

Министерство общего и профессионального образования Свердловской области
государственное автономное профессиональное образовательное учреждение
Свердловской области
«Уральский радиотехнический колледж им. А.С. Попова»

Методические указания
по выполнению курсовой работы
ПМ.01 Разработка программных модулей программного обеспечения для
компьютерных систем
для студентов специальности 09.02.03
«Программирование в компьютерных системах»

2016 год

Утверждено

На заседании цикловой комиссии ИТ

Протокол № ____ от « ____ » _____

Председатель ЦМК _____ Н.Н.Шутова

Разработал: преподаватель Е.В.Тыщенко

1 ОБЩИЕ ПОЛОЖЕНИЯ

Курсовая работа является самостоятельной работой студента, позволяет оценить качество знаний и отражает приобретенные студентом практические навыки.

Курсовая работа позволяет расширить объем знаний студентов в области программирования и создать реальную основу использования своих знаний для решения на ЭВМ задач по другим дисциплинам и в своей дальнейшей профессиональной деятельности.

Перед студентом ставится задача разработать приложение для Windows с целью решения конкретной задачи. Результатом решения является:

- а) исполняемый файл программы;
- б) пояснительная записка, составленная с учетом требования стандартов ЕСПД.

Для решения поставленной задачи студенту необходимо предварительно ознакомиться с литературой, посвященной теме задания. При этом следует обратить внимание на средства, используемые для решения аналогичных задач или для решения каких-либо ключевых моментов задачи.

Сформулированные в настоящих указаниях задания на курсовую работу представляют студенту простор для творчества. После ознакомления с литературой студент должен оценить возможности языка программирования и вычислительной техники, на которой предлагается реализовать решение. Результатом этой работы должна быть точная формулировка задачи со всеми ограничениями и требованиями.

При решении задачи необходимо придерживаться техники пошаговой детализации, использовать стандартные структуры, не забывая при этом о развитии программного окружения программиста, расширяя возможности языка за счет включения новых процедур и функций.

При разработке алгоритма необходимо предусмотреть средства проверки и тестирования программы, удобство работы пользователя, возможные

модификации.

При написании программы не следует забывать о хорошем стиле программирования, о таких понятиях, как читабельность, эффективность, надежность. Необходимо искать наиболее простые и естественные приемы и методы решения.

Выполнение курсовой работы состоит из трех этапов:

- 1) Подготовительный этап (разработка технического проекта).
- 2) Разработка рабочего проекта
- 3) Оформление пояснительной записки.

2 СОДЕРЖАНИЕ РАЗДЕЛОВ КУРСОВОЙ РАБОТЫ

Все этапы разработки программы отражаются в пояснительной записке.

Содержание пояснительной записки:

Введение

1 Постановка задачи

2 Системный проект

2.1 Описание алгоритма

2.2 Организация входных и выходных данных

3 Технический проект

3.1 Спецификация программы

3.2 Текст программы

3.3 Тестирование программы

3.4 Руководство пользователя

Список использованных источников

При написании пояснительной записки необходимо придерживаться требований единой системы программной документации (ЕСПД): шрифт Times New Roman, кегль 14, полуторный интервал.

Полужирный шрифт и подчеркивание не допускаются.

Величина абзацного отступа 1,5 см. Текст печатается с соблюдением полей: левого – 25 мм, правого, нижнего, верхнего – 15 мм. Страницы нумеруются арабскими цифрами. Номер страницы проставляют в нижней части листа без точки. Титульный лист и задание включают в общую нумерацию страниц, номера страниц на них не проставляются.

Разделы должны иметь порядковые номера, обозначенные арабскими цифрами без точки, и записаны с абзацного отступа.

Заголовок раздела печатается прописными буквами без точки в конце.

Каждый раздел начинается с нового листа.

Подразделы нумеруются в пределах раздела.

Номер подраздела состоит из номера раздела и подраздела, разделенных

точкой. Например, 1.1

В конце номера точка не ставится.

Заголовки подразделов печатаются с абзацного отступа с прописной буквы, в разрядку на 1,1 пункт без точки в конце.

Перед началом перечислений ставится двоеточие.

Перед каждым элементом перечисления ставится маркер в виде дефиса.

Все иллюстрации в тексте размещают сразу после первой ссылки на них и обозначают словом рисунок.

Иллюстрации нумеруются арабскими цифрами. После номера рисунка ставится тире и пишется наименование с прописной буквы.

Документу присваивается десятичный номер разработки, который проставляется на титульном листе и в основной надписи. (ГОСТ 2.104).

