

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 08.09.2021 16:32:07
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования

«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
(ЮЗГУ) 2017г.



ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ОТЛАДКА ДО- КУМЕНТИРОВАНИЯ ПРОГРАММ СРЕДНЕЙ СЛОЖНОСТИ

Методические указания по выполнению курсового проекта по дис-
циплине «Основы реверсинжиниринга программных средств»,
«Методы защиты программного обеспечения» для студентов спе-
циальности 10.03.01

Курск 2017

УДК 004.725.7

Составители: А.Л. Марухленко

Рецензент

Кандидат технических наук, доцент *И.В. Калуцкий*

Проектирование, разработка и отладка документирования программ средней сложности : методические указания к выполнению курсового проекта по дисциплинам », «Методы защиты программного обеспечения»/ Юго-Зап. гос. ун-т; сост. А. Л. Марухленко Курск, 2017. - 19с.

Содержат сведения по вопросам проектирования, разработки и отладки программ. Указывается порядок выполнения курсового проекта, правила оформления.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по направлению подготовки «Информационная безопасность».

Предназначены для студентов направления подготовки 10.03.01.

Текст печатается в авторской редакции

Подписано в печать 01.11.2017. Формат 60x84 1/16.

Усл.печ. л. 1,1. Уч.-изд.л. 1,0. Тираж 30 экз. Заказ _____. Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

Введение.....	4
Цель курсового проекта.....	5
1. Методика работы над курсовым проектом.	5
2. Объем и содержание пояснительной записки.	11
3. Оформление курсового проекта.	12
Библиографический список.	13
Приложение 1	13

ВВЕДЕНИЕ

Программное обеспечение (ПО) является одной из основных составляющих любой современной ИС: отдельного ПК, вычислительных и телекоммуникационных сетей различного размера и назначения, от небольших локальных до глобальных.

В последнее время резко возрос интерес к программированию. Это связано с развитием и внедрением в повседневную жизнь информационно-коммуникационных технологий. Если человек имеет дело с компьютером, то рано или поздно у него возникает желание, а иногда и необходимость программировать.

Разработка любой программы или программной системы начинается с определения требований к ней для конкретного набора пользователей и заканчивается эксплуатацией системы этими пользователями.

Существуют различные подходы и технологии разработки алгоритмов и программ. Хотя программирование в значительной степени искусство, тем не менее, можно систематизировать и обобщить накопленный профессиональный опыт.

ЦЕЛЬ КУРСОВОГО ПРОЕКТА

Приобретение студентами практических навыков проектирования, разработки, отладки и документирования программ средней сложности или небольших программных систем.

1. Методика работы над курсовым проектом

1.1. Основные этапы и сроки выполнения курсового проекта

Процесс создания программного изделия, а также назначение и содержание подготавливаемых на каждом этапе документов регламентируются комплексом стандартов, объединенных под общим названием: Единая Система Программированной Документации (ЕСПД). Основными стадиями, согласно ГОСТ 19.102-77, являются:

- разработка технического задания (ТЗ),
- разработка технического проекта (ТП),
- разработка рабочего проекта (РП),
- внедрение.

В связи с учебным характером выполняемой работы и необходимостью более глубокого рассмотрения таких вопросов, как умение писать структурные программы на неструктурном языке, изучение методики и техники систематического тестирования программы, своевременного ее документирования, ознакомление с инженерными и организационными аспектами программирования, современными технологиями программирования, содержание этапов выполнения курсового проекта и подготавливаемой документации несколько отличается от указанных в ЕСПД. Названия основных этапов, сроки выполнения представлены в таблице 1.

Таблица 1 – Основные этапы и сроки выполнения

№	Название этапа	% от общего объе-	Сроки выпол-
---	----------------	-------------------	--------------

		ма работы	нения
1	Постановка задачи	5%	1неделя
2	Составление проекта программы и плана отладки	10%	2недели
3	Алгоритмизация. Выбор основных средств отладки	20%	4недели
4	Программирование. Подготовка тестов	10%., 10%	3недели
5	Отладка программы	40%	6недель
6	Оформление результатов. Сдача отчета	5%	1неделя

1.2. Содержание основных этапов

1.2.1. Постановка задачи

Задания приведены в приложении 1. Данные темы работ могут изменяться в зависимости от выполняемых ведущей кафедрой хоздоговорных и госбюджетных НИР.

Студент, выполняющий курсовой проект, должен представить в письменном виде постановку задачи (данный этап аналогичен разработке технического задания ГОСТ 19.201-78 ЕСПД).

Постановка задачи должна содержать:

- содержательную и, по возможности, математическую формулировку решаемой задачи;
- необходимые характеристики разрабатываемой программы;
- требования к взаимодействию с программой;

- описание входных и выходных данных;
- описание возможной реакции программы на отклонения в структуре исходных данных, а также наличие ошибок в данных и реакцию на сбои ЭВМ;
- способ проверки готовой продукции.

Письменно сформулированное задание будет препятствовать подмене или отходу в процессе разработки программы от ее оговоренных требований, сделает возможным обсуждение, оценку или согласованную с руководителем работы корректировку отдельных требований в ходе разработки программы, а также позволит избежать таких ошибок и противоречий, которые могут быть обнаружены только после разработки большей части программы или уже на стадии анализа полученных результатов выполнения программы.

1.2.2. Составление проекта программы и плана отладки

На основании анализа формализованной постановки задачи выбирается основной метод ее решения (если он не оговорен специально на предыдущем этапе) и составляется общий проект программы. Выбранный подход к решению задачи должен обеспечивать правильные результаты для тех условий функционирования программы, которые предусмотрены постановкой задачи, удобство использования программы, простоту подготовки исходных данных и т.п.

Помимо формулировки выбранного метода решения задачи в проекте должны описываться основные части разрабатываемой программы, их функции, взаимосвязь и последовательность выполнения, а также точно определяться входные и выходные данные как всей программы, так и ее частей. На данном этапе разрабатывается и проект инструкции для пользователей, в которой фиксируется (и, таким образом, может быть заранее оценен и исправлен) предполагаемый режим обращения пользователя (и оператора) с про-

граммой.

С целью избежания появления в разрабатываемой программе ошибок следует разработать на данном этапе и основную стратегию отладки, общий план ее проведения, необходимо решить, каким путем можно будет в дальнейшем убедиться, что выбранный метод решения задачи и разработанная программа обеспечивают получение правильных результатов для указанных в постановке задачи диапазонов изменения величин и условий функционирования программы. Для этого разрабатываются общая методика проверки правильности программы и система необходимых для отладки контрольных примеров.

1.2.3. Алгоритмизация

Данный этап может пропускаться или объединяться с предыдущим в случае простых программ, но в случае достаточно сложных программ качественное выполнение этапа алгоритмизации может существенно облегчить выполнение следующего за ним этапа программирования, особенно в случаях, когда программист - разработчик недостаточно освоил язык, на котором предстоит программировать.

На данном этапе с помощью блок-схем или других специальных языков отображения алгоритмов (таблица решений, НПРО - схем и т.д.) представляются общая структура программы и алгоритмы выполнения отдельных частей (блоков) программы.

Для достаточно сложных программ алгоритмизация проводится в несколько шагов с целью постепенной детализации алгоритма. Критерием окончания детализации некоторого блока (блоков) алгоритма является формирование у программиста мысленного представления о том, как данный блок алгоритма может быть реализован средствами данного языка программирования. Рекомендуется размещать каждую блок - схему на одном стан-

дартном листе бумаги.

После каждого шага детализации проводится проверка полученного алгоритма для выявления возможных ошибок. При этом, возможно, придется изменять и уточнять решения, принятые на этом и предыдущем этапах.

Все такие изменения обязательно должны вноситься в проект программы.

На этапе алгоритмизации следует предусмотреть способы контроля отдельных блоков и методы предстоящей локализации возможных ошибок. Для этого следует подготовить контрольные примеры, в блок - схемах алгоритмов наметить места, моменты и объемы необходимой отладочной печати, которые должны обеспечить возможность быстрой локализации ошибок при отладке.

1.2.4. Программирование

Программирование по разработанным блок - схемам алгоритмов представляет собой традиционный этап в создании прикладных программ. Чем выше качество подготовленных блок - схем, тем более легким будет составление программы. Основные трудности и, следовательно причины ошибок на этом этапе заключаются, во - первых, в необходимости знания всех требований и ограничений выбранного языка программирования и, во - вторых, в необходимости постоянного внимания ко многим деталям языка, которые приходится учитывать в ходе написания программы. Кроме того, вследствие возможных ошибок, на этапе алгоритмизации, на этапе программирования наряду с реализацией системы осуществляется логический контроль правильности ранее принятых решений. При обнаружении ошибок такого рода следует внести соответствующие изменения в алгоритм, поскольку к нему придется обращаться на следующих этапах, и тексты алгоритма и программы должны соответствовать друг другу. Изменяя некоторые части алгоритма,

всегда следует проверять не окажут ли внесенные коррективы влияния на выполнение других блоков и частей.

После составления программы проводится ее проверка для обнаружения и исправления ошибок, внесенных на данном этапе. Следует помнить об обязательной реализации плана и средств отладки, намеченных на предыдущих этапах.

Чем более тщательно выполнены этапы 2, 3 и 4, и, в частности, чем более детально разработан план отладки, чем больше внимания было уделено проверке составленного алгоритма и программы, а также применению отладочных средств для облегчения предстоящей отладки, тем меньше времени потребуется на проведение самой отладки и получение корректных результатов.

1.2.5. Отладка программы

На этом этапе производится обнаружение с помощью ЭВМ ошибок в программе и их исправление. Здесь можно выделить следующие этапы:

- Устранение синтаксических ошибок, обнаруженных транслятором.
- Контроль правильности выполнения программы путем пропуска на ЭВМ подготовленных на предыдущих этапах специальных контрольных примеров (тестов) и сравнения промежуточных и окончательных результатов работы программы с полученными при подготовке тестов. Таким способом определяются содержательные (семантические) ошибки, которые не проявляются при трансляции программы.
- Локализация ошибок, т.е. точное определение места, где в программе допущена ошибка, выявленная на предыдущем этапе.
- Исправление ошибок. Исправления вносятся как в программу, так и в алгоритм.

2. ОБЪЕМ И СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Пояснительная записка объемом в 30 - 35 листов рукописного текста с иллюстрациями по курсовому проекту должна содержать:

- 1) содержательную и формализованную постановку задачи;
- 2) обоснование выбранного метода решения задачи и проекта системы;
- 3) блок - схемы алгоритмов для каждого уровня детализации с необходимыми пояснениями;
- 4) тексты программ;
- 5) инструкцию по работе с программой в которой указываются:
 - цели и основные задачи, для решения которых предназначена данная программа, сфера ее применения;
 - минимальная конфигурация вычислительной системы, требуемая для работы программы в различных вариантах и с разными объемами исходных данных;
 - сведения о типе и числе необходимых устройств ввода-вывода, необходимом объеме внешней и внутренней памяти, а также типе операционной системы, на применение которой ориентирована программа;
 - ранее разработанные программные средства или обрабатывающие программы операционной системы, требуемые для работы данной программы;
 - данные о времени работы программы для нормативных объемов исходной информации;
 - основные логически обособленные этапы обработки данных. Обычно такие этапы реализуются в форме шагов одного задания, в выполнении программной системы. Одновременно могут быть проиллюстрированы возможные потоки информации в программе как данные, переходящие от одного шага задания к другому. Кроме того, здесь четко прослеживаются функции модулей, реализующих выполнение каждого шага задания;
 - режим функционирования программы и средства их запроса;

- форма и правила подготовки исходных данных, которые могут быть обработаны программой;

- описание исходных, промежуточных и основных массивов, обрабатываемых и составляемых программой;

- все диагностические сообщения, выдаваемые программой, с соответствующими разъяснениями и указанием действий обслуживающего персонала по устранению той или иной ошибки.

б) контрольный пример, демонстрирующий правильность выполнения программы в различных режимах ее функционирования и с различными вариантами типов исходных данных;

7) список используемой литературы.

Основную часть составляют материалы, с которыми шла работа на этапах проектирования, алгоритмизации и программирования (проект разработки, общая блок-схема, алгоритмы и т.д.). Поэтому все перечисленные материалы должны быть в рабочем состоянии и по содержанию вполне соответствовать друг другу в отлаженной программе. Кроме того, уже на этапах разработки их нужно представлять в таком виде, чтобы они могли быть использованы при составлении пояснительной записки без дополнительных переделок.

3. ОФОРМЛЕНИЕ КУРСОВОГО ПРОЕКТА

Пояснительная записка по курсовому проекту должна быть оформлена в соответствии с:

ГОСТ 19.001-77, ГОСТ 19.002-80, ГОСТ 19.003-80, ГОСТ 19.004-80, ГОСТ 19.104-78, ГОСТ 19.105-78, ГОСТ 19.106-78, ГОСТ 19.201-78, ГОСТ 19.301-79, ГОСТ 19.401-78, ГОСТ 19.402-78, ГОСТ 19.403-79, ГОСТ 19.404-79.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Хьюз Д., Мичтом Д. Структурный подход к программированию.-М.: Мир, 2006.
2. Йодан Э. Структурное проектирование и конструирование систем.- М.: Мир,2007.
3. Вирт Н. Систематическое программирование.-М.: Мир, 2009.
4. Майерс Г. Надежность программного обеспечения.- М.: Мир, 2010.
5. Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ.- М.: Мир, 2011.
6. Безбородов Ю.М. Индивидуальная отладка программ.- М.: Наука,2010.

ПРИЛОЖЕНИЕ 1

1. Обеспечение доступа к ресурсам сети.

Ввести с экрана пароль пользователя. Вводимые символы не показывать. Предусмотреть возможность удаления введенного символа. Проверить введенный пароль и сообщить пользователю о разрешении на использование ресурсов сети. Предусмотреть возможность изменение эталона.

2. Анализ процедуры "быстрой" сортировки.

Получить временные данные по сортировке разновеликих массивов целых чисел (использовать как минимум два метода). Нарисовать диаграмму зависимости времени сортировки от размера массива.

3. Обработка записей Базы Данных.

Ведение журнала результатов сдачи сессии студентами группы.

Режимы:

- ввод списка студентов группы (пять групп),
- ввод оценок по десяти предметам,
- выдача сводки по успеваемости (Отлично - NN, Хорошо - NN),

- сохранение информации в файле,
- возможность чтения и дополнения журнала.

4. Редактирование строки на экране.

Составить процедуру редактирования строки на текстовом экране.

Предусмотреть:

- движение курсора по строке,
- вставку и замещение символов с показом текущего состояния (Вставка или Замещение),
- удаление символов,
- движение по словам строки,
- длина строки 10..200 символов.

5. Автоматизация процессов подготовки документации на программное обеспечение.

Для файла, заданного пользователем, содержащего текст программы на языке Паскаль, составить список используемых в нем переменных, констант, меток, процедур и функций. Показать или записать в другой файл этот список. Переменные, константы и метки отсортировать по алфавиту. Процедуры и функции списка сгруппировать следующим образом:

- системные процедуры ;
- системные функции ;
- процедуры пользователя ;
- функции пользователя.

6. Обработка текстов.

Форматирование текста Паскаль - программы с выравниванием и смещением вложенных блоков, пропуском строк между частями программы, оформлением только одного оператора на строке, единым видом ключевых

слов и идентификаторов.

7. Клавиатурный тренажер.

Составьте программу, которая развивает навыки работы с клавиатурой. Игровая ситуация: буквы 'падают' с верхней строки экрана и пользователь должен правильно и быстро нажимать соответствующие клавиши до того, как буква опустится на предпоследнюю строку экрана. Скорость падения все время возрастает. Игра заканчивается при пропуске порогового числа букв. Задача должна учитывать точность работы пользователя и выставлять оценку в баллах.

8. Графический редактор с ограниченным набором функций.

Написать программу создания пользователем "картинки", сохранения графического изображения в файле с последующим воспроизведением.

9. Редактирование текстов на экране.

Размер файла не должен превышать числа строк экрана. Режимы:

- ввод имени редактируемого файла (новый файл),
- редактирование с движением курсора по экрану, вставкой, замещением, удалением символов,
- удаление и вставка строк,
- сохранение текста с экрана в файл.

10. Имитация электронной таблицы.

Имитация электронной таблицы в пределах экрана с обработкой арифметических операций с элементами таблицы.

Строки: латинские буквы (A-Z)

Столбцы: цифры.

Ссылка на элемент: A1, B5.

Режимы:

- движение по клеткам таблицы,
- ввод значений элементов,
- ввод в качестве элемента формул для расчета с использованием имен элементов (A1+B3),
- сохранение и чтение таблицы.

11. Игра в слова с компьютером .

Длина слова - пять символов. Предусмотреть выбор права первого хода: человек или машина. Первый игрок предлагает слово, второй меняет в нем одну букву и передает ход. Проигрывает тот, кто не может предложить следующее слово. (Предусмотреть контроль за правильностью введенных слов от игрока - человека с вопросом о корректности и пополнением словаря. Словарь оформить в виде файла.)

12. Игра с компьютером. Составляем слова.

Из букв введенного (предложенного ЭВМ) слова составить как можно больше разных слов. Право первого хода может разыгрываться или выбираться. Выигрывает тот, кто составит как можно больше слов. Можно играть с ЭВМ или двум игрокам - пользователям. Слова вводятся (показываются) по очереди. (ЭВМ может пополнять свой запас слов. Желательно файл со словами закодировать).

13. Игра с компьютером. "Города".

Два игрока (один - компьютер) последовательно называют города, так, чтобы название следующего города начиналось с последней буквы от названия предыдущего.

Предусмотреть возможность выбора или розыгрыша первого хода.

Предусмотреть справку для каждого города, не только для того, название которого предлагает ЭВМ, но и, по возможности, для того, который указывает человек. Справка должна содержать:

- название континента (часть континента: Юго - Запад, Европа);
- название страны;
- столица государства.

Вызов справки - по нажатию клавиши .

14. Игра "Пятнадцать". На квадратном поле 4x4 с помощью датчика случайных чисел расставлены 15 фишек с номерами от 1 до 15. Имеется одна свободная позиция. Расставить фишки по возрастанию их номеров. Передвигать фишки можно только на соседнюю свободную позицию.

15. Игра "Морской бой". На поле 10x10 позиций стоят невидимые вражеские корабли: 4 корабля по 1 клетке, 3 корабля по 2 клетки, 2 корабля по 3 клетки, 1 корабль в 4 клетки. Необходимо поразить каждую из клеток кораблей. Позиции указываются русскими буквами от А до К (по строкам) и цифрами от 1 до 10 (по столбцам). Конфигурация и положение кораблей на поле выбираются с помощью датчика случайных чисел. Если клетка корабля угадана играющим верно, она отмечается крестиком; в противном случае точкой.

16. Игра "Ипподром". Играющий выбирает одну из трёх лошадей, соревнующихся на бегах, и выигрывает, если его лошадь приходит первой. Скорость передвижения лошадей на разных этапах выбирается программой с помощью датчика случайных чисел.

17. Игра "Тезей". Напишите программу, которая по исходным данным m и n строит прямоугольный лабиринт $m \times n$ (проверьте, допустимы ли заданные m и n). Предусмотрите, чтобы программа при каждом обращении к ней порождала разные лабиринты. Лабиринт должен иметь единственное решение, и, чтобы получившийся лабиринт был интересным, все ячейки должны

быть соединены с основным путём, дающим решение.

18. Игра "Проще простого". Напишите программу, которая отображает целые числа на плоскость некоторым регулярным образом и отмечает на рисунке места, где находятся простые числа. Выведите формулы, описывающие прямые линии на вашем рисунке, и напечатайте те из них, которые особенно изобилуют простыми числами; печатайте также долю простых чисел на этих прямых. Обеспечьте высокую эффективность ваших программ проверки целых чисел на простоту, так чтобы вам хватило времени для анализа весьма отдалённых отрезков натурального ряда.

19. Игра в слова с компьютером "Балда". Вписываются слова из 5 букв в квадрат 5 на 5 по горизонтали, среднее слово составляется из существующих и дописанной 1 буквы в квадрат.

20. Игра "Крисс - кросс". Построение кроссворда из слов предложенных пользователем. Напишите программу, читающую список слов и строящую для этого списка правильную схему крисс-кросса. Представьте заполненную схему как доказательство того, что она правильная. Возможно, хотя и маловероятно, что для данного списка слов не существует решения (как и в кроссворде, схема должна быть связной). Ваша программа должна сообщать о всех неудачах при построении схемы и о всех ситуациях, нарушающих однозначность (таких, например, как наличие повторяющихся слов). Попутно решите ещё одну задачу - получите красивый графический вывод.