

Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М.А.Бонч-Бруевича

Кафедра «Безопасность информационных систем»

РАЗРАБОТКА ЗАЩИЩЕННЫХ ВЕБ- ПРИЛОЖЕНИЙ

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
ПО КУРСОВОЙ РАБОТЕ

Санкт-Петербург
2018

В методических рекомендациях изложены основные требования, порядок выполнения и правила оформления курсовой работы. Рекомендации предназначены для студентов, изучающих дисциплину «Разработка защищенных веб-приложений». Рекомендации могут также оказать помощь при дипломном проектировании и оформлении конкурсных работ.

СОДЕРЖАНИЕ

1 Цель и задачи курсовой работы	4
2 Методические указания по выполнению задания на курсовую работу	4
3. Требования к содержанию и форме пояснительной записи.....	6
4 Защита курсовой работы.....	9
5 Методические рекомендации по оцениванию курсовой работы.....	9
Приложение А.....	10

1 Цель и задачи курсовой работы

Курсовая работа имеет целью научить студентов применять полученные знания для самостоятельного решения на конкретных практических задачах.

На кафедре БИС выполняется курсовая работа по дисциплине «Разработка защищенных веб-приложений», которая направлена на закрепление и углубление теоретических знаний и практических навыков решения задач связанных с разработкой защищенных веб-приложений и закрепления навыков в оформлении курсовой работы.

В процессе выполнения курсовой работы студенты должны:

- закрепить, углубить и обобщить знания, полученные при изучении дисциплины;
- закрепить навыки разработки веб-приложений, а также в отладке программ;
- приобрести навыки самостоятельной работы с учебной и справочной литературой;
- получить навыки в разработке календарного плана выполнения курсовой работы и правильного оформления пояснительной записи.

Полученные в ходе курсовой работы знания являются базовыми для выполнения последующих курсовых работ, а также для разработки дипломного проекта.

2 Методические указания по выполнению задания на курсовую работу

Выполнение задания целесообразно разбить на ряд относительно самостоятельных этапов, которые вытекают из общей последовательности решения задачи. К этим этапам относятся:

- уяснение задания курсовой работы;
- формализация задачи и разработка структуры информационной системы (ИС);
- разработка структуры базы данных (БД)
- разработка веб-интерфейса ввода и вывода информации в БД;
- разработка модулей обработки и преобразования данных;
- формирование элементов информационной защиты данных веб-приложения
- отладка программного кода ИС.

Начальным этапом курсовой работы является уяснение задания. Опыт показывает, что для того, чтобы минимизировать количество ошибок, необходимо вникнуть в содержание задачи, то есть четко уяснить себе основные функции, которые надо реализовать, исходные данные для работы программы, накладываемые ограничения и так далее. На основе этого составляется ориентировочный план работы.

Этап формализации задачи и разработки структуры и программного кода включает в себя следующие виды работ:

- изучение необходимой литературы;
- выбор методов решения поставленной задачи;
- выбор рациональной схемы подходящего способа поиска, упорядочения, сортировки в информационно-поисковых задачах;
- разработку алгоритма решения задачи.

После выбора метода решения задачи и форм организации данных, разработка структуры базы данных является важнейшей задачей, так как является фундаментом всей информационной системы. Алгоритм взаимодействия отдельных программных модулей полностью определяет структуру информационной системы, позволяет облегчить ее реализацию, предотвратить возникновение ошибок и, следовательно, сократить время отладки.

В основу разработки алгоритма целесообразно положить метод пошаговой детализации. Суть метода состоит в следующем. Разработка алгоритма производится по принципу

«от общего к частному, сверху вниз». На первом шаге детализации обычно разрабатывается укрупненная схема алгоритма, в которой хорошо просматривается основная идея выбранного метода решения задачи. Она является как бы генеральным планом и обычно имеет вид простой последовательности. На втором и последующих шагах раскрываются отдельные функциональные блоки путем замены их стандартными блоками. Степень детализации зависит от опыта программиста и языка программирования, на котором будет реализован алгоритм. При разработке алгоритма необходимо выделять одинаковые повторяющиеся участки и по возможности оформлять их в виде подпрограмм. Чтобы облегчить переход от алгоритма к программе и обратно, желательно использовать в схеме алгоритма только те обозначения, которые допустимы в программе. Разработанный алгоритм представляется в виде схем.

Этап разработки программного кода ИС. Программные модули реализуются путем записи алгоритма решения задачи на языке программирования соответствующего требованием решаемой задачи. При этом необходимо стремиться, чтобы программные модули получались, наглядными и читаемыми. Чем проще написаны программы, чем лучше они оформлена, тем удобнее такие программы в отладке и эксплуатации. Удобочитаемость программы связана со стилем программирования. Поясним основные требования хорошего стиля программирования.