Он складывается:

- код учебного заведения;
- код специальности;
- номер группы по журналу группы;
- код документа;
- номер в списке по журналу группы.

Пример написания десятичного номера:

РК 09.02.03.417 09 ПЗ

2.1 Содержание

Содержание составляется в соответствии с содержанием пояснительной записки и должно отражать все разделы курсовой работы. После написания пояснительной записки в содержании проставляются страницы.

2.2 Введение

В разделе "Введение" указывается задание на курсовую работу и дается краткая характеристика разделов пояснительной записки.

Например.

Темой курсовой работы является разработка приложения для Windows, представляющего собой компьютерную игру "Лабиринт". Пояснительная записка содержит семь разделов.

В разделе «Постановка задачи» приводится условие задачи, описание исходных данных и целей ее решения.

В разделе «Описание алгоритма» дается обобщенное словесное описание алгоритма решения поставленной задачи и приводится схема алгоритма.

Раздел «Организация входных и выходных данных» содержит описание и обоснование выбора метода организации входных и выходных данных.

Раздел «Описание программы» содержит инструкцию по работе с программой.

В разделе «Спецификация программы» приводится точное название программы и ее состав

Раздел «Текст программы» содержит тексты всех программных модулей.

В разделе «Тестирование программы» перечисляются требования, подлежащие проверке при испытании программы.

2.5 Постановка задачи

Решение задачи начинается с ее постановки. Дается точное описание исходных данных, условий задачи и целей ее решения. В этом разделе могут быть описаны основные приемы программирования и типы данных, используемые при решении аналогичных задач. Например, если в задаче используются динамические структуры, то перечисляются виды динамических структур данных и основные процедуры по работе с динамическими структурами. Если задача заключается в формировании базы данных и дальнейшей работе с базой, то приводится описание используемых типов данных (тип запись) и приемы работы с файлами.

Далее описываются возможные пути решения задачи с указанием их достоинств и недостатков. Выбирается и обосновывается метод решения задачи.

Описываются ограничения, накладываемые на исходные данные, необходимая разрядность и точность представления исходных данных и результатов решения. Указываются возможные пределы изменения входных параметров задачи.

Например.

Разработать приложение для Windows, представляющее собой компьютерную игру "Лабиринт". Игра "Лабиринт" состоит в том, что играющий перемещается в двухмерном пространстве по помещениям здания, план которого неизвестен. Начиная с произвольного помещения, путешественник должен найти выход из здания. Каждое помещение может иметь четыре двери: север, восток, юг, запад. План здания необходимо считать из текстового файла в связанный список. Порядок следования помещений в списке должен быть произвольным. Находясь в N-ом помещении, игрок может получить справку о правильном направлении движения, если верно ответит на вопрос по теме "Алгоритмические языки и программирование".

В условии задачи игры "Лабиринт" указывается на необходимость использования динамической структуры "связанный список" и текстового файла для хранения информации о плане помещений лабиринта, поэтому нужно в постановку задачи включить определения динамических структур и организации файлов, а также обзор методов работы с такими структурами. В задаче также требуется организовать подсказку для выбора направления движения в лабиринте. Подсказкой может служить правильно выбранный ответ по теме курса. Представляемые играющему вопросы и варианты ответа также должны быть считаны из файла на диске. На организацию этого файла не накладываются дополнительные требования, поэтому студент должен сам принять решение о структуре файла.

После описания общих положений тематики работы, необходимо указать конкретные методы решения поставленной задачи. Предположим, что помещения здания соединяются между собой так, как показано на рис. 1.

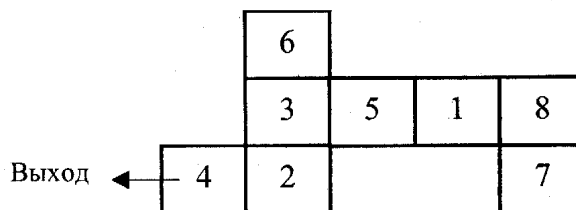


Рис. 1 - Пример плана помещений

Информация о плане здания должна храниться в файле в последовательности: номер комнаты; номера комнат, смежных с данной. Смежные комнаты перечислены в порядке: север, восток, юг, запад (рис. 2). Порядок следования помещений в этом списке может быть любым, как и последовательность номеров помещений на плане.

Номер комнаты	Дверь в комнату			
	норд	ост	зюйд	вест.
7	8	0	0	0
1	0	8	0	5
8	0	0	7	1
5	0	1	0	3
6	0	0	3	0
3	6	5	2	0
2	3	0	0	4
4	0	2	0	999
999	0	0	0	0

(0 = нет двери. 999 = выход)

Рис. 2 – Информация о плане здания, представленном на рис.3

Файл с данными считывается и помещается в список, исходя из номера помещения (рис. 3). Sp_i – указатель на начало списка.

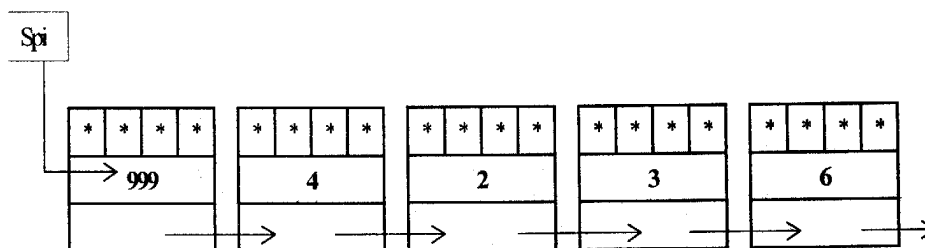


Рис. 3 – Начало списка помещений после первого прочтения представленного на рисунке 2 файла

Затем файл считывается еще раз и с учетом наличия дверей генерируется связь помещений (рис. 4).

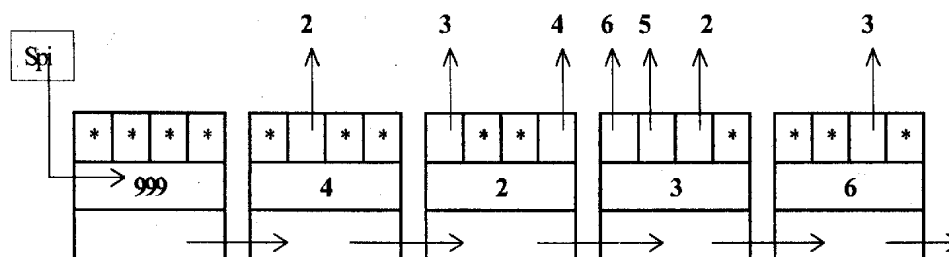


Рис. 4 - Начало списка помещений после второго прочтения представленного на рис. 2 файла.

Звездочкой помечены ссылки, равные Nil.

Перемещение игрока в лабиринте осуществляется путем выбора им одного из вариантов направления движения. Игрок вводит номер комнаты, с которой начинается путешествие. Если такого помещения в лабиринте нет, то об этом выдается сообщение. Если такая комната существует, то играющему предлагается выбрать направление движения, которое можно определить, воспользовавшись подсказкой. Помощь заключается в том, что на экран выводится вопрос по теме курса "Алгоритмические языки и программирование" и 4 варианта ответа. Правильный ответ указывает направление дальнейшего движения. Неправильный ответ дает неверное направление или вызывает сообщение о невозможности передвижения. Вывод подсказки на экран и выбор правильного ответа осуществляются с помощью специальной процедуры, которая обращается к файлу на диске. Файл-подсказку сделаем типизированным. Количество компонентов файла равно количеству помещений в здании-лабиринте. Номер компонента файла соответствует номеру помещения. Правильный ответ расположен в положении, определяющем верное направление. Компонент типизированного файла имеет тип "запись", где отводится поле для хранения вопроса по теме курса и поле - массив предлагаемых ответов.

Запись:

- поле вопроса;
- поле 4-х ответов.

Признаком окончания игры является ссылка на помещение с номером 999. В программу включим возможность создания и корректирования файлов

плана помещения и подсказки.

2.6 Описание алгоритма

В разделе дается обобщенное словесное описание алгоритма решения поставленной задачи, излагаются основные требования к алгоритму и пути их реализации. Приводится схема алгоритма, состоящая из укрупненных модулей. Дается пояснение назначения и состава каждого модуля. Обобщенный алгоритм обычно использует обозначения и термины исходной задачи.

На следующем этапе каждый модуль детализируется. Выделяются укрупненные команды, реализуемые по вспомогательным алгоритмам. Тот же подход применяется при разработке вспомогательных алгоритмов.

Пример.

В программе решаются три независимые друг от друга подзадачи:

- перемещение по лабиринту;
- запись на диск плана помещений здания;
- запись на диск файла подсказки.

Задача "Перемещение по лабиринту" решается в два этапа:

- 1) формирование списка помещений;
- 2) выбор направления движения и переход из одного помещения в другое.

Обобщенная схема алгоритма приведена на рис. 5.

При формировании списка плана помещений сначала производится считывание номеров комнат из текстового файла и построение связанного списка затем построение многосвязанного списка, содержащего в себе план помещений здания.

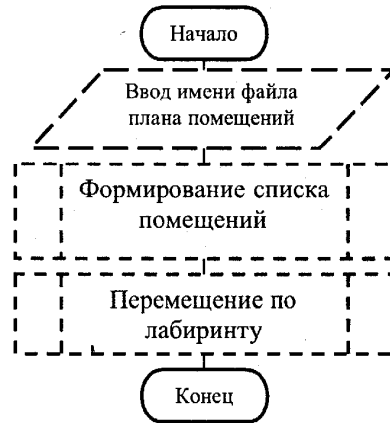


Рис. 5 – Обобщенная схема алгоритма

Перемещение по лабиринту начинается с выбора номера помещения для начала путешествия. Если такого помещения нет, то игра заканчивается. Если такое помещение в лабиринте есть, то игроку предоставляется возможность выбрать направление движения или воспользоваться файлом подсказки. Укрупненная схема алгоритма приведена на рис. 6.

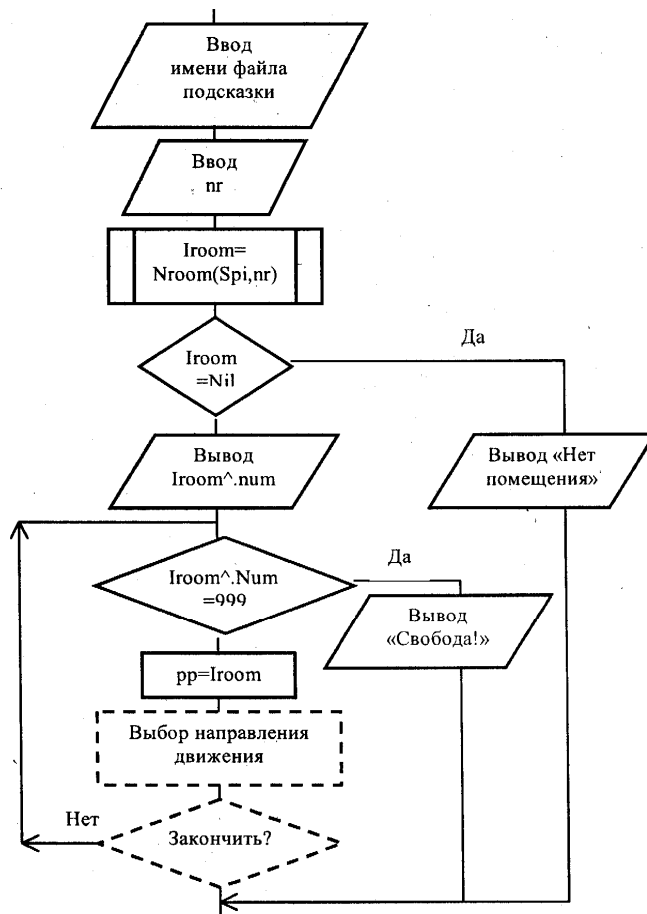


Рис.6 – Схема алгоритма этапа «Перемещение по лабиринту»

Аналогично детализируются все составляющие алгоритма. На этапе разработки технического проекта детализация на уровне операторов программы необязательна.

2.5 Организация входных и выходных данных

Данный раздел содержит описание и обоснование выбора метода организации входных и выходных данных.

В приведенном примере по условию задачи требуется использование файлов прямого доступа. Файл, содержащий план лабиринта, может быть организован непосредственно в приложении. Для этого разрабатывается отдельный фрагмент программы. Структура файла приведена на рис. 2. В текстовом файле в одной строке записаны номер комнаты и номера комнат смежных помещений. Смежные помещения располагаются в порядке: норд, ост, зюйд, вест. Если смежное помещение отсутствует, то вводится значение 0. Номера комнат в файле могут располагаться в произвольном порядке. Важным является то, что информация о комнате должна быть расположена в отдельной строке.

Разрабатываемое приложение предусматривает использование файлов прямого доступа. В файлах прямого доступа хранится информация подсказки. Файл-помощь может быть сформирован непосредственно в приложении. Для этого разрабатывается отдельный фрагмент программы. Количество компонентов файла равно количеству помещений в здании-лабиринте. Номер компоненты файла соответствует номеру помещения. Правильный ответ расположен в положении