведение.
2. Запросить число и вывести на экран четные делители введенного числа, а также их количество.
3. Запросить число и вывести на экран нечетные делители введенного числа, а также их количество.
4. Запросить число и вывести на экран простые делители введенного числа, а также их количество.
5. Запросить число и вывести на экран максимальный и минимальный нечетные делители введенного числа.
6. Запросить число и вывести на экран максимальный и минимальный четные делители введенного числа.
7. Найти и вывести на экран все четырехзначные числа, у которых сумма крайних цифр равна сумме средних.
8. Запросить число n и в интервале от 1 до n найти число, сумма делителей которого максимальна. Найденное число вывести на экран.
9. Запросить два числа, найти и вывести на экран наименьший общий делитель, отличный от 1.
10. Запросить два числа, найти и вывести на экран наибольший общий делитель.
11. Запросить два числа, найти и вывести на экран все общие нечетные делители.

Назначение Visio

Visio представляет собой векторный редактор, являющийся на текущий момент одной из лучших программ, предназначенных для построения схем и диаграмм различного назначения. Visio позволяет создавать диаграммы, изучаемые в рамках дисциплины (через тире указано имя набора объектов):

- IDEF0 - IDEF0 Diagram Shapes;
- DFD - Data Flow Diagram Shapes;
- ERD - Entity Relationship;
- блок-схемы - Basic Flowchart Shapes;
- диаграммы UML - UML

Общие сведения об интерфейсе Visio

В окне объектов отображаются наборы объектов, используемые при создании схем или диаграмм. Отображение или скрытие этого окна выполняется с помощью пункта меню "Вид / Окно объектов". Добавление конкретного набора объектов в окно объектов выполняется через пункт меню "Файл / Объекты / ...".

Создание и редактирование блок-схем

После создания нового файла необходимо задать размеры чертежа через пункт меню "Файл / Параметры страницы".

Набор объектов, предназначенных для создания блок-схем, выбирается с помощью пункта меню "Файл / Объекты / Flowchart / Basic Flowchart Shapes".

Создание блок-схемы заключается в переносе символа (объекта) из набора в рабочую область чертежа, указания его положения и размеров, а также создания надписей внутри символа.

Ниже приводятся некоторые особенности при работе с Visio:

- изменение размеров объекта выполняется с помощью изменения положения зеленых прямоугольников, расположенных по контуру объекта (рис.2). Если менять положение прямоугольников, расположенных в углах, при нажатой клавише Ctrl или Alt, то размеры объекты будут меняться пропорционально по высоте и ширине. Аналогичного эффекта можно добиться, если менять положение прямоугольников, расположенных по серединам граней объекта, при нажатой клавише Shift;

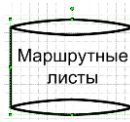


Рис. 2. Объект

- поворот объекта вокруг его центра выполняется с помощью изменения положения зеленого круга;
- для изменения положения, размера или угла наклона группы объектов необходимо вначале выделить эти объекты при нажатой клавише Shift, а затем выполнить требуемое действие;
- для создания надписи внутри объекта необходимо выполнить двойной щелчок левой кнопкой мыши при нахождении указателя мыши над объектом;
- для соединения символов линиями рекомендуется использовать объект "Dynamic connector". В этом случае при изменении положения объектов или их размеров Visio автоматически задаст новое положение соединяющих их линий.

Контрольные вопросы для самопроверки

1. Блок-схемы. Назначение, виды схем и символов.
2. Блок-схемы. Символы данных.
3. Блок-схемы. Символы процессов.
4. Блок-схемы. Символы линий и специальные линии.
5. Блок-схемы. Правила и рекомендации построения.

Основная литература

1.-2. из раздела 7.

Дополнительная литература

3.-4. из раздела 7.

9.2. Методические указания по выполнению контрольной работы

КР состоит из двух частей: приложение/программа и пояснительная записка к КР.

Требования к приложению:

1. Среда – Delphi. 10 компонент – edit, button, label, dbgrid, table, mainmenu ... 5 библиотек – standart, additional, internet, control ... 5 форм
2. Подключение/использование документа Word, Excel, Access
3. 10 функций – sin|cos, strtoint, intostr ...
4. 5 **различных** типов процедур – onclick button, create form ...
5. Подключение/использование данных из таблицы
6. Подключение/использование данных из БД (3 связанные таблицы)
7. Подключение/использование запроса к данным из таблицы
8. Подключение/использование запроса к данным из БД (3 связанные таблицы)
9. Использование данных из файла
10. Запись данных в файл
11. Обработка массива (числовых/текстовых) данных – сортировка или выборка (наименьшего, наибольшего)
12. Интерфейс одной из форм – 5 фреймов (например, ф1 – головное меню, ф2 – инструменты, ф3 – зона вывода (основной фрейм), ф4 – вспомогательный фрейм (например, ввод данных/параметров запроса, ф5 – запасной)
13. Создание html-страницы. Переход/открытие html-страницы. Информация/справка о разработчике

Требования к оформлению пояснительной записки:

1. Times new roman, 14
2. Межстрочный интервал – полуторный
3. Список литературы – 10 (ссылки на них внутри пояснительной записки)
4. **Обязательно** по одной таблице (например, анализ инструментальных средств разработки), графику, диаграмме (brwin, Erwin, rational rose), схеме (например, блок-схема алгоритма), рисунку
5. **Структура пояснительной записки (обязательные элементы):**
 - Введение (актуальность, проблема, пути решения проблемы, выбранный путь, цель курсовой работы, задачи кр, использованные методы исследования)
 - Анализ предметной области/постановка задачи

- Анализ инструментальных средств разработки
- Описание интерфейса программы/приложения
- Описание алгоритма работы приложения
- Описание разработанного приложения
- Заключение (аналогично введению)
- Используемые источники

Дополнительные элементы структуры пояснительной записки (можно вынести в приложения):

- Контекстная диаграмма и диаграмма декомпозиции бизнес-процессов исследуемой предметной области
- Логическая и физическая модели данных
- Диаграммы состояний, сценария, последовательности, развертывания
- Описание входных и выходных данных (типы данных, объемы, периодичность появления/изменения, документы, из которых они берутся/передаются)

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

- Microsoft Windows Russian
- Microsoft Office Russian
- Антивирусное программное обеспечение Kaspersky Security;
- Bpmn.io, Modelio
- Visual Studio Community

11 ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

<i>Вид занятия (Лк, ПЗ, СР...)</i>	<i>Наименование аудитории</i>	<i>Перечень основного оборудования</i>	<i>№ Лк, ПЗ</i>
1	3	4	5
Лк	Лекционная аудитория (мультимедийный класс)	Интерактивная доска SMART Board 680i2/Unifl, Интерактивный планшет Wacom PL-720, Колонки Microlab Solo-7C, Ноутбук Samsung R610<NP-R610-FS08>, Телевизор плазменный Samsung 63 PS-63A756T1M	Лк № 3,5-7
ПЗ	Дисплейный класс	Системный блок AMD A10-7800 Radeon R7 (12 шт.), Системный блок для слабовидящих пользователей AMD A10-7850K (1 шт.), Монитор Philips233 V5QHABP (13 шт.)	ПЗ № № 1-9
кр	Дисплейный класс	Системный блок AMD A10-7800 Radeon R7 (12 шт.), Системный блок для слабовидящих пользователей AMD A10-7850K (1 шт.), Монитор Philips233 V5QHABP (13 шт.)	
	Читальный зал №1	Оборудование 10 ПК i5-2500/H67/4Gb(монитор TFT19 Samsung); принтер HP LaserJet P2055D	
СР	Читальный зал №1	Оборудование 10 ПК i5-2500/H67/4Gb(монитор TFT19 Samsung); принтер HP LaserJet P2055D	-

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

1. Описание фонда оценочных средств (паспорт)

№ компетенции	Элемент компетенции	Раздел	Тема	ФОС
ПК-1	владение навыками использования основных теорий мотивации, лидерства и власти для решения стратегических и оперативных управленческих задач, а также для организации групповой работы на основе знания процессов групповой динамики и принципов формирования команды, умение проводить аудит человеческих ресурсов и осуществлять диагностику организационной культуры	1. Основы анализа и проектирования информационных систем	1.1 Основные понятия и терминология проектирования экономических информационных систем управления объектом 1.2 Информационные технологии 1.3 Жизненный цикл информационной системы	вопросы к экзамену 1.1-1.9
		3. Разработка поведенческой модели.	3.1 Основы построения поведенческих моделей 3.2 Блок-схемы алгоритмов 3.3 Правила и рекомендации построения блок-схем	вопросы к экзамену 3.1-3.4
ПК-11	владение навыками анализа информации о функционировании системы внутреннего документооборота организации, ведения баз	1. Основы анализа и проектирования информационных систем	1.1 Основные понятия и терминология проектирования экономических информационных систем управления объектом	вопросы к экзамену 1.10-1.16

	данных по различным показателям и формированию информационного обеспечения участников организационных проектов	2. Разработка информационной модели	2.1 Основы проектирования баз данных 2.2 Концептуальное проектирование с использованием методологии IDEF1X 2.3 Логическое проектирование с использованием методологии IDEF1X 2.4 Физическое проектирование с использованием методологии IDEF1X	вопросы к экзамену 2.1-2.28
ПК-13	Умение моделировать бизнес-процессы и использовать методы реорганизации бизнес-процессов в практической деятельности организаций	1 Основы анализа и проектирования информационных систем	1.1 Основные понятия и терминология проектирования экономических информационных систем управления объектом 1.2 Информационные технологии 1.3 Жизненный цикл информационной системы 1.4 Модели жизненного цикла 1.5 Основы анализа и проектирования информационных систем 1.6 Технологии и подходы к анализу и проектированию информационных систем 1.7 Разработка функциональной модели	вопросы к экзамену 1.17-1.50

2. Экзаменационные вопросы

№ п/п	Компетенции		ВОПРОСЫ К ЭКЗАМЕНУ	№ и наименование раздела
	Код	Определение		
1	2	3	4	5
1.	ПК-1	владение навыками использования основных теорий мотивации, лидерства и власти для решения стратегических и оперативных управленческих задач, а также для организации групповой работы на основе знания процессов групповой динамики и принципов формирования	1.1 Основные понятия и терминология проектирования экономических информационных систем управления объектом (ЭИС); 1.2 Технология проектирования; состав и содержание этапов проектирования 1.3 Проведение предпроектного обследования; 1.4 Проектирование технологических процессов обработки данных и внедрение программного продукта 1.5 Структура экономической информации. 1.6 Управленческая информация 1.7 Источники управленческой информации 1.8 Отбор информации и систематизация информации 1.9 Виды и формы коммуникаций	1. Основы анализа и проектирования информационных систем
			3.1 Определение понятия «схема» согласно ГОСТ	3.

		команды, умение проводить аудит человеческих ресурсов и осуществлять диагностику организационной культуры	19.701-90 3.2 Типы схем, их краткая характеристика 3.3 Типы символов, их краткая характеристика 3.4 Основные правила и рекомендации построения блок-схем	Разработка поведенческой модели.
2.	ПК-11	владение навыками анализа информации о функционировании системы внутреннего документооборота организации, ведения баз данных по различным показателям и формирования информационного обеспечения участников организационных проектов	1.10 Электронный документооборот, виды ЭДО, задачи ЭДО 1.11 Анализ существующей системы работы с документами 1.12 Формулирование требований к системе электронного документооборота (СЭД) и разработка плана ее внедрения 1.13 Разработка регламента работы СЭД 1.14 Положение об электронном документообороте 1.15 Проблемы внедрения систем электронного документооборота 1.16 Система электронного документооборота Workflow и ЕСМ-системы в рамках автоматизация бизнес-процессов	1. Основы анализа и проектирования информационных систем
			2.1 Основные этапы проектирования баз данных. 2.2 Основные элементы диаграмм «сущность–связь». 2.3 Основные возможности CASE-средств, поддерживающих построение информационных моделей. 2.4 Отличие независимой сущности от зависимой 2.5 Виды атрибутов. 2.6 Домен. Ключ. Типы ключей и их характеристика 2.7 Основные характеристики связи между сущностями. 2.8 Методология IDEF1X. Элементы диаграмм. 2.9 Концептуальное проектирование БД. Стадии. 2.10 Концептуальное проектирование БД. Сущности. 2.11 Концептуальное проектирование БД. Связи. 2.12 Концептуальное проектирование БД. Атрибуты. 2.13 Концептуальное проектирование БД. Ключи. 2.14 Концептуальное проектирование БД. Суперклассы и подклассы. 2.15 Цель логического проектирования БД. Стадии. 2.16 Логическое проектирование БД. Удаление элементов, не отвечающих реляционной модели данных. 2.17 Логическое проектирование БД. Определение требований поддержки целостности данных. 2.18 Основные элементы концептуальной модели, которые могут не отвечать реляционной модели данных. 2.19 Определение понятия «нормализация». 2.20 Первые четыре нормальные формы. 2.21 Полная функциональная и транзитивная зависимости. 2.22 Ограничения целостности.	2. Разработка информационной модели

			<p>2.23 Триггер. Действия, вызывающие срабатывание триггера</p> <p>2.24 Основные стратегии поддержания целостности при помощи триггеров на примере удаления.</p> <p>2.25 Цель физического проектирования. Стадии.</p> <p>2.26 Денормализация данных?</p> <p>2.27 Основные способы денормализации данных и их характеристику.</p> <p>2.28 Основные механизмы защиты, применяемые в реляционных базах данных</p>	
3.	ПК-13	<p>умение моделировать бизнес-процессы и использовать методы реорганизации бизнес-процессов в практической деятельности организаций</p>	<p>1.17 Классификация моделей жизненного цикла</p> <p>1.18 Инкрементная стратегия</p> <p>1.19 Спиральная стратегия</p> <p>1.20 Сравнительный анализ моделей</p> <p>1.21 Методологии, поддерживающие спиральную модель</p> <p>1.22 Особенности анализа и проектирования крупных систем</p> <p>1.23 Документы, содержащие требования на разработку системы</p> <p>1.24 Case-технологии анализа и проектирования ИС. Назначение и основные возможности Case-средств.</p> <p>1.25 Модели ИС.</p> <p>1.26 Принципы построения моделей.</p> <p>1.27 Сущность структурного подхода к анализу и проектированию ИС.</p> <p>1.28 Краткая характеристика методологий структурного анализа и проектирования ИС.</p> <p>1.29 Основные принципы проектирования</p> <p>1.30 Классификация моделей информационной системы</p> <p>1.31 Сущность структурного анализа и проектирования</p> <p>1.32 Основы функционального анализа и проектирования систем</p> <p>1.33 Назначение и состав методологии SADT (IDEF0)</p> <p>1.34 Методология IDEF0. Виды диаграмм.</p> <p>1.35 Методология IDEF0. Элементы диаграмм.</p> <p>1.36 Методология IDEF0. ICOM-коды.</p> <p>1.37 Методология IDEF0. Типы связей между работами.</p> <p>1.38 Методология IDEF0. Модели AS-IS, TO-BE и SHOULD-BE.</p> <p>1.39 Методология IDEF0. Правила и рекомендации построения диаграмм</p> <p>1.40 Элементы графической нотации IDEF0</p> <p>1.41 Типы связей между работами</p> <p>1.42 Правила и рекомендации построения диаграмм IDEF0</p> <p>1.43 ICOM-коды</p> <p>1.44 Элементы графической нотации DFD</p> <p>1.45 Правила и рекомендации построения DFD</p> <p>1.46 Диаграммы потоков данных. Назначение.</p> <p>1.47 Диаграммы потоков данных. Элементы диаграмм.</p> <p>1.48 Диаграммы потоков данных. Расширение DFD для систем реального времени.</p> <p>1.49 ERD. Назначение и основные элементы моделей.</p> <p>1.50 Возможности современных CASE-средств моделирования данных.</p>	<p>1. Основы анализа и проектирования информационных систем</p>

3. Описание показателей и критериев оценивания компетенций

Показатели	Оценка	Критерии
<p>Знать (ПК-1):</p> <ul style="list-style-type: none"> - источники управленческой информации, методики отбора источников информации; <p>(ПК-11):</p> <ul style="list-style-type: none"> - основные методы, способы и средства получения, хранения, обработки и передачи информации; - виды, состав и структуру документов, отражающих систему внутреннего документооборота организации; - методы и методику ведения баз данных по различным показателям и формирования информационного обеспечения участников организационных проектов; 	отлично	<p>Оценка «отлично» выставляется в случае, если студент демонстрирует:</p> <ul style="list-style-type: none"> – всестороннее систематическое знание основ теории анализа проектирования информационных систем; – правильное выполнение практических заданий, направленных на использование глубоких теоретических знаний и овладение инструментария проектирования информационных систем; – владеет современными методами разработки информационных моделей.
<p>(ПК-13):</p> <ul style="list-style-type: none"> - методы анализа и моделирования бизнес-процессов - основные стандарты, технологии и нотации моделирования бизнес-процессов - инструментальные системы, используемые для описания и анализа бизнес-процессов; <p>Уметь (ПК-1):</p> <ul style="list-style-type: none"> - находить способы решения конфликта в организации и проектирования межличностных, групповых и организационных коммуникаций; 	хорошо	<p>Оценка «хорошо» выставляется в случае, если студент демонстрирует:</p> <ul style="list-style-type: none"> – недостаточно знание основ теории анализа проектирования информационных систем; – выполнение с несущественными ошибками практических заданий, направленных на использование глубоких теоретических знаний и овладение проектирования информационных систем; – не в полной мере владеет современными методами разработки информационных моделей.
<p>(ПК-11):</p> <ul style="list-style-type: none"> - осуществлять выбор информационных средств для их обработки в соответствии с поставленной задачей; <p>(ПК-13):</p> <ul style="list-style-type: none"> - моделировать, анализировать и совершенствовать бизнес-процессы с использованием изученных стандартов, технологий и нотаций моделирования; - рецензировать модель бизнес-процесса; 	удовлетворительно	<p>Оценка «удовлетворительно» выставляется в случае, если студент демонстрирует:</p> <ul style="list-style-type: none"> – частичное знание основ теории анализа проектирования информационных систем; – частичное выполнение с ошибками практических заданий, направленных на использование глубоких теоретических знаний и овладение проектирования информационных систем; – не в полной мере владеет современными методами разработки информационных моделей.
<p>Владеть (ПК-1):</p> <ul style="list-style-type: none"> - навыками выполнения заданий в составе команды и организации партнерского взаимодействия для решения стратегических и оперативных управленческих задач; <p>(ПК-11):</p> <ul style="list-style-type: none"> - навыками анализа информации о функционировании системы внутреннего документооборота, проектирования и ведения баз данных; <p>(ПК-13):</p> <ul style="list-style-type: none"> - практическими навыками 	неудовлетворительно	<p>Оценка «неудовлетворительно» выставляется в случае, если студент демонстрирует:</p> <ul style="list-style-type: none"> – существенные пробелы в знании основных теоретических положений экономики как науки на уровне организации; – принципиальные ошибки при выполнении практических заданий, направленных на использование глубоких теоретических знаний и овладение проектирования информационных систем; – не владеет современными методами

<p>моделирования, анализа и документирования бизнес-процессов с помощью инструментальных сред;</p> <ul style="list-style-type: none"> - терминологией из области моделирования бизнес-процессов; - методами построения, анализа и документирования моделей бизнес-процессов. 		<p>разработки информационных моделей и систем.</p>
--	--	--

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности

Цель и задачи дисциплины Б1.В.08 Современные технологии анализа и проектирования информационных систем представлены в разделе 1 настоящей рабочей программы. Место дисциплины в структуре образовательной программы представлено в разделе 2 настоящей рабочей программы. Распределение объема дисциплины по формам обучения с указанием видов учебных занятий представлено в разделе 3 настоящей рабочей программы. Содержание дисциплины указано в разделе 4 настоящей рабочей программы.

Учебно-методические материалы для самостоятельной работы студентов по дисциплине находятся в свободном доступе в соответствии с разделом 6 настоящей рабочей программы.

При изучении дисциплины необходимо использовать литературу, указанную в разделе 7 настоящей рабочей программы, а также перечень ресурсов информационно-телекоммуникационной сети «Интернет», представленных в разделе 8 настоящей рабочей программы.

Консультации для студентов по дисциплине проводятся в соответствии с графиком проведения консультаций, представленном на стенде кафедры, за которой закреплена указанная дисциплина.

К экзамену допускаются студенты очной формы обучения, которые выполнили, оформили и защитили все практические работы, предусмотренные в конкретном семестре. Методические указания по выполнению и оформлению представлены в разделе 9.1. настоящей рабочей программы.

К экзамену допускаются студенты заочной формы обучения, которые теоретически подготовлены по всем вопросам и владеют навыками эффективного применения основных подходов и методов современного анализа и проектирования информационных систем.

Информационные технологии, используемые при освоении дисциплины, перечислены в разделе 10 настоящей рабочей программы.

Оценка знаний, умений, навыков осуществляется в процессе промежуточной аттестации обучающихся по дисциплине, которая осуществляется в виде экзамена. Для оценивания знаний, умений, навыков используются ФОС по дисциплине.

Экзамен проводится в устной форме по выданному преподавателем заданию.

По итогам выполненного задания преподаватель оценивает уровень знаний, умений, навыков. Описание показателей и критериев оценивания компетенций, сформированных по итогам изучения дисциплины, представлено в разделе 3 Приложения 1 настоящей рабочей программы. Основными оценочными средствами при проведении промежуточной аттестации являются вопросы к экзамену.

АННОТАЦИЯ

рабочей программы дисциплины

Современные технологии анализа и проектирования информационных систем

1. Цель и задачи дисциплины

Цель дисциплины

Целью изучения дисциплины является: овладение обучающимися основными теоретическими положениями в области проектирования автоматизированных информационных систем, знаниями этапов создания и видов создаваемых проектов, а также получение обучающимися практических навыков использования современных подходов в проектировании информационных автоматизированных систем.

Задачами изучения дисциплины являются:

- знакомство с концептуальными основами CASE-технологий, с эволюцией развития и классификацией CASE-средств, понятием и основными принципами функционального моделирования, с основными методологиями.
- закрепление навыков работы с инструментами структурного анализа и проектирования SADT (Structure Analysis and Design Technique)
- получение практических навыков построения структурных карт, диаграмм переходов состояний;
- получение практических навыков использования процессного подхода к моделированию деятельности.

2. Структура дисциплины

2.1 Распределение трудоемкости по отдельным видам учебных занятий, включая самостоятельную работу: 18 ч. лекции, 36 ч. практические работы, самостоятельная работа 63 ч.

Общая трудоемкость дисциплины составляет 144 ч., 4 зачетных единицы

2.2 Основные разделы дисциплины:

1. Основы анализа и проектирования информационных систем.
2. Разработка информационной модели.
3. Разработка поведенческой модели.

3. Планируемые результаты обучения (перечень компетенций)

Процесс изучения дисциплины направлен на формирование следующих компетенций:

ПК-1 - владение навыками использования основных теорий мотивации, лидерства и власти для решения стратегических и оперативных управленческих задач, а также для организации групповой работы на основе знания процессов групповой динамики и принципов формирования команды, умение проводить аудит человеческих ресурсов и осуществлять диагностику организационной культуры;

ПК-11 - владение навыками анализа информации о функционировании системы внутреннего документооборота организации, ведения баз данных по различным показателям и формирования информационного обеспечения участников организационных проектов;

ПК-13 - умение моделировать бизнес-процессы и использовать методы реорганизации бизнес-процессов в практической деятельности организаций.

4. Вид промежуточной аттестации: экзамен

*Протокол о дополнениях и изменениях в рабочей программе
на 20__-20__ учебный год*

1. В рабочую программу по дисциплине вносятся следующие дополнения:

2. В рабочую программу по дисциплине вносятся следующие изменения:

Протокол заседания кафедры № _____ от « ____ » _____ 20 ____ г.,
(разработчик)

Заведующий кафедрой _____

(подпись)

(Ф.И.О.)

Программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 38.03.02 Менеджмент от «12» января 2016 г. № 7

для набора 2014 года: и учебным планом ФГБОУ ВО «БрГУ» для заочной формы обучения от «03» июля 2018 г. № 413

для набора 2015 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «03» июля 2018 г. № 413, для заочной формы обучения от «03» июля 2018 г. № 413

Программу составил:

Вахрушева М.Ю., доцент баз. каф. МиИТ, доцент, к.физ.-мат.н. _____

Рабочая программа рассмотрена и утверждена на заседании базовой кафедры МиИТ

от «19» декабря 2018 г., протокол № 8

И.о. заведующего базовой кафедрой МиИТ _____ Е.И. Луковникова

СОГЛАСОВАНО:

И.о. заведующего выпускающей базовой кафедрой МиИТ _____ Е.И. Луковникова

Директор библиотеки _____ Т.Ф. Сотник

Рабочая программа одобрена методической комиссией факультета ФЭиУ

от «28» декабря 2018 г., протокол № 4

Председатель методической комиссии факультета _____ Е.В. Трапезникова

СОГЛАСОВАНО:

Начальник учебно-методического управления _____ Г.П. Нежевец

Регистрационный № _____