Первое требование. Программа должна содержать комментарии. Комментарии нужны как на стадиях проектирования и отладки программы, так и позже. Можно выделить комментарии вводные и пояснительные. Вводные комментарии должны помещаться в начале каждой программы или подпрограммы (функции) и пояснить их назначение. Пояснительные комментарии сопровождают наиболее трудно описываемые части программы. Средней нормой можно считать одну строку комментариев на десять строк программы.

Второе требование хорошего стиля программирования заключается в пропуске строк, что значительно повышает наглядность программы. Как в естественном языке мы пользуемся пропусками строк для отделения разделов, так и в программе с помощью пропуска строк можно разделять ее отдельные фрагменты. Например, пропуском одной строки можно отделять группу логически связанных операторов, пропуском двух строк – основные логически связанные фрагменты программы. Использование незаполненных строк облегчает поиск отдельных частей в программе.

Третье требование заключается в рациональном выборе имен переменных так, чтобы они наилучшим образом характеризовали те величины, которые они представляют. Все обозначения объектов в алгоритме и программе должны по возможности совпадать.

Четвертое требование касается размещения операторов. В большинстве языков программирования высокого уровня допускается размещение нескольких операторов на одной строке. Однако запись нескольких операторов на одной строке затрудняет чтение программы. Кроме того, по сообщению транслятора о синтаксической ошибке легче обнаружить эту ошибку при размещении в одной строке одного оператора.

Оформление программного кода ИС

Пятое требование. Для указания связи операторов, между ними делают одинаковые отступы от начала строки. Отступы, не оказывая влияния на логику программы, существенно улучшают ее читаемость. Правильно сделанные отступы хорошо подчеркивают структуру программы.

Этап тестирования и отладки программы. Этот этап является наиболее сложным этапом, так как общих и одновременно эффективных методов поиска ошибок не существует.

Все ошибки, которые возникают в процессе работы можно разделить на две большие группы: алгоритмические и программные. *Алгоритмические* ошибки приводят к неправильному или неэффективному решению задачи и обусловлены либо некорректной постановкой задачи, либо непониманием ее сути. *Программные* ошибки возникают из-за недостаточного знания программистом используемых языков программирования. Кроме алгоритмических и программных ошибок могут также возникать ошибки, вызываемые невнимательными действиями программиста в процессе набора кода.

При тестировании и отладке программного кода выполняются следующие виды работ:

- а) Подготовка контрольных вариантов для проверки правильности реакции информационной системы на произведенные внешние действия.
- б) Трансляция и исправление обнаруженных транслятором синтаксических и логических ошибок. Исправление выявленных ошибок представляет собой сложный этап отладки программы. Следует отметить, что на этапе отладки выявляются не все ошибки. Ошибки логического характера транслятор обнаружить не может.

При наличии большого количества и разнообразия синтаксических ошибок невозможно дать конкретные рекомендации по поводу способов их устранения. Однако можно указать следующий подход. Если в программных модулях выявлено очень большое количество ошибок, следует просто исправить понятные и очевидные ошибки и повторить трансляцию. Некоторые сообщения о синтаксических ошибках являются на самом деле ложными, так как вызваны другими ошибками. Поэтому не стоит тратить много времени на то, чтобы разобраться с каждым выданным сообщением, если характер ошибки не очевиден сразу. Ложных сообщений об ошибках особенно много тогда, когда в программе имеет место какое-то серьезное упущение, например, неправильное написание начала главной функции, отсутствие описания массива и так далее.

3. Требования к содержанию и форме пояснительной записи

Пояснительная записка по своему содержанию должна соответствовать заданию на курсовую работу. В структурном плане пояснительная записка должна включать в себя следующие последовательно размещённые части :

- титульный лист;
- аннотацию;
- содержание (оглавление);
- основную часть;
- заключение;
- список использованной литературы;
- приложения.

Титульный лист выполняется стандартным образом.

В **аннотации** кратко излагается содержание проделанной работы и краткие выводы по полученным результатам. Объём аннотации- до одной страницы. Аннотация оформляется по правилам основной части, описание которой будет приведено ниже.

Содержание пояснительной записи включает номера и наименования разделов и подразделов основной части пояснительной записи с указанием номера листа, на котором размещается их начало. В содержание включают также Заключение, Список использованной литературы и Приложения. Пункты и подпункты в содержание пояснительной записи не включают.

Слово «Содержание» записывают в виде заголовка (симметрично тексту) с прописной буквы без точки в конце. Все номера и наименования, включённые в содержание, записывают строчными буквами, начиная с прописной буквы.

Основная часть содержит материал, подлежащий обязательному представлению, т.е. последовательно излагает содержание работы, выполненной в соответствии с заданием:

- 1 Условие задачи и способ ее решения.
- 2 Формализация задачи.
- 3 Схемы алгоритмов.
- 4 Текст программного кода.
- 5 Руководство пользователя.

Основная часть оформляется по общим правилам оформления пояснительной записи в соответствии с требованиями ГОСТ 2.105-95 «Общие требования к текстовым документам».

Текст пояснительной записи следует выполнять шрифтом Times New Roman, размер 14пт, интервал полуторный. Пояснительная записка оформляется на одной стороне стандартных листов бумаги формата А4 (297*210 мм). Поля для текста: левое – 25 мм; верхнее и нижнее -20 мм; правое – 15 мм.

Текст пояснительной записи при необходимости разделяют на разделы и подразделы, снабжая каждый номером и заголовком.

Разделы должны иметь порядковые номера в пределах всего документа, обозначенные арабскими цифрами без точки и записанные с отступом, равного пяти символам или 15-17 мм. Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номеров раздела и подраздела, разделённых точкой. В конце номера подраздела точка не ставится, например: 1.1, 1.2 и т.д. Разделы, как и подразделы, могут состоять из одного или нескольких пунктов. Если документ не имеет подразделов, то нумерация пунктов в нём должна быть в пределах каждого раздела, и номер пункта должен состоять из номеров раздела и пункта, разделённых точкой. В конце номера пункта точка не ставится, например: 1.1, 1.2, 1.3 и т.д. Если документ имеет подразделы, то нумерация пунктов должна быть в пределах подраздела и номер пункта должен состоять из номеров раздела, подраздела и пункта, разделённых точками, например: 1.2.1, 1.2.2, 1.2.3 и т.д.

Пункты, при необходимости, могут быть разбиты на подпункты, которые должны иметь порядковую нумерацию в пределах каждого пункта, например: 4.2.1.1, 4.2.1.2, 4.2.1.3 и т.д. Внутри пунктов или подпунктов могут быть приведены перечисления. Перед каждой позицией перечисления следует ставить дефис или при необходимости ссылки в тексте документа на одно из перечислений, строчную букву, после которой ставится скобка. Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, а запись производится с абзацного отступа, как показано в примере.

Пример.

- а) _____
 б) _____
 1) _____
 2) _____
 в) _____

Разделы и подразделы должны иметь заголовки. Пункты, как правило, заголовков не имеют. Заголовки следует писать с прописной буквы, без точки в конце, не подчёркивая. Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой.

Для пояснения текстового материала в пояснительной записке могут использоваться *иллюстрации*, которые независимо от вида и содержания (схемы, алгоритмы и др.) следует называть *рисунками*. Рисунки могут быть расположены как по тексту пояснительной записи, так и в конце её или даны в приложении. Рисунки, за исключением рисунков приложений, следует нумеровать арабскими цифрами сквозной нумерацией. Если рисунок один, то он обозначается «Рисунок 1». Рисунки приложений обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. Например – Рисунок А.3. Допускается нумеровать рисунки в пределах раздела. В этом случае номер рисунка состоит из номера раздела и порядкового номера рисунка, разделённых точкой. Например – Рисунок 1.1.

При ссылках на рисунки следует писать «...в соответствии с рисунком 2» при сквозной нумерации, а при нумерации в пределах раздела «...в соответствии с рисунком 1.2».

Рисунки, при необходимости, могут иметь наименование и пояснительные данные (подрисуночный текст). Слово «Рисунок» и наименование помещают после пояснительных данных и располагают следующим образом: Рисунок 1 – Структура базы данных.

Текст программного кода разрабатывается также в соответствии с методическими рекомендациями данного раздела и оформляется в виде отдельного раздела.

Руководство пользователя оформляется в виде отдельного раздела. Содержит всю информацию для эффективного использования разработанной информационной системы, в том числе:

- общие сведения об информационной системе, т.е. перечисление и краткие характеристики программного обеспечения, необходимого для функционирования разработанной системы;
- назначение веб-приложения и сведения об ограничениях на применение, если такие есть;
- характеристика технических средств, необходимых для работы веб-приложения;
- сообщения пользователю (оператору), выдаваемые при работе веб-приложения;

Нумерация листов. Листы нумеруют арабскими цифрами. Номер листа проставляют в правом нижнем углу. Титульный лист и аннотацию включают в общую нумерацию, но номера листов на них не ставят.

Заключение должно содержать конкретные выводы по результатам работы. Примерный перечень пунктов выводов имеет следующий вид:

- особенности веб-приложения (как осуществлена проверка вводимых данных, как реализован дружественный интерфейс пользователя, как применялись комментарии);
- какие разделы пояснительной записи оказались сложными и почему;
- техническая характеристика разработанной веб-приложения (языки программирования, версии, среда программирования, объём кода и т.д.);

Заключение оформляется по правилам основной части. Заключение разделом не считается и не нумеруется. Слово «ЗАКЛЮЧЕНИЕ» записывают в виде заголовка прописными буквами симметрично тексту.

Список использованной литературы. В пояснительной записке должны быть сделаны ссылки на использованную литературу. Ссылка на литературный источник производится указанием в тексте порядкового номера источника из списка литературы, заключенного в квадратные скобки. Ссылки на номер и страницу источника также заключаются в квадратные скобки, например:[15, с.124 -128]. Список использованной литературы должен содержать только те источники, которые использовались при разработке курсовой работы и на которые есть ссылки в тексте пояснительной записи. Построение списка производится в порядке упоминания литературных источников в тексте пояснительной записи. Нумерация литературных источников в списке производится арабскими цифрами с точкой. После номера приводится краткое библиографическое описание источника. Пример выполнения списка литературы имеет следующий вид:

Литература

1. Красов, А. В. Разработка защищенных приложений [Электронный ресурс] : учебное пособие / А. В. Красов, А. Ю. Цветков ; рец. С. Е. Душин ; Федеральное агентство связи, Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования "Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича". - СПб. : СПбГУТ, 2013. - 82 с. : ил. - 119.87 р.
2. Руби, Сэм Rails 4. Гибкая разработка веб-приложений [Электронный ресурс] / Сэм Руби, Дэвид Томас, Дэвид Хейнмейер Хэнсон. - СПб. : Питер, 2014. - 448 с. : ил. - ISBN 978-5-496-00898-3 : Б. ц.

3. Байдачный, С. С. NET Framework 2.0. Секреты создания Windows-приложений [Электронный ресурс] : учебное пособие / Байдачный С. С. - Москва : СОЛООНПРЕСС, 2008. - 520 с. - ISBN 5-98003-245-2 : Б. ц. Книга находится в Премиумверсии ЭБС IPRbooks.

Приложения. Материал, дополняющий текст документа, допускается помещать в приложениях. Приложениями могут быть, например, экранные формы интерфейсов разработанного веб-приложения, графический материал, образцы оформления, таблицы и т.д.

Приложение оформляют как продолжение пояснительной записи на последующих её листах или выпускают в виде самостоятельного документа. В пояснительной записи на все приложения должны быть даны ссылки.

Каждое приложение следует начинать с новой страницы с указанием наверху посередине страницы слова «Приложение» и его обозначения. Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ъ. После слова «Приложение» следует буква, обозначающая его последовательность. Если в документе одно приложение, оно обозначается «Приложение А».

Приложения должны иметь общую с остальной частью пояснительной записи сквозную нумерацию страниц. Все приложения должны быть перечислены в содержании пояснительной записи.

4 Защита курсовой работы

Защита курсовой работы проводится в случае, если работа допущена к защите, с целью формирования окончательной оценки за работу студента.

Допускается выставление оценки по результатам выполненной курсовой работы без ее защиты на последнем занятии при условии, что работа выполнена досрочно и оформлена с высоким качеством.

При получении неудовлетворительной оценки студенты выполняют работу по новой теме или перерабатывают прежнюю в установленные сроки.

5 Методические рекомендации по оцениванию курсовой работы

Итоговая оценка за курсовую работу складывается из двух частных оценок – оценки за качество выполненной работы и оценки за качество её защиты.

Качество выполненной работы оценивается на основании следующих показателей:

- Степень самостоятельности при выполнении работы.
- Оптимальность структуры ИС и правильность его оформления.
- Грамотность программной реализации (проверка вводимых данных, «дружественность» интерфейса, применение комментариев).
- Правильность и качество оформления пояснительной записи, грамотность изложения материала.

Качество защиты курсовой работы оценивается на основании следующих показателей:

- Форма доклада, т.е. степень свободы изложения материала и корректность использования понятий и терминов.
- Степень адекватности восприятия и правильность ответов на поставленные вопросы.

Пример титульного листа

**Санкт-Петербургский государственный университет телекоммуникаций
им.проф. М.А.Бонч-Бруевича
Факультет информационных систем и технологий**

**Курсовая работа
по дисциплине
«Разработка защищенных веб-приложений»**

Выполнил
студ.3 курса, гр. ИКС-621
направление подготовки 230400
_____ Иванов А.А.
«___»____ 2018

Принял
преподаватель каф..БИС

_____ оценка
_____ Петров А.А.
«___»____ 2018

Санкт-Петербург
2018