

Государственное бюджетное профессиональное образовательное учреждение
«Байконурский электрорадиотехнический техникум имени М.И. Неделина»
(ГБ ПОУ «БЭРТТ»)

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА**

по дисциплине «Web-программирование»
для специальности 09.02.03 (230115)
«Программирование в компьютерных системах»
(базовый уровень)

Байконур
2015

СОДЕРЖАНИЕ

- 1 ОРГАНИЗАЦИЯ КУРСОВОГО ПРОЕКТИРОВАНИЯ
 - 1.1 Назначение, цели и задачи курсового проектирования
 - 1.2 Этапы и сроки выполнения курсового проекта
 - 2 ОБЪЕМ И СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОГО ПРОЕКТА
 - 2.1 Титульный лист
 - 2.2 Задание
 - 2.3 Аннотация
 - 2.4 Содержание
 - 2.5 Введение
 - 2.6 Общая часть
 - 2.7 Требования к ИНТС
 - 2.8 Специальная часть
 - 2.9 Разработка функциональной модели ИНТС
 - 2.10 Исходные данные для проектирования. Стандарты
 - 2.11 Проектирование пользовательского интерфейса
 - 2.12 Комплекс технических средств
 - 2.13 Руководство пользователю
 - 2.14 Руководство по сопровождению
 - 2.15 Структура программных модулей. Разработка алгоритмов. Логический анализ структур ИНТС
 - 2.16 Автоматизация процесса проектирования с использованием Case-технологий
 - 2.17 Заключение
 - 2.18 Список использованных источников
 - 2.19 Вспомогательные указатели
 - 2.20 Приложения
 - 3 ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ДОКУМЕНТА
 - 3.1 Формулы и уравнения
 - 3.2 Использование ссылок
 - 3.3 Оформление иллюстраций
 - 3.4 Оформление таблиц
 - 4 КОНТРОЛЬ КУРСОВОГО ПРОЕКТИРОВАНИЯ
 - 5 ЗАЩИТА КУРСОВОГО ПРОЕКТА
- ПРИЛОЖЕНИЕ А. Пример оформления титульного листа
- ПРИЛОЖЕНИЕ Б. Пример оформления листа задания
- ПРИЛОЖЕНИЕ В. Пример оформления содержания

1 ОРГАНИЗАЦИЯ КУРСОВОГО ПРОЕКТИРОВАНИЯ

1.1 Назначение, цели и задачи курсового проектирования

Курсовое проектирование является промежуточным этапом учебного процесса студентов техникума по специальности 09.02.03 и имеет целью систематизацию, закрепление и расширение теоретических знаний, приобретение необходимых практических навыков в области разработки и размещения Web-приложений.

Курсовое проектирование является завершающим этапом по изучению дисциплины «Web-программирование».

Основными задачами курсового проектирования являются:

- совершенствование профессионального мышления;
- закрепление и углубление знаний в области прикладного программирования, разработки главной страницы и разделов Web-сайта, администрирования сайта, защиты информации на сайте;
- овладение приемами разработки Web-приложений для Internet- систем;
- адаптация Web-приложений к конкретной области применения, а также автоматизированных систем обработки информации и управления;
- отработка умений подготовки программной документации в соответствии с требованиями ЕСПД, ЕСКД;
- умение организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

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.

В результате работы над курсовым проектом студент должен:

иметь представление:

- о способах сбора, обработки, накопления и передачи информации на примере конкретной предметной области;

- о методах и технологии решения задач с использованием ПК;

иметь практический опыт:

- работы с различными системами конструирования сайтов, Web-приложений, средствами адаптации и размещения Web-сайтов в Internet;

- построения концептуальной модели баз данных и разработки структуры баз данных (для коммерческих и информационных Web-приложений);

- использования средств создания и заполнения базы данных;

- использования стандартных методов защиты информации и баз данных в Internet;

уметь:

- создавать объекты баз данных в современных системах управления базами данных и управлять доступом к этим объектам;

- работать с современными case-средствами проектирования баз данных;

- формировать и настраивать схему базы данных;

- применять стандартные методы для защиты объектов базы данных;

знать:

- основные положения теории баз данных, хранилищ данных, баз знаний;

- основные принципы построения концептуальной, логической и физической моделей данных;

- современные инструментальные средства разработки схемы базы данных;

- методы описания схем баз данных в современных системах управления базами данных (СУБД);

- методы организации целостности данных;

- способы контроля доступа к данным и управления привилегиями;

- основные методы и средства защиты данных в базах данных;

- основы разработки приложений баз данных

Итогом курсового проектирования является оценка, которая выставляется руководителем курсового проекта на основании выполнения студентом индивидуального задания. При оценке курсового проекта учитывается качество выполнения пояснительной записки и качество разработанного программного продукта.

Руководителем курсового проекта является преподаватель, который ведет дисциплину «Web-программирование». Во время курсового проектирования руководитель должен:

- определить тему и выдать задание;

- составить совместно со студентами график выполнения проекта и определить сроки окончания и защиты проекта;

- проводить консультации по пунктам проекта;
- проводить проверку законченного проекта;
- проводить прием защиты проектов студентами.

2 ОБЪЕМ И СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОГО ПРОЕКТА

Пояснительная записка курсового проекта представляется в отдельной папке (скоросшивателе) объемом не менее 25-30 страниц машинописного текста, при 28-29 строках на одной странице согласно ГОСТ 2.105-95.

Обязательными элементами курсового проекта являются:

1. титульный лист;
2. задание;
3. аннотация;
4. содержание;
5. введение;
6. общая часть;
7. специальная часть;
8. руководство пользователя;
9. руководство по сопровождению;
10. заключение;
11. список использованных источников.

Дополнительными элементами курсового проекта являются:

12. вспомогательные указатели;
13. приложения.

Ниже представлено описание содержания обязательных элементов.

2.1 Титульный лист

Титульный лист должен содержать:

- полное наименование техникума;
- название темы курсового проекта;
- сведения об исполнителе (Ф.И.О. студента, номер группы);
- сведения о руководителе (Ф.И.О.).

Сокращения слов на титульном листе не допускаются. Пример оформления титульного листа представлен в приложении А.

2.2 Задание

Лист задания должен содержать:

- полное наименование техникума;
- название темы курсового проекта;

- сведения об исполнителе (Ф.И.О. студента, номер группы);
- сведения о руководителе (Ф.И.О.);
- исходные данные (бланки документов, нормативные документы, общие положения, схемы и т.д.);
- состав пояснительной записки;
- содержание графической части;
- дата выдачи задания.

Сокращения слов на листе задания не допускаются. Пример оформления листа задания представлен в приложении Б.

2.3 Аннотация

Аннотация – сокращенное изложение содержания работы с основными фактическими сведениями и выводами. Акцентируется внимание на новых данных, определяется целесообразность работы.

Текст аннотации должен содержать:

- сведения об объеме проекта,
- количество иллюстраций (рисунков), таблиц, приложений, использованных источников,
- перечень ключевых слов,
- объект исследования,
- цель работы,
- метод исследования и полученные результаты, их новизну,
- область применения и внедрения.

Изложение материала в аннотации должно быть кратким и точным. Рекомендуемый объем аннотации 500-1000 печатных знаков (1 лист).

2.4 Содержание

Содержание включает перечень структурных элементов курсового проекта с указанием номеров страниц, с которых начинается их месторасположение в тексте.

Образец оформления содержания представлен в Приложении В.

2.5 Введение

Введение отражает актуальность и социальную значимость рассматриваемой темы, процесс исторического развития и оценку современного состояния описываемой темы, обоснование необходимости проведения работ по данной теме, цель и задачи, методы исследования.

Введение должно содержать:

- обоснование необходимости автоматизированного решения задачи, применения средств вычислительной техники, цель и задачи проекта;

- наименование, краткая характеристика области применения информационной системы и базы данных;
- требования заказчика к разрабатываемой базе данных, программному продукту;
- наименование, краткая характеристика предметной области, для которой предполагается использование продукта, ИНТС, пользователи ИНТС.

Введение может содержать 1 лист текста.

Следует обратить внимание на то, что степень подробности представления сведений о предметной области зависит от того, носит ли решаемая задача традиционный характер в хорошо знакомой всем сфере деятельности или же задача в известной мере нестандартная, область не относится к общеизвестной, имеет ярко выраженную специфику. Чего сегодня уж точно не следует делать, так это предоставлять тексты типа "В настоящее время бурное развитие вычислительной техники привело к ..." или говорить о том, что интерфейс должен быть дружественным. Это, по меньшей мере, наивно, а для тех специальностей, для которых составлены методические рекомендации, подобные высказывания могут свидетельствовать об узости профессионального кругозора.

В случае если решаемая задача является частью более общей задачи или частью большого проекта, необходимо дать краткое описание общей проблемы, представить структуру проекта на функциональном уровне. Для того чтобы выделить подзадачу, пользуются схематическим изображением основных блоков разрабатываемого продукта и выделяют на нем свою функциональную часть.

Во введении необходимо перечислить все разделы курсового проекта с кратким описанием их содержания.

2.6 Общая часть

Общую часть следует делить на главы, параграфы, пункты и подпункты (не менее двух глав). Каждый элемент основной части должен представлять собой законченный в смысловом отношении фрагмент работы.

Общая часть пояснительной записки состоит из следующих элементов:

1 Общая часть

1.1 Общая постановка задачи

1.1.1 Характеристика объекта автоматизации

Содержит задачи и функции, выполняемые сотрудниками, а также перечень регламентирующих документов.

1.1.2 Общие сведения о проектируемой ИНТС

Содержит назначение, задачи и цели создания системы, области применения системы, проектные ограничения разрабатываемой ИНТС, потенциальные пользователи АС, а также перспективы и проблемы продукта.

1.2 Обзор существующих программных продуктов, близких к теме курсового проекта

Анализ должен содержать отличие и преимущество разрабатываемого продукта от уже разработанных аналогичных программ (2-3 примера программных аналогов с описанием их функций, достоинств, недостатков).

2.7 Требования к ИНТС

Данный раздел целесообразно разбить на подразделы: «Требования к системе в целом», «Требования к численности и квалификации персонала», «Требования к надежности», «Требования к эргономике и технической эстетике», «Требования к условиям эксплуатации», «Требования к защите информации», «Требования к задачам ИНТС», «Требования к видам обеспечения».

В подразделе «Требования к системе в целом» необходимо перечислить функции, которые будет выполнять база данных, отразить те требования, которые должны быть предъявлены:

- к составу выполняемых функций;
- к способам организации входных, выходных и нормативно-справочным данным;
- к временным параметрам решаемой задачи.

В подразделе «Требования к численности и квалификации персонала» нужно указать количество человек, которые будут задействованы в обработке данных ИНТС (посменно, одновременно, режим работы), необходимый уровень профессиональной подготовленности персонала (владение навыками работы в определенных программных средствах).

В подразделе «Требования к надежности ИНТС» необходимо указать:

- способы контроля входной информации;
- способы, обеспечивающие надежность выходной информации;
- меры по обеспечению надежного и правильного функционирования базы данных.

В подразделе «Требования к эргономике и технической эстетике» необходимо указать допустимые уровни показателей (яркость фона, критическая частота мелькания, освещенность, условия содержания рабочего места и т.д.) для нормальной работы оператора.

В подразделе «Требования к условиям эксплуатации» необходимо указать:

- вид обслуживания ИНТС;
- число активных пользователей;
- требования к видам и типам носителей информации (входной, выходной, нормативно-справочной);
- допустимые параметры окружающей среды (температура, относительная влажность атмосферы, расстояние между ПЭВМ);

– необходимые требования к особенностям эксплуатации информационной системы (например, наличие определенного пароля с правами доступа к данным; обязательная архивация данных с определенной периодичностью и т.п.)

В подразделе «Требования к защите информации» необходимо указать:

- способы защиты входной информации;
- способы, обеспечивающие защиту выходной информации;
- меры по обеспечению защиты базы данных.

В подразделе «Требования к задачам ИНТС» указываются временные рамки выполнения каждой задачи.

В подразделе «Требования к видам обеспечения» указывают необходимый состав технических средств, применяемых при работе информационной системы. Состав технических средств необходимо привести с указанием основных технических характеристик.

Требования могут быть отражены в виде обоснования необходимости именно данных технических средств.

Например, могут быть указаны требования ИНТС:

- наличие принтера;
- наличие принтера для широкой печати, если ИНТС предполагает использование для печати широкого принтера.

Этот подраздел может составить 1 лист пояснительной записки.

2.8 Специальная часть

Специальную часть следует делить на главы, параграфы, пункты и подпункты. Каждый элемент специальной части должен представлять собой законченный в смысловом отношении фрагмент работы.

2 Специальная часть

2.1 Обоснование выбора программных средств

В этой части осуществляется обзор существующих средств построения ИНТС (не менее 3-х), выбор и обоснование средства, при помощи которой будет реализована ИНТС.

2.2 Проектирование ИНТС

В настоящее время рассматриваются три уровня абстракции для определения структуры данных: концептуальный (точка зрения заказчика), логический (точка зрения разработчика) и физический (точка зрения администратора ИНТС). В соответствии с этим рассматриваются три уровня модели и три шага проектирования.

2.9 Разработка функциональной модели ИНТС

Предпроектная стадия, включающая проведение необходимых исследований по формированию структуры ИНТС и получению 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 СУ 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 определяет 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.

2.10. 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.

2.11 Проектирование пользовательского интерфейса

Проектирования 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.

2.12 Комплекс технических средств

Данная часть должна содержать описание комплекса технических средств использованных при создании программы и минимальный набор технических средств, позволяющих работать с программой.

2.13 Руководство пользователя

Руководство пользователя - содержит подробное описание работы пользователя с приложением. Описание руководства обязательно сопровождается ссылками на рисунки приложений «Макеты экранных форм и отчетов», «Структура меню ИНТС».

2.14 Руководство по сопровождению

Руководство по сопровождению - это описание комплекса мероприятий, связанных с реализацией изменений (модификаций) базы данных информационной системы, и соответствующей документации, вызванных возникшими проблемами или потребностями в модернизации или настройке, с обеспечением поддержки пользователей при эксплуатации базы данных.

2.15 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 системы 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.

2.16 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 ИНТС;
- 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.

2.17 Заключение

Заключение - подводит итог выполненной работы, содержит краткое описание рассмотренных вопросов каждой части пояснительной записки (приводится перечень того, что рассмотрено, изучено, проанализировано, разработано, составлено, выполнено в процессе работы).

2.18 Список использованных источников

Использованные в процессе работы над ИНТС специальные, литературные источники указываются в конце курсового проекта перед приложением. При этом список составляется в алфавитном порядке фамилий авторов или наименований пособий. Список оформляется в соответствии с требованиями ГОСТ 7.1-84, сокращения оформляются по ГОСТ 7.12 и 7.11.

Каждый литературный источник отражается в списке в следующем порядке:

- для многотиражной литературы порядковый номер в списке; фамилия и инициалы автора; название книги (для статьи ее заглавие, название сборника, журнала, его номер); издательство; место и год выпуска;

- для малотиражных материалов название документа (отчета) и его обозначение; наименование министерства (ведомства), института, конструкторского бюро, место и год выпуска.

При ссылке на литературный источник в тексте пояснительной записки приводится порядковый номер используемой литературы, заключенный в квадратные скобки.

2.19 Вспомогательные указатели

Вспомогательные указатели могут содержать:

– список сокращений (оформляется в виде алфавитного перечня принятых в курсовом проекте сокращений и соответствующих им полных обозначений понятий);

– список условных обозначений (оформляется в виде перечня используемых в тексте курсового проекта условных обозначений с соответствующей расшифровкой);

– указатель таблиц и иллюстраций (оформляется в виде перечня названий таблиц или иллюстраций, упорядоченных в соответствии с их порядковыми номерами, с указанием страниц их месторасположения в тексте курсового проекта).

В работе следует применять научно-технические термины, обозначения и определения, установленные соответствующими стандартами, а при их отсутствии - общепринятые в научно-технической литературе. Если сокращения, условные обозначения, символы, единицы и термины повторяются в работе менее трех раз, отдельный список не составляют, а расшифровку дают непосредственно в тексте при первом упоминании.

Если в работе принята специфическая терминология, то должен быть перечень принятых терминов с соответствующими разъяснениями. Перечень включают в содержание документа.

Перечень допускаемых сокращений слов установлен в ГОСТ 2.316.

Вспомогательные указатели приводятся до пункта «Аннотация» пояснительной записки курсового проекта.

2.20 Приложения

Приложения призваны облегчить восприятие содержания курсового проекта и могут включать материалы, дополняющие текст: структура программы, алгоритма работы программы, а так же расчеты, иллюстрации, схемы, листинги программ, бланки, постановления, положения и другие документы в полном объеме или выдержки из них со ссылкой на документ.

При изложении обязательных требований в тексте курсового проекта рекомендуется применять слова «должен», «следует», «необходимо», «требуется, чтобы», «разрешается только», «не допускается», «запрещается», «не следует». При изложении других положений — «могут быть», «как правило», «при необходимости», «может быть», «в случае» и т.д. При этом допускается использовать повествовательную форму изложения текста документа, например «применяют», «указывают» и т.п.

3 Требования к оформлению документа

Текст пояснительной записки оформляется с соблюдением правил:

1. Размер страницы должен соответствовать формату А4 (210x297).
2. Размеры полей: левое – 20 мм, правое, верхнее и нижнее – 10 мм.
3. Шрифт – «Times New Roman», размер – 14, на странице около 1800 знаков, включая пробелы и знаки препинания (т.е. 57 - 60 знаков в строке, 28 - 30 строк на странице).
4. Выравнивание текста - по ширине, красная строка - 1,25 (1,27 мм), отступ слева и справа - 0 мм, запрет висячих строк, междустрочный интервал – полуторный.
5. Заголовки структурных элементов документа и разделов основной части следует располагать в середине строки без точки в конце и печатать прописными буквами, не подчеркивая. Если заголовок включает несколько предложений, их разделяют точками. Переносы слов в заголовках не допускаются.
6. Линии, буквы, цифры и знаки должны быть четкими, одинаково черными по всему тексту.
7. Общая нумерация страниц начинается с титульного листа, но номер страницы пишется, начиная с листа «Содержание».

При оформлении пояснительной записки необходимо придерживаться следующих ГОСТов:

1. ГОСТ 8.417-81 (заменен на ГОСТ 8.417-2002) ГСИ. Единицы физических величин;
2. ГОСТ 7.54–88 Система стандартов по информации, библиотечному и издательскому делу. Представление численных данных о свойствах веществ и материалов в научно-технических документах. Общие требования;

3. ГОСТ 7.9-77 Система стандартов по информации, библиотечному и издательскому делу. Реферат и аннотация;
4. ГОСТ 7.1-84 Система стандартов по информации, библиотечному и издательскому делу. Библиографическое описание документа. Общие требования и правила оформления;
5. ГОСТ 7.11-78 Система стандартов по информации, библиотечному и издательскому делу. Сокращение слов и словосочетаний на иностранных европейских языках в библиографическом описании;
6. ГОСТ 7.12-93 Система стандартов по информации, библиотечному и издательскому делу. Сокращение слов на русском языке. Общие требования и правила;
7. ГОСТ 7.32-91 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления;
8. ГОСТ 2.105-95 Общие требования к текстовым документам.

3.1 Формулы и уравнения

Пояснение значений символов и числовых коэффициентов следует приводить непосредственно под формулой в той же последовательности, в которой они даны в формуле. Значение каждого символа и числового коэффициента следует давать с новой строки. Первую строку пояснения начинают со слова «где» без двоеточия.

Уравнения и формулы следует выделять из текста в отдельную строку. Выше и ниже каждой формулы или уравнения должно быть оставлено не менее одной свободной строки. Если уравнение не уместится в одну строку, оно должно быть перенесено после знака равенства (=) или после знаков плюс (+), минус (-), умножения (x), деления (:), или других математических знаков.

Формулы в тексте следует нумеровать порядковой нумерацией в пределах всей работы арабскими цифрами в круглых скобках в крайнем правом положении на строке. Если приводится только одна формула или уравнение, их не нумеруют.

3.2 Использование ссылок

Ссылаться следует на документ в целом или его разделы. Ссылки на подразделы, пункты, таблицы и иллюстрации не допускаются, за исключением подразделов, пунктов, таблиц и иллюстраций данной работы.

При ссылках на стандарты и технические условия указывают только их обозначение, при этом допускается не указывать год их утверждения при условии записи обозначения с годом утверждения в конце текстового документа под рубрикой «Ссылочные нормативные документы».

Ссылки на использованные источники и литературу в тексте работы заключаются в квадратные скобки, сначала указывают номер источника по списку

использованной литературы, потом, через точку с запятой, номер страницы ([8; 243] или [8; 243,245,289-294]). При перечислении источников каждый из них заключается в квадратные скобки ([8; 243], [11; 31-33], [17; 9]).

3.3 Оформление иллюстраций

Количество иллюстраций должно быть достаточным для пояснения излагаемого текста. Иллюстрации могут быть расположены как по тексту документа (возможно ближе к соответствующим частям текста), так и в конце его. Иллюстрации, за исключением иллюстраций приложений, следует нумеровать арабскими цифрами сквозной нумерацией. Если рисунок один, то он обозначается «Рисунок 1». Допускается нумеровать иллюстрации в пределах раздела. В этом случае номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой. Например – Рисунок 1.1.

При ссылках на иллюстрации следует писать «... в соответствии с рисунком 2» при сквозной нумерации и «... в соответствии с рисунком 1.2» при нумерации в пределах раздела.

Иллюстрации, при необходимости, могут иметь наименование и пояснительные данные (подрисуночный текст). Слово «Рисунок» и наименование помещают под рисунком по центру страницы, например, Рисунок 1 — Блок-схема.

3.4 Оформление таблиц

Таблицы позволяют систематизировать текст, обеспечить наглядность информации. Каждая таблица должна иметь название, точно и кратко отражающее содержание таблицы. Название помещается над таблицей. Заголовок и слово "Таблица" начинают с прописной буквы. Заголовок не подчеркивают. Слово «Таблица» и порядковый номер – над таблицей в правом верхнем углу над названием. Заголовки граф таблиц начинаются с прописных букв, подзаголовки - со строчных (если они составляют одно предложение с заголовком), с прописных (если они самостоятельные). Заголовки таблиц не делятся по диагонали. Высота строк должна быть не менее 8 мм. Граф у "№ п/п" в таблицу включать не следует.

Таблицы в зависимости от их размера располагают после текста, в котором они упоминаются впервые, или на следующей странице, а при необходимости – в приложении. На все таблицы в тексте документа должны быть ссылки. Например – «Таблица 1.1.».

4 Контроль курсового проектирования

Для контроля курсового проектирования студентов составляется график выполнения курсового проекта в виде таблицы (см. таблицу 4.1).

Таблица 4.1

Контроль хода курсового проектирования

№	Ф.И.О.	0 %	5 %	10 %	20 %	30 %	36 %	42 %	47 %	57 %	67 %	72 %	82 %	90 %	95 %	100 %	Оценка
1	Бородач А.Р.																
2	Петров П.А.																
3	Сидоров А.С.																

5 Защита курсовых проектов

Защита курсового проекта осуществляется студентом только после допуска к защите руководителем курсового проекта. Студент не допускается к защите в случае, если замечания, указанные руководителем, не были устранены.

Оценка проекта проводится по пятибалльной системе. Оценка курсового проекта осуществляется по следующим критериям:

- соответствие содержания пояснительной записки созданному программному продукту (задачи, иллюстрации, ссылки на рисунки, тематика и т.д.);
- плановость выполнения проекта во время проектирования,
- соответствие требованиям ГОСТ графической части и пояснительной записки;
- качество защиты курсового проекта.

Для доклада студенту предоставляется 5-10 мин. За это время студент должен:

- изложить поставленную в проекте задачу (цели, целесообразность разработки ИС, конечные пользователи);
- дать краткую характеристику предметной области;
- привести описание каждой части пояснительной записки;
- описать содержание графической части;
- отметить последовательность выполнения работы.

При защите курсового проекта руководитель может задавать вопросы относительно содержания проекта и вопросы, касающиеся дисциплины «Web-программирование».

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

1. Кузнецов М.В., Симдянов И.В., Голышев С.В. РНР 5. Практика разработки Web-сайтов. – СПб: БХВ-Петербург, 2007
2. Кузнецов, М.В., Симдянов, И.В. РНР. Практика создания Web-сайтов., 2-ое издание – СПб: БХВ-Петербург, 2009
3. Основы программирования на РНР: курс лекций: учеб. Пособие для студентов вузов, обучающихся по специальностям в области информ. технологий/ Н.В. Савельева. – М.: Интернет – Ун-т информ. технологий, 2005

Дополнительные источники:

1. Дунаев В. Самоучитель JavaScript – СПб.: Питер, 2005
2. Хольцнер С. РНР в примерах. Пер. с англ. – М.: ООО «Бином-Пресс», 2007

Профильные периодические издания:

1. Webdix.Journal (он-лайн версия) (<http://webdix.livejournal.com/>)

Специализированные порталы:

1. <http://www.intuit.ru>
2. <http://habrahabr.ru/blogs/programming/>
3. <http://phpclub.ru/>
4. <http://www.webscript.ru/>

ПРИЛОЖЕНИЕ А

Пример оформления титульного листа

Государственное бюджетное профессиональное образовательное учреждение
«Байконурский электрорадиотехнический техникум имени М.И. Неделина»
(ГБ ПОУ «БЭРТТ»)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**к курсовому проекту**

По специальности: 09.02.03 «Программирование в компьютерных системах»
(код и наименование специальности)

По дисциплине: Web-программирование

На тему: Разработка Web-сайта подросткового клуба

Выполнил студент: Петров Иван Сергеевич
(Фамилия Имя Отчество)

Курса 4-го группы П-45

К защите курсового проекта допущен: _____ А.Р. Бородач
(подпись) (И.О. Фамилия)

Курсовой проект защищён с оценкой _____

Дата защиты курсового проекта: « _____ » _____ 201__ г.

Руководитель курсового проекта: _____ В.В. Владимиров
(подпись) (И.О. Фамилия)

ПРИЛОЖЕНИЕ Б

Пример оформления листа задания

Государственное бюджетное профессиональное образовательное учреждение
«Байконурский электрорадиотехнический техникум имени М.И. Неделина»
(ГБ ПОУ «БЭРТТ»)

ЗАДАНИЕ

на курсовое проектирование

Студенту _____ Бородач А.Р. _____ курса 4-го группы П-4
(Фамилия И.О.)

По специальности: 09.02.03 «Программирование в компьютерных системах»
(код и наименование специальности)

По дисциплине: Web-программирование

Тема задания: Разработка Web-сайта подросткового клуба

Исходные данные: нормативные документы по учету оформления заграничных паспортов, общие положения по деятельности группы паспортизации, схема порядка оформления документов для заграничных паспортов.

В процессе работы над курсовым проектом должно быть выполнено следующее:

I. Пояснительная записка

Аннотация

Содержание

Введение;

Общая часть

Специальная часть

Руководство пользователя

Руководство по сопровождению

Заключение

Список использованных источников

II. Графическая часть

Лист 1. Инфологическая модель данных

Лист 2. Концептуальная схема АИС группы паспортизации войсковой части

Лист 3. Структура интерфейса пользователя

Лист 3. Макеты экранных форм и отчетов

Дата выдачи задания: «__» _____ 201__ г.

Руководитель курсового проекта: _____ В.В. Владимиров
(подпись) (И.О. Фамилия)

ПРИЛОЖЕНИЕ В

Пример оформления содержания

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ОБЩАЯ ЧАСТЬ.....	6
1.1 Общая постановка задачи	6
1.1.1 Характеристики объекта автоматизации.....	6
1.1.2 Общие сведения о проектируемой ИНТС	7
1.2 Анализ существующих программных средств, реализующих постановку задачи	10
1.3 Требования к ИНТС	12
2 СПЕЦИАЛЬНАЯ ЧАСТЬ.....	16
2.1 Обоснование выбора программных средств	16
2.2 Проектирование ИНТС.....	18
2.2.1 Построение информационной структуры предметной области	20
2.2.2 Построение логической структуры ИНТС	20
2.3 Дополнительные программные средства	21
2.4 Разработка ИНТС.....	22
2.5 Комплекс технических средств	22
3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	23
4 РУКОВОДСТВО ПО СОПРОВОЖДЕНИЮ.....	24
ЗАКЛЮЧЕНИЕ	25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	27
ПРИЛОЖЕНИЕ А. Структура разделов ИНТС	29
ПРИЛОЖЕНИЕ Б. Структура главной страницы.....	30
ПРИЛОЖЕНИЕ В. Макеты страниц разделов сайта	31

					<i>КП.09.02.03.35.2015.ПЗ</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>				
<i>Разраб.</i>		<i>Ф.И.О.</i>			<i>Название работы</i>	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Провер.</i>		<i>Ф.И.О.</i>					30	1
<i>Реценз.</i>		<i>Ф.И.О.</i>						
<i>Н. Контр.</i>		<i>Ф.И.О.</i>						
<i>Утверд.</i>		<i>Ф.И.О.</i>						

