

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Ламбузович
Должность: И.о. ректора
Дата подписания: 21.08.2023 02:39:13
Уникальный программный ключ:
2a04bb882d7edb7f479cb266eb4aaaaedebeea849

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО**

**«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»**

**Учебно-методические указания по выполнению
курсовой работы по дисциплине
«Web – программирование»
для студентов направления подготовки бакалавров
01.03.02 – «Прикладная математика и информатика»**

Махачкала 2022

УДК 519.682

Учебно-методические указания по выполнению курсовой работы по дисциплине «Web – программирование» для студентов направления подготовки бакалавров 01.03.02 – «Прикладная математика и информатика» - Махачкала: ДГТУ, 2022. -23с.

В учебно-методических указаниях рассмотрены основные этапы выполнения курсовой работы, содержание каждого раздела, требования к оформлению и процедуре защиты. Представлены теоретические сведения и практические рекомендации по основным этапам проектирования и реализации профессиональных Интернет – приложений.

В приложениях приводятся примерная тематика работ и образцы оформления.

Указания предназначены для студентов очной формы обучения по направлению подготовки бакалавров 01.03.02 «Прикладная математика и информатика»

Составители:

доцент кафедры ПМИИ, к.т.н.
зав. каф. ПМИИ, доцент, к.ф.-м.н.

Мирземагомедова М.М.
Исабекова Т.И.

Рецензенты:

Заведующий кафедры ПИВЭиИТ ДГТУ, к.э.н., доцент

Мурадов М.М.

Директор НИИ РПИ ДГНУХ, д.э.н.

Савзиханова С.Э.

Печатается согласно постановлению
Ученого Совета Дагестанского Государственного Технического Университета.

от «_____» _____ 2022г.

Оглавление

Цели курсовой работы.....	4
Тематика курсовых работ.....	4
1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОЦЕССА ПРОЕКТИРОВАНИЯ ИНТЕРНЕТ – ПРИЛОЖЕНИЯ	4
2. ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ	5
3. ОСНОВНЫЕ РАЗДЕЛЫ СОДЕРЖАНИЯ КУРСОВОЙ РАБОТЫ, ПОРЯДОК И ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ	5
4. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ИНТС	6
5. ИСХОДНЫЕ ДАННЫЕ ДЛЯ ПРОЕКТИРОВАНИЯ, СТАНДАРТЫ	8
6. ПРОЕКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА	9
7. ПОСТРОЕНИЕ ИНФОРМАЦИОННО-ЛОГИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ.....	10
8. РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ДАННЫХ.....	12
9. СТРУКТУРА ПРОГРАММНЫХ МОДУЛЕЙ, РАЗРАБОТКА АЛГОРИТМОВ, ЛОГИЧЕСКИЙ АНАЛИЗ СТРУКТУР ИНТС	13
10. РАЗРАБОТКА МОДЕЛЕЙ ЗАЩИТЫ ДАННЫХ	15
11. АВТОМАТИЗАЦИЯ ПРОЦЕССА ПРОЕКТИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ CASE-ТЕХНОЛОГИЙ.....	17
12. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ	18
13. ПОРЯДОК ЗАЩИТЫ И ОТВЕТСТВЕННОСТЬ СТУДЕНТА ЗА ВЫПОЛНЕНИЕ ЗАДАНИЯ КУРСОВОЙ РАБОТЫ	19
Список литературы	20
Приложение 1	21
Приложение 2	22
Приложение 3	23

Цели курсовой работы.

Целью курсовой работы является закрепление теоретических знаний, а также практических навыков проектирования и разработки Web – сайтов, полученных при изучении дисциплины «Web – программирование» и смежных с ней дисциплин, изучаемых по кафедре «Прикладная математика и информатика».

Курсовая работа представляет собой комплексную проектную, практическую, внедренческую разработку, которая в дальнейшем может стать основой выпускной дипломной работы.

Тематика курсовых работ.

Темы курсовых работ можно разбить на несколько групп:

- 1) Проектирование и разработка интерактивных и динамических Web – сайтов для конкретных предметных областей.
- 2) Возможности интеграции средств разработки Web – сайтов с системами управления и администрирования баз данных.
- 3) Сравнительный анализ средств автоматизации проектирования Web – сайтов.
- 4) Научно-исследовательские темы по любому из направлений по тематике «Web – программирование».

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОЦЕССА ПРОЕКТИРОВАНИЯ ИНТЕРНЕТ – ПРИЛОЖЕНИЯ.

Современные информационные технологии предоставляют широкий набор способов реализации Интернет-приложений, выбор которых осуществляется на основе требований со стороны предполагаемых пользователей, которые, как правило, изменяются в процессе разработки. Для теории принятия решений процесс проектирования системы – это процесс принятия проектно-конструкторских решений, направленных на получение версии системы, удовлетворяющей требованиям заказчика.

Под *проектом* будем понимать проектно-конструкторскую и технологическую документацию, в которой представлено описание проектных решений по созданию и эксплуатации системы в конкретной программно-технической среде.

Под *проектированием Интернет-системы* (ИнтС) понимается процесс преобразования входной информации об объекте проектирования, о методах проектирования и об опыте проектирования объектов аналогичного назначения в проект ИнтС. С этой точки зрения проектирование ИнтС сводится к последовательной формализации проектных решений на различных стадиях жизненного цикла системы: предпроектного анализа требований, технического и рабочего проектирования, внедрения и эксплуатации ИнтС. Осуществление проектирования системы предполагает использование проектировщиками определенной технологии проектирования, соответствующей масштабу и особенностям разрабатываемого проекта.

Технология проектирования Интернет-системы – это совокупность методологии и средств проектирования системы, а также методов и средств организации проектирования (управление процессом создания и модернизации проекта системы).

В основе технологии проектирования лежит технологический процесс, который определяет действия, их последовательность, состав исполнителей, средства и ресурсы, требуемые для выполнения этих действий.

Технологический процесс проектирования системы в целом делится на совокупность последовательно-параллельных, связанных и соподчиненных цепочек действий, каждое из которых может иметь свой предмет.

Проектирование системы – трудоемкий, длительный и динамический процесс. Технологии проектирования, применяемые в настоящее время, предполагают поэтапную разработку системы. Этапы по общности могут разделяться в стадии.

Совокупность стадий и этапов, которые проходит система в своем развитии с момента принятия решения о создании системы до момента прекращения функционирования системы, называется *жизненным циклом системы*.

Суть содержания жизненного цикла разработки системы в различных подходах одинакова и сводится к выполнению следующих стадий:

Планирование и анализ требований (предпроектная стадия) – системный анализ. Исследование и анализ существующей системы, определение требований к создаваемой системе, оформление технико-экономического обоснования и технического задания на разработку системы.

Проектирование (техническое проектирование, логическое проектирование). Разработка в соответствии со сформулированными требованиями состава автоматизируемых функций и состава обеспечивающих подсистем, оформление технического проекта системы.

Реализация проекта (рабочее проектирование, физическое проектирование, программирование).

Разработка и настройка программ, наполнение баз данных, создание рабочих инструкций для персонала, оформление рабочего проекта.

Внедрение (тестирование, опытная эксплуатация). Комплексная отладка подсистем, обучение персонала, поэтапное внедрение системы в эксплуатацию, оформление акта о приемо-сдаточных испытаниях системы.

Эксплуатация системы (сопровождение, модернизация). Сбор рекламаций и статистики о функционировании системы, исправление ошибок и недоработок, оформление требований к модернизации системы и ее выполнение.

2. ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Курсовая работа представляет собой самостоятельную разработку программной, аппаратной или технологической компоненты ИС.

Основные этапы выполнения курсовой работы являются **контрольными заданиями**, информирующими преподавателя о ходе выполнения работы студентом.

В курсовой работе, как правило, должны быть представлены результаты отладки проектируемых компонент в средствах выбранной CASE-среды.

Тема курсовой работы утверждается в начале семестра на установочных консультациях. По мере выполнения этапов курсовой работы, студенты в часы консультаций представляют материалы преподавателю, и в ходе диалога уточняют и формируют соответствующие разделы (консультации могут осуществляться через электронную почту).

Студент обязан выполнять этапы курсовой работы в соответствии со следующим графиком:

- постановка задачи, анализ решений и функциональная разработка системы – (4-5)-я недели;
- разработка информационного обеспечения, функциональных модулей, интерфейсов – (10-11)-я недели;
- отладочные работы, оформление пояснительной записки, графических материалов и подготовка к защите – (14-15) недели.

Защита курсовой работы проводится по утвержденному графику, консультации организуются по расписанию кафедры.

3. ОСНОВНЫЕ РАЗДЕЛЫ СОДЕРЖАНИЯ КУРСОВОЙ РАБОТЫ, ПОРЯДОК И ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Пояснительная записка курсовой работы должна содержать следующие элементы:

1. Титульный лист (приложение 3).
 2. Лист задания (приложение 2).
 3. Аннотацию.
 4. Оглавление.
 5. Введение.
 6. Основную часть.
 7. Заключение.
 8. Библиографию и список ссылок на Интернет-ресурсы.
 9. Приложения.
1. Во введении обосновывается актуальность выбранной темы, т.е. предметной области, а также выбранной технологии и инструментальных средств проектирования и разработки.
 2. Системный анализ предметной области (показать специфику предметной области, которую необходимо учесть при разработке Интернет-приложения).
 3. Постановку задачи (включая все ее компоненты). Требования к ИнтС.
 4. Описание этапов проектирования ИнтС.
 - 4.1. Проектирование структуры ИнтС (описание компонентов системы согласно специфике работы, определение состава элементов системы, синтез конструктивных компонентов системы, проектирование нескольких вариантов будущей системы, выбор наилучшего варианта для реализации с обоснованием).
 - 4.2. Проектирование базы данных (концептуальное проектирование: перечень сущностей, перечень атрибутов; инфологическое проектирование: модель «сущность-связь», описание связей между сущностями; нормализация отношений).
 5. Выбор средств реализации проекта. Требования к техническому обеспечению.
 6. Описание этапов реализации ИнтС.
 - 6.1. Дatalogическое проектирование баз данных (создание системы таблиц, обеспечение целостности данных, запросы к базе данных).
 - 6.2. Поэтапное описание реализации компонентов ИнтС.
 - 6.3. Описание этапов тестирования и внедрения.
 7. Заключение должно содержать оценку полученных результатов и изложение основных направлений дальнейшего совершенствования работы.

В приложения к пояснительной записке могут быть включены следующие материалы:

 - Словарь терминов, используемых в данной предметной области.
 - Логическая и физическая структуры Web-сайта.
 - Формы входных документов.
 - Формы выходных документов.
 - Распечатки реализованных запросов.
 - Распечатки реализованных отчетов.
 - Программные коды компонентов приложения.
 - Инструкция по использованию ИнтС.
 8. Графические объекты (ссылка на которые осуществляется из основной части пояснительной записки):
 - 9.1. Экранные формы.
 - 9.2. Дополнительные схемы для подробного описания структуры ИнтС.

4. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ИНТС

Предпроектная стадия, включающая проведение необходимых исследований по формированию структуры ИнтС и получении в конечном счете задания на проектирование, предполагает проведение работы по сбору необходимой информации в связи с

особенностями предметной области.

Функциональная модель предполагает наличие схем, подробно описывающих назначение (функции) каждого элемента в структуре системы. Для построения функциональной модели системы можно использовать различные подходы: структурный анализ или объектно-ориентированный подход.

Структурным анализом SADT (Structured Analysis and Design Technique) принято называть метод исследования системы с помощью ее графического модельного представления, которое начинается с общего обзора и последующей детализации, в иерархическую структуру со все большим числом уровней. Для таких методов характерно: разбиение на уровни абстракции с ограничением числа элементов на каждом из уровней (обычно от 3 до 9); ограниченный контекст, включающий лишь существенные на каждом уровне детали; дуальность данных и операций над ними; использование строгих формальных правил записи; последовательное приближение к конечному результату.

Анализ является первым этапом создания ИС, на котором требования заказчика уточняются, формализуются и документируются. Фактически на этом этапе дается ответ на вопрос: «Что должна делать будущая система?». Именно здесь лежит ключ к успеху всего проекта.

Структурный анализ начинается с исследования того, как организована система управления предприятием, с обследования функциональной и информационной структуры системы управления. По результатам обследования аналитик на первой стадии анализа строит обобщенную логическую модель исходной предметной области, отображающую ее функциональную структуру, особенности основной деятельности и информационное пространство, в котором эта деятельность осуществляется. Используя специальную терминологию, можно сказать, что аналитик строит модель «как есть».

Вторая стадия работы, к которой привлекаются заинтересованные представители заказчика, а при необходимости и независимые эксперты, состоит в анализе модели «как есть», выявлении ее недостатков и узких мест, определение путей совершенствования системы управления на основе выделенных критериев качества.

Третья стадия анализа, содержащая элементы проектирования, – создание усовершенствованной обобщенной логической модели, отображающей реорганизованную предметную область или ее часть, которая подлежит автоматизации. Эту модель можно назвать моделью «как надо», т.е. здесь происходит формализация системы.

На ряду со структурным подходом существует и более мощный подход называемый объектно-ориентированным ООП. Эта методология создана для проектирования больших и сложных систем и имеет ряд преимуществ перед структурным подходом.

ООП базируется на создании объектной модели, которая описывает структуру объектов, составляющих систему, их атрибуты, операции, взаимосвязи с другими объектами. Цель разработки объектной модели – описать объекты, составляющие в совокупности проектируемую систему, а также выявить и указать различные зависимости между объектами.

Декомпозиция проблемы на объекты – творческий, плохо формализуемый процесс. В объектной модели должны быть отражены те понятия и объекты реального мира, которые важны для разрабатываемой системы. В объектной модели отражается прежде всего прагматика разрабатываемой системы, что выражается в использовании терминологии прикладной области, связанной с использованием разрабатываемой системы.

Объектная модель имеет четыре главных элемента: абстрагирование, инкапсуляция, модульность, иерархия.

Эти элементы являются главными в том смысле, что без любого из них модель не будет объектно-ориентированной. Кроме главных, имеются еще три дополнительных элемента: типизация, параллелизм, сохраняемость.

Абстракция выделяет существенные характеристики некоторого объекта,

отличающие его от всех других видов объектов и, таким образом, четко определяют его концептуальные границы с точки зрения наблюдателя.

Инкапсуляция – это процесс отделения друг от друга элементов объекта, определяющих его устройство и поведение; инкапсуляция служит для того, чтобы изолировать контрактные обязательства абстракции от их реализации.

Модульность – это свойство системы, которая была разложена на внутренне связанные, но слабо связанные между собой модули.

Иерархия – это упорядочение абстракций, расположение их по уровням.

Типизация – это способ защититься от использования объектов одного класса вместо другого, или по крайней мере управлять таким использованием.

Параллелизм – это свойство, отличающее активные объекты от пассивных.

Сохраняемость – способность объекта существовать во времени, переживая породивший его процесс, и (или) в пространстве, перемещаясь из своего первоначального адресного пространства.

5. ИСХОДНЫЕ ДАННЫЕ ДЛЯ ПРОЕКТИРОВАНИЯ, СТАНДАРТЫ

Методы, которые используются для изучения предметной области и технологии получения от экспертов сведений о системе, подлежащей описанию, называются сбором данных, а в информатике она больше известна как опрос (интервьюирование) или извлечение знаний. Но как бы она не называлась, способность собрать необходимую информацию, основанную на знаниях экспертов, весьма существенна для построения точной и полезной модели. Поэтому технология сбора информации составляет важную часть структурной методологии SADT.

Опрос – это сбор сведений. Первый опрос служит точкой отсчета в процессе моделирования. Чтобы провести опрос, аналитик вначале выбирает наилучший источник информации (документ или конкретного человека), а затем организует его "опрос". Цель опроса – получение порции информации, необходимой для начала либо для продолжения построения определенной части модели. После первого опроса SADT – модель используется для определения той информации, которую необходимо получить в ходе следующего опроса. В соответствии с иерархией модели может быть проведена последовательность опросов для выяснения все более конкретных деталей рассматриваемой области.

Обычно источниками информации служат эксперты. Часто именно они являются наилучшими источниками, потому что им знакомы текущие нюансы и недокументированные аспекты системы. Самое важное – это то, что экспертам известны факты, которые не отражены в документах или которые трудно объяснить. Их можно получить только путем опроса экспертов. Чтобы подготовиться к такому опросу, нужно исследовать другие источники информации, например документы. Существует множество различных стратегий для извлечения информации из этих источников: чтение документов, наблюдение за выполняемыми операциями, анкетирование, использование собственных знаний, составление описания.

Документы – хороший источник информации, потому что они чаще всего доступны и их можно "опрашивать" в удобном для себя темпе. Чтение документов – прекрасный способ получить первоначальное представление о системе и сформулировать вопросы к экспертам.

Наблюдение за работой моделируемой системы – хорошая стратегия получения информации. Оно должно проводиться всегда, когда есть такая возможность. Через наблюдение, а возможно, и участие аналитики получают информацию о происходящих день за днем операциях из первых рук. Во время наблюдения за работой системы часто возникают вопросы, которые никогда бы не появились, если бы аналитик только читал документы или разговаривал с экспертами. Однако следует соблюдать осторожность.

Слишком долгие наблюдения могут привести к избыточному привыканию к текущему состоянию дел. Из-за потери объективности можно не увидеть альтернативные пути описания функций системы.

Анкетирование проводится для того, чтобы опросить большие группы экспертов в сжатые сроки. Его можно использовать, например, когда необходимо быстро получить сведения о работе какой-либо определенной части системы с разных позиций. Анкетирование при опросе экспертов позволяет выявить, какие части системы более всего нуждаются в улучшении. На практике, однако, информация, полученная от экспертов с помощью анкет, оказывается недостаточно достоверной.

Еще одна полезная стратегия – придумать описание и дать его экспертам для корректировки. Придуманные описания могут дать альтернативные схемы функционирования системы – схемы, о которых эксперты никогда не думали.

Однако для реализации таких схем нужна поддержка. Предварительно нужно изучить предметную область и найти хотя бы одну доброжелательно настроенную группу экспертов, прежде чем пытаться придумать описание. В этом случае описание будет иметь шансы отразить реальность. Придуманные описания могут, однако, потерпеть неудачу, если эксперты не готовы воспринимать новые возможности. В идеале, прежде чем прибегать к этой стратегии, автор должен установить надежные контакты с несколькими экспертами.

6. ПРОЕКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Проектирование интерфейса пользователя – сложная многофакторная и многовариантная задача, требующая системного подхода. В настоящее время считается доказанным, что решение данной задачи заключается не в том, чтобы рационально «вписать» человека в контур управления, а в том, чтобы, исходя из задач управления объектом, разработать систему взаимодействия двух равноправных партнеров (человек-оператор и аппаратно-программный комплекс информационной системы).

Цель создания эффективного эргономичного пользовательского интерфейса состоит в том, чтобы отобразить информацию настолько эффективно насколько это возможно для человеческого восприятия и структурировать отображение на дисплее таким образом, чтобы привлечь внимание к наиболее важным единицам информации.

Пользователь должен иметь возможность манипулировать объектами в среде приложения. Неплохо, если они (графические элементы) будут ему понятны и станут нести в себе информацию о том, что это такое и что произойдет, если выбрать или произвести действие над каким-то объектом. Иллюстрация этой идеи – панель быстрого доступа Word'a. Что еще, кроме как сохранение файла, можно ожидать от кнопки с изображением дискеты? Необходимо, чтобы пользователь имел наглядное средство отображения информации на различных этапах решения задач, он должен видеть, как его действия влияют на объекты, расположенные на экране.

Создание эффективного интерфейса заключается в быстром, насколько это возможно, развитии у пользователей простой концептуальной модели интерфейса. Концепция согласованности состоит в том, что при работе с компьютером у пользователя формируется система ожидания одинаковых реакций на одинаковые действия, что постоянно подкрепляет пользовательскую модель интерфейса.

Приложение должно проектироваться таким образом, чтобы пользователь в любой момент и на любом этапе работы мог получить помощь, контекстную справку или подсказку.

Безусловно, пользователю нужно дать возможность экспериментировать в приложении (нажатие любых кнопок, изменение настроек и т.д.). Но при этом необходимо избавить его от тупиковых ситуаций: все последствия экспериментов должны быть исправимы, а в лучшем случае еще и обратимы.

Интерфейс должен предоставлять информацию о том, что происходит в данный момент на компьютере. Нельзя допускать, чтобы пользователь долгое время ожидал реакции приложения на некоторое свое действие.

Цветовая гамма, компоновка элементов, пиктограммы, звуки, анимация – все должно помогать пользователю при выполнении задачи. Но здесь важно не переборщить, т.к. в этом случае внимание человека начнет рассеиваться, у него появится раздражение и, как следствие снизится эффективность работы.

Начальным этапом разработки пользовательского интерфейса являются создание его ассоциативной модели, после чего осуществляется проработка концептуального дизайна. Здесь необходимо разработать необходимый набор интерфейсных элементов, каждый из которых должен обладать определенным цветом, формой, надписью и т. п., и все вместе они должны составлять единую систему, вызывающую стойкую систему ассоциаций у пользователей.

Цвет – мощный визуальный инструмент, который необходимо использовать очень осторожно, чтобы не вызвать неадекватной реакции пользователя. Целесообразно ограничить число цветов до 4 на экране и до 7 для последовательности экранов. Для неактивных элементов рекомендуется использовать бледные цвета. Необходимо использовать цвета согласно представлениям пользователя. Например, для картографа зеленый цвет – лес, желтый – пустыня, синий – вода. Если цвет используется для кодировки информации, необходимо удостовериться, что код адекватно воспринимается пользователем: красный – опасность/стоп, зеленый – нормально/продолжение работы, желтый – предостережение. Для привлечения внимания наиболее эффективны белый, желтый и красный цвета.

Меню необходимый элемент любой автоматизированной системы, позволяющий пользователю выполнять задачи внутри приложения и управлять процессом решения. Достоинство меню в том, что пользователи не должны помнить название элемента или действия, которое они хотят выполнить – они должны только распознать его среди пунктов меню.

Сообщения необходимы для направления пользователя в нужную сторону, подсказок и предупреждений для выполнения необходимых действий на пути решения задачи. Они также включают подтверждения действий со стороны пользователя и подтверждения, что задачи были выполнены системой успешно либо по каким-то причинам не выполнены. Сообщения могут быть обеспечены в форме диалога, экранных заставок и т.п.

7. ПОСТРОЕНИЕ ИНФОРМАЦИОННО-ЛОГИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью.

Примеры понятий – "сотрудник", "отдел", "проект", "зарплата". Примеры взаимосвязей между понятиями – "сотрудник числится ровно в одном отделе", "сотрудник может выполнять несколько проектов", "над одним проектом может работать несколько сотрудников". Примеры ограничений – "возраст сотрудника не менее 16 и не более 60 лет".

Логическая модель данных является начальным прототипом будущей базы данных. Логическая модель строится в терминах информационных единиц, но без привязки к конкретной СУБД. Более того, логическая модель данных необязательно должна быть выражена средствами именно реляционной модели данных. Основным средством разработки логической модели данных в настоящий момент являются различные варианты ER-диаграмм (Entity-Relationship, диаграммы сущность-связь).

Одну и ту же ER-модель можно преобразовать как в реляционную модель данных, так и в модель данных для иерархических и сетевых СУБД, или в постреляционную модель

данных. Решения, принятые на предыдущем уровне, при разработке модели предметной области, определяют некоторые границы, в пределах которых можно развивать логическую модель данных, в пределах же этих границ можно принимать различные решения. Например, модель предметной области складского учета содержит понятия "склад", "накладная", "товар". При разработке соответствующей реляционной модели эти термины обязательно должны быть использованы, но различных способов реализации тут много – можно создать одно отношение, в котором будут присутствовать в качестве атрибутов "склад", "накладная", "товар", а можно создать три отдельных отношения, по одному на каждое понятие.

При разработке логической модели данных возникают вопросы: хорошо ли спроектированы отношения? Правильно ли они отражают модель предметной области, а следовательно и саму предметную область?

Для того чтобы оценить качество принимаемых решений на уровне логической модели данных, необходимо сформулировать некоторые критерии качества в терминах физической модели и конкретной реализации и посмотреть, как различные решения, принятые в процессе логического моделирования, влияют на качество физической модели и на скорость работы базы данных.

Таких критериев может быть очень много и выбор их произволен. Некоторые из таких критериев являются важными с точки зрения получения качественной базы данных: адекватность базы данных предметной области, легкость разработки и сопровождения базы данных, скорость выполнения операций обновления данных (вставка, обновление, удаление кортежей), скорость выполнения операций выборки данных.

База данных должна адекватно отражать предметную область. Это означает, что должны выполняться следующие условия.

1 Состояние базы данных в каждый момент времени должно соответствовать состоянию предметной области.

2 Изменение состояния предметной области должно приводить к соответствующему изменению состояния базы данных.

3 Ограничения предметной области, отраженные в модели предметной области, должны некоторым образом отражаться и учитываться базе данных.

Практически любая база данных, за исключением совершенно элементарных, содержит некоторое количество программного кода в виде триггеров и хранимых процедур.

Хранимые процедуры – это процедуры и функции, хранящиеся непосредственно в базе данных в откомпилированном виде и которые могут запускаться пользователями или приложениями, работающими с базой данных. Основное назначение хранимых процедур – реализация бизнес- процессов предметной области.

Триггеры – это хранимые процедуры, связанные с некоторыми событиями, происходящими во время работы базы данных. В качестве таких событий выступают операции вставки, обновления и удаления строк таблиц. Если в базе данных определен некоторый триггер, то он запускается автоматически всегда при возникновении события, с которым этот триггер связан. Триггер срабатывает независимо от того, кто из пользователей и каким способом инициировал событие, вызвавшее запуск триггера. Таким образом, основное назначение триггеров – автоматическая поддержка целостности базы данных.

Очевидно, что чем больше программного кода в виде триггеров и хранимых процедур содержит база данных, тем сложнее ее разработка и дальнейшее сопровождение.

На уровне логического моделирования определяются реляционные отношения и атрибуты этих отношений. На этом уровне можно определять какие-либо физические структуры хранения (индексы, хеширование и т.п.). Единственное, чем можно управлять – это распределение атрибутов по различным отношениям.

Можно описать немного отношений с большим количеством атрибутов, или сформировать большое количество отношений, каждое из которых содержит мало атрибутов. Таким образом, необходимо попытаться ответить на вопрос – влияет ли

количество отношений и количество атрибутов в отношениях на скорость выполнения операций обновления данных. Такая постановка не является достаточно корректной, т.к. скорость выполнения операций с базой данных зависит от физической реализации базы данных. Тем не менее, целесообразно качественно оценить это влияние при одинаковых подходах к физическому моделированию.

Основными операциями, изменяющими состояние базы данных, являются операции вставки, обновления и удаления записей. В базах данных, требующих постоянных изменений (складской учет, системы продаж билетов и т.п.) производительность определяется скоростью выполнения большого количества небольших операций вставки, обновления и удаления.

Обычно, вставка записи производится в одну из свободных страниц памяти, выделенной для данной таблицы. СУБД постоянно хранит информацию о наличии и расположении свободных страниц. Если для таблицы не созданы индексы, то операция вставки выполняется фактически с одинаковой скоростью независимо от размера таблицы и от количества атрибутов в таблице. Если в таблице имеются индексы, то при выполнении операции вставки записи индексы должны быть перестроены. Таким образом, скорость выполнения операции вставки уменьшается при увеличении количества индексов у таблицы и мало зависит от числа строк в таблице.

Для операции обновления и удаления записей из таблицы, прежде, чем обновить или удалить запись, ее необходимо найти. Если таблица не индексирована, то единственным способом поиска является последовательное сканирование таблицы в поиске нужной записи. В этом случае, скорость операций обновления и удаления существенно увеличивается с увеличением количества записей в таблице и не зависит от количества атрибутов. Но на самом деле неиндексированные таблицы практически никогда не используются. Для каждой таблицы обычно объявляется один или несколько индексов, соответствующий потенциальным ключам. При помощи этих индексов поиск записи производится очень быстро и практически не зависит от количества строк и атрибутов в таблице (хотя, конечно, некоторая зависимость имеется). Если для таблицы объявлено несколько индексов, то при выполнении операций обновления и удаления эти индексы должны быть перестроены, на что тратится дополнительное время. Таким образом, скорость выполнения операций обновления и удаления также уменьшается при увеличении количества индексов у таблицы и мало зависит от числа строк в таблице.

Одно из назначений базы данных – предоставление информации пользователям. Информация извлекается из реляционной базы данных при помощи оператора SQL – SELECT. Одной из наиболее дорогостоящих операций при выполнении оператора SELECT является операция соединение таблиц. Таким образом, чем больше взаимосвязанных отношений было создано в ходе логического моделирования, тем больше вероятность того, что при выполнении запросов эти отношения будут соединяться, и, следовательно, тем медленнее будут выполняться запросы. Таким образом, увеличение количества отношений приводит к замедлению выполнения операций выборки данных, особенно, если запросы заранее неизвестны.

8. РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ДАННЫХ

Распределенная обработка данных – методика выполнения прикладных программ группой систем.

Сущность ее заключается в том, что пользователь получает возможность работать с сетевыми службами и прикладными процессами, расположенными в нескольких взаимосвязанных абонентских системах. При этом возможны несколько видов работ, которые он может выполнять: удаленный запрос, например, команда, позволяющая посылать одиночную заявку на выполнение обработки данных; удаленная транзакция, осуществляющая направление группы запросов прикладному процессу; распределенная

транзакция, дающая возможность использования нескольких серверов и прикладных процессов, выполняемых в группе абонентских систем.

Для распределенной обработки осуществляется сегментация прикладных программ. Передача данных происходит при помощи удаленного вызова процедур либо электронной почты. Первая технология характеризуется высоким быстродействием, а вторая – низкой стоимостью. Известны также программные средства Системы Управления Распределенной Базой Данных (СУРБД), содержатся инструментальные средства распределенной среды обработки данных.

Распределенная среда обработки данных – представляет собой технологию распределенной обработки данных.

Эта среда обычно - набор сетевых служб, предназначенный для выполнения прикладных процессов, рассредоточенных по группе абонентских систем гетерогенной сети. Среда чаще всего имеет трехступенчатую архитектуру: прикладная программа – база данных – клиент. Функции, выполняемые средой, включают прикладные службы:

- каталогов, позволяющую клиентам находить нужные им серверы;
- интерфейса многопоточной обработки;
- удаленного вызова процедур;
- обслуживания файлов;
- безопасности данных;
- времени, синхронизирующей часы в абонентских системах.

Программное Обеспечение среды погружается, как правило, в Сетевую Операционную Систему. Серверы могут иметь свои, различные операционные системы. В роли сервера может, также, выступать главный компьютер со своей операционной системой.

Функционирование распределенной среды требует выполнения ряда административных задач. К ним, в первую очередь, относятся средства:

- задачи регистрации и контроля за лицензиями пользователей на работу с прикладными программами;
- унифицированных интерфейсов прикладных программ;
- обеспечения безопасности данных;
- инвентаризации программного и технического обеспечения абонентских систем, работающих в сети.

С точки зрения логического управления среда обработки данных делится на ячейки распределенной среды обработки. В каждую из них может включаться от нескольких единиц до тысяч абонентских систем. Размеры ячеек территориально не ограничены. Входящие в одну и ту же ячейку системы могут быть расположены даже на разных континентах.

9. СТРУКТУРА ПРОГРАММНЫХ МОДУЛЕЙ, РАЗРАБОТКА АЛГОРИТМОВ, ЛОГИЧЕСКИЙ АНАЛИЗ СТРУКТУР ИНТС

Современные технологии разработки программного обеспечения информационных систем предполагают большое разнообразие концепций, подходов и методик. Наиболее популярные из них это структурный и объектный подходы.

Структурный метод, о котором шла речь при разработке информационного обеспечения, при создании программного обеспечения базируется на использовании технологии моделирования потоков данных.

Диаграммы потоков данных (DFD) являются основным средством моделирования функциональных требований к проектируемой информационной системе. С их помощью эти требования представляются в виде иерархии функциональных компонентов (процессов), связанных, потоками данных. Главная цель такого представления – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а

также выявить отношения между этими процессами. Диаграммы потоков данных известны очень давно.

В основе методологии (Gane/Sarson) лежит построение модели анализируемой системы – проектируемой или реально существующей. В соответствии с методологией модель системы определяется как иерархия диаграмм потоков данных (ДПД или DFD), описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю. Диаграммы верхних уровней иерархии (контекстные диаграммы) определяют основные процессы или подсистемы системы с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут такой уровень декомпозиции, на котором процесс становится элементарными и детализировать их далее невозможно.

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 информации. Таким образом, основными компонентами диаграмм потоков данных являются:

- внешние сущности;
- системы/подсистемы;
- процессы;
- накопители данных;
- потоки данных.

Внешняя сущность представляет собой материальный предмет или физическое лицо, представляющее собой источник или приемник информации, например, заказчики, персонал, поставщики, клиенты. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой системы. В процессе анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой системы, если это необходимо, или, наоборот, часть процессов может быть вынесена за пределы диаграммы и представлена как внешняя сущность.

Процесс представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. Физически процесс может быть реализован различными способами: это может быть подразделение организации (отдел), выполняющее обработку входных документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство и т.д.

Номер процесса служит для его идентификации. В поле имени вводится наименование процесса в виде предложения с активным недвусмысленным глаголом в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют существительные в винительном падеже, например:

- "Ввести сведения о клиентах";
- "Выдать информацию о текущих расходах";
- "Проверить кредитоспособность клиента".

Использование таких глаголов, как "обработать", "модернизировать" или "отредактировать" означает, как правило, недостаточно глубокое понимание данного процесса и требует дальнейшего анализа.

Информация в поле физической реализации показывает, какое подразделение организации, программа или аппаратное устройство выполняет данный процесс.

Накопитель данных представляет собой абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.

Накопитель данных в общем случае является прообразом будущей базы данных и описание хранящихся в нем данных должно быть увязано с информационной моделью.

Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока. Каждый поток данных имеет имя, отражающее его содержание.

Первым шагом при построении иерархии ДПД является построение контекстных диаграмм. Обычно при проектировании относительно простых систем строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы. Если же для сложной системы ограничиться единственной контекстной диаграммой, то она будет содержать слишком большое количество источников и приемников информации, которые трудно расположить на листе бумаги нормального формата, и кроме того, единственный главный процесс не раскрывает структуры распределенной системы. Признаками сложности (в смысле контекста) могут быть:

- наличие большого количества внешних сущностей (десять и более);
- распределенная природа системы;
- многофункциональность системы с уже сложившейся или выявленной
- группировкой функций в отдельные подсистемы.

Для сложных систем строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

Разработка контекстных диаграмм решает проблему строгого определения функциональной структуры сис на самой ранней стадии ее проектирования, что особенно важно для сложных многофункциональных систем, в разработке которых участвуют разные организации и коллективы разработчиков.

Для каждой подсистемы, присутствующей на контекстных диаграммах, выполняется ее детализация при помощи ДПД. Каждый процесс на ДПД, в свою очередь, может быть детализирован при помощи ДПД или миниспецификации.

Миниспецификация (описание логики процесса) должна формулировать его основные функции таким образом, чтобы в дальнейшем специалист, выполняющий реализацию проекта, смог выполнить их или разработать соответствующую программу.

При построении иерархии ДПД переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которое описывается при помощи структур данных. Структуры данных конструируются из элементов данных и могут содержать альтернативы, условные вхождения и итерации. Условное вхождение означает, что данный компонент может отсутствовать в структуре. Альтернатива означает, что в структуру может входить один из перечисленных элементов. Итерация означает вхождение любого числа элементов в указанном диапазоне, Для каждого элемента данных может указываться его тип (непрерывные или дискретные данные). Для непрерывных данных может указываться единица измерения (кг, см и т.п.), диапазон значений, точность представления и форма физического кодирования. Для дискретных данных может указываться таблица допустимых значений.

10. РАЗРАБОТКА МОДЕЛЕЙ ЗАЩИТЫ ДАННЫХ

Большое внимание в настоящее время уделяется вопросам формирования принципов построения механизмов защиты информации (ЗИ) и системы требований к ним. На основе имеющегося опыта можно сформулировать следующие фундаментальные принципы организации защиты информации:

- системность;
- специализированность;
- неформальность.

Основные требования принципа системности сводятся к тому, что для обеспечения надежной защиты информации в современных Информационных системах (ИС) должна быть обеспечена надежная и согласованная защита во всех структурных элементах, на всех технологических участках автоматизированной обработки информации и во все время функционирования ИС.

Специализированность как принцип организации защиты предполагает два аспекта:

1) ввиду специфических особенностей рассматриваемой проблемы надежный механизм защиты может быть спроектирован и организован лишь профессиональными специалистами по защите информации,

2) для обеспечения эффективного функционирования механизма защиты в составе ИС должны функционировать специалисты по защите информации.

В соответствии с данным принципом значительное распространение за рубежом получает специализация по различным аспектам ЗИ. В США, например, на вопросах ЗИ специализируется свыше 100 фирм.

Принцип *неформальности* означает, что методология проектирования механизма защиты и обеспечения его функционирования в основе своей является неформальной. Эта неформальность интерпретируется в том смысле, что в настоящее время не существует инженерной методики проектирования механизма защиты в традиционном понимании этого термина.

Общие требования к механизму защиты следующие:

1) адекватность, т.е. обеспечение требуемого уровня защиты (определяется степенью секретности подлежащей обработке информации) при минимальных издержках на создание механизма защиты и обеспечение его функционирования;

2) удобство для пользователей, основу чего составляет требование, чтобы механизм защиты не создавал для пользователей дополнительных трудностей, требующих значительных усилий для их преодоления; минимизация привилегий в доступе, предоставляемых пользователям, т.е. каждому пользователю должны предоставляться только действительно необходимые ему права по обращению к ресурсам системы и данным;

3) полнота контроля, т.е. обязательный контроль всех обращений к защищаемым данным; наказуемость нарушений, причем наиболее распространенной мерой наказания является отказ в доступе к системе;

4) экономичность механизма, т.е. обеспечение минимальности расходов на создание и эксплуатацию механизма;

5) несекретность проектирования, т.е. механизм защиты должен функционировать достаточно эффективно даже в том случае, если его структура и содержание известны злоумышленнику.

В автоматизированных банках данных должно быть предусмотрено наличие в них средств идентификации пользователей и ресурсов системы с периодической сменой идентифицирующей информации, многоаспектного разграничения доступа к элементам баз данных (по элементам, по разрешенным процедурам, по условиям операций и др.), криптографического закрытия данных, регистрации обращений к защищаемым данным, контроля за использованием защищаемых данных и т.д.

Основные положения по разработке систем ЗИ могут быть сформулированы так:

1) защита информации является не разовым мероприятием и даже не совокупностью мероприятий, а непрерывным процессом, который должен протекать (осуществляться) во все время и на всех этапах жизненного цикла ИС;

2) осуществление непрерывного процесса защиты информации возможно лишь на базе промышленного производства средств защиты;

3) создание эффективных механизмов защиты может быть осуществлено высококвалифицированными специалистами-профессионалами в области защиты

информации;

4) поддержание и обеспечение надежного функционирования механизмов защиты информации в ИС сопряжено с решением специфических задач и поэтому может осуществляться лишь профессионально подготовленными специалистами.

Сохранность информации может быть нарушена в двух основных случаях: при получении несанкционированного доступа к информации и нарушении функционирования ЭВМ. Система защиты от этих угроз включает следующие основные элементы: защиту ИС и ее аппаратуры, организационные мероприятия по обеспечению сохранности информации, защиту операционной системы, файлов, терминалов и каналов связи. Следует при этом иметь в виду, что все типы защиты взаимосвязаны и при выполнении своих функций хотя бы одной из них сводит на нет усилия других.

Предлагаемые и реализованные схемы защиты информации в ИС очень разнообразны, что вызвано в основном выбором наиболее удобного и легко осуществимого метода контроля доступа, т.е. изменением функциональных свойств системы.

В качестве классификационного признака для схем защиты можно выбрать их функциональные свойства. На основе и этого признака выделяются системы: без схем защиты; с полной защитой; с единой схемой защиты.

11. АВТОМАТИЗАЦИЯ ПРОЦЕССА ПРОЕКТИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ CASE-ТЕХНОЛОГИЙ

Под термином "CASE-средства" (Computer Aided Software Engineering) понимаются программные средства, поддерживающие процессы создания и сопровождения ИС, включая анализ и формулировку требований, проектирование прикладного ПО и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы. CASE-средства вместе с системным ПО и техническими средствами образуют полную среду разработки ИС.

Появлению CASE-технологии и CASE-средств предшествовали исследования в области методологии программирования. Программирование обрело черты системного подхода с разработкой и внедрением языков высокого уровня, методов структурного и модульного программирования. CASE-технология представляет собой методологию проектирования ИС, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах разработки и сопровождения ИС и разрабатывать приложения в соответствии с информационными потребностями пользователей. Большинство существующих CASE-средств основано на методологиях структурного (в основном) или объектно-ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств.

Наиболее трудоемкими этапами разработки ИС являются этапы анализа и проектирования, в процессе которых CASE-средства обеспечивают качество принимаемых технических решений и подготовку проектной документации.

Графические средства моделирования предметной области позволяют разработчикам в наглядном виде изучать существующую ИС, перестраивать ее в соответствии с поставленными целями и имеющимися ограничениями.

CASE-средства обладают следующими основными особенностями:

а) имеют мощные графические средства для описания и документирования ИС, обеспечивающие удобный интерфейс с разработчиком и развивающие его творческие возможности;

б) осуществляют интеграцию отдельных компонент CASE-средств, обеспечивающую управляемость процессом разработки систем;

в) используют специальным образом организованное хранилище проектных метаданных (репозитория).

Интегрированное CASE-средство должно содержать следующие компоненты:

а) репозиторий, являющийся основой CASE-средства. Он должен обеспечивать хранение версий проекта и его отдельных компонентов, синхронизацию поступления информации от различных разработчиков при групповой разработке, контроль метаданных на полноту и непротиворечивость;

б) графические средства анализа и проектирования, обеспечивающие создание и редактирование иерархически связанных диаграмм (DFD, ERD и др.), образующих модели ИС;

в) средства разработки приложений, включая языки 4GL и генераторы кодов;

г) средства конфигурационного управления;

д) средства документирования;

е) средства тестирования;

ж) средства управления проектом;

з) средства реинжиниринга.

Современный рынок программных средств насчитывает около 300 различных CASE-средств, наиболее мощные из которых используются практически всеми ведущими западными фирмами.

Все современные CASE-средства могут быть классифицированы в основном по типам и категориям. Классификация по типам отражает функциональную их ориентацию на те или иные процессы ЖЦ.

Классификация по категориям определяет степень интегрированности по выполняемым функциям и включает следующее :

а) отдельные локальные средства, решающие небольшие автономные задачи (tools);

б) набор частично интегрированных средств, охватывающих большинство этапов жизненного цикла систем (toolkit);

в) полностью интегрированные средства, поддерживающие весь ЖЦ систем и связанные общим репозиторием.

Помимо этого CASE-средства можно классифицировать по следующим признакам:

а) применяемым методологиям и моделям систем и БД;

б) степени интегрированности с СУБД;

в) доступным платформам.

Классификация по типам в основном совпадает с компонентным составом CASE-средств.

На сегодняшний день российский рынок программного обеспечения располагает следующими наиболее развитыми CASE-средствами: Vantage Team Builder (Westmount I-CASE), Designer/2000, Silverrun, Erwin, Bpwin, S-Designor, CASE-Аналитик, CASE /4/0, PRO-IV, System Architect, Visible Analyst Workbench, EasyCASE.

12. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Текстовая часть оформляется в виде пояснительной записки в соответствии с требованиями стандарта ГОСТ 2.105–95 и ГОСТ 2.106–96.

Общий объем пояснительной записки не должен превышать 50 страниц, в том числе введение – не более 3 страниц. В пояснительной записке приводятся схемы, экранные формы, входные и выходные документы, тексты запросов. Листинг программ выносится в приложение.

Пояснительная записка оформляется на листах формата А4. При оформлении текста используется шрифт Times New Roman 14 пт, интервал полуторный; поля должны иметь следующие размеры: левое – 30 мм, правое – 10 мм, верхнее – 25 мм, нижнее – 20

мм. Титульный лист является первым листом пояснительной записки. Он должен быть оформлен на типовом бланке (прилож. 2).

Нумерация страниц должна быть сквозной, порядковый номер листа ставится в правом верхнем углу. Абзацы в тексте начинают отступом, равным 5 знакам. Содержание основной части пояснительной записки следует делить на разделы, подразделы, пункты.

Иллюстративным материалом для защиты курсовой работы служит плакат формата А1, на который может быть вынесена инфологическая или даталогическая модель данных. Лист графики оформляется в соответствии с требованиями стандарта ГОСТ 2.104 с рамкой и с основной надписью.

Также в качестве иллюстративного материала может быть использована компьютерная презентация, видеоролик и т.п.

13. ПОРЯДОК ЗАЩИТЫ И ОТВЕТСТВЕННОСТЬ СТУДЕНТА ЗА ВЫПОЛНЕНИЕ ЗАДАНИЯ КУРСОВОЙ РАБОТЫ

Пояснительная записка сдается на проверку руководителю курсовой работы в срок не менее чем за 10 дней до защиты. После проверки руководитель либо допускает студента к защите, либо возвращает проект на доработку.

Программная реализация обязательно прилагается на дискете и демонстрируется руководителю. Разработанные студентом реляционные таблицы должны быть заполнены данными, используемыми при выполнении запросов.

Порядок защиты курсовых работ и состав комиссии утверждаются на заседании кафедры. Студент делает доклад, сопровождаемый компьютерной презентацией (около 5 минут), в котором кратко излагает результаты работы и демонстрирует программную реализацию.

Если студент на защите получает оценку «неудовлетворительно», то тема курсового проекта изменяется.

Список литературы

1. Веллинг Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томпсон. — 4-е издание.— М: Вильямс, 2010. — 846с.
2. Измайлов В.Г., Лисьев Г.А., Озерова М.Ю., Трейбач А.Л. Проектирование web-приложений и программных систем в Open Source: учебное пособие. М.: ФЛИНТА, 2011. - 98 с.
3. Колисниченко Д. PHP и MySQL. Разработка веб-приложений. / Д. Колисниченко. — СПб.: БХВ-Петербург: 2013. — 543с.
4. Колисниченко Д.Н. PHP 5/6 и MySQL 6. Разработка web приложений / Д.Н. Колисниченко. – СПб.:БХВ-Петербург, 2010. 560 с.
5. Котеров Д.В. PHP 5. / Д.В. Котеров, А.Ф. Костарев. – СПб: БХВ Петербург, 2006, - 1120с.
6. Кузнецов М. Объектно-ориентированное программирование на PHP /М. Кузнецов, И. Симдянов. – СПб.: БХВ-Петербург, 2007. 608с.
7. Ленгсторф Д. PHP и jQuery для профессионалов / Д. Ленгсторф. — М: Вильямс Ленгсторф: 2011. — 401с.
8. Ловэйн П. PHP 5. Объектно-ориентированное программирование. / П. Ловэйн. – М.:ИТ Пресс, 2007. 224с.
9. Мазуркевич А.М. PHP: настольная книга программиста / А.М. Мазуркевич, Д.С. Еловой . –Минск.: Новое знание, 2003, - 480с.
10. Маклафлин Б. PHP и MySQL. Исчерпывающее руководство / Б. Маклафин — СПб.: Питер: 2013. — 512с.
11. Мишель Е. Д. Изучаем PHP и MySQL / Е. Д. Мишель, А. Ф. Джон. – СПб.: Символ-Плюс, 2008. 442с.
12. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 [Текст] / Р. Никсон. — СПб.: Питер, 2015. — 688 с.
13. Прохоренок Н. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера [Текст] / Н Прохоренок. — СПб.: БХВ-Петербург, 2010. — 912с.
14. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Webмастера. – 4-е изд., перераб. и доп. / Н.А. Прохоренок, В.А. Дронов. – СПб.: БХВ Петербург, 2016. – 768 с.
15. Прохоренок Н. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера [Текст] / Н Прохоренок. — СПб.: БХВ-Петербург, 2010. — 912с.
16. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Webмастера. – 4-е изд., перераб. и доп. / Н.А. Прохоренок, В.А. Дронов. – СПб.: БХВ Петербург, 2016. – 768 с.
17. Прохоренок Н. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера [Текст] / Н Прохоренок. — СПб.: БХВ-Петербург, 2010. — 912с.
18. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Webмастера. – 4-е изд., перераб. и доп. / Н.А. Прохоренок, В.А. Дронов. – СПб.: БХВ Петербург, 2016. – 768 с.

ПРИМЕРНЫЙ СПИСОК ТЕМ КУРСОВЫХ РАБОТ

1.	Создание персонального сайта для каталогизации цифровых фотографий с применением PHP и MySQL.
2.	Создание электронного портала для исследовательской группы в области адаптивной Оптики с применением PHP и MySQL..
3.	Разработка системы с web-интерфейсом для хранения и систематизации электронных публикаций с применением PHP и MySQL..
4.	Разработка веб-инфраструктуры для хранения геоинформационных данных. ("Электронная карта города/страны/планеты") с применением PHP и MySQL.
5.	Разработка веб-сайта для хранения и представления метеорологических данных.
6.	Создание веб-сайта для агентства недвижимости средствами PHP и MySQL.
7.	Создание автоматизированной системы оценки деловых и личностных качеств персонала средствами языка JavaScript средствами PHP и MySQL.
8.	Разработка web-сайта для образовательного учреждения средствами PHP и MySQL.
9.	Разработка web-сайта автомобильной фирмы средствами PHP и MySQL.
10.	Разработка web-сайта автомобильной фирмы средствами PHP и MySQL.

Приложение 2

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение высшего образования
«Дагестанский государственный технический университет»

ЗАДАНИЕ
на курсовую работу по
дисциплине «Web - программирование»

Студенту _ курса группы _____ направление _____

Фамилия, имя, отчество _____

1 Тема _____

2 Исходные данные к проекту _____

3 Содержание пояснительной записки (перечень подлежащих разработке вопросов)

4 Перечень графического материала с точным указанием обязательных чертежей

5 Литература, пособия:

6 Дата выдачи задания «___» _____ 20__ г.

7 Срок сдачи законченной работы «___» _____ 20__ г.

Руководитель работы _____ / _____ /

Задание принял к исполнению _____ / _____ /
«___» _____ 20__ г.

Приложение 3

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное
учреждение высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «**Прикладная математика и информатика**»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

по дисциплине «Web - программирование»
на тему “ТЕМА РАБОТЫ”

Выполнил студент _____
(фамилия и инициалы)

Группа _____

Руководитель _____
(фамилия и инициалы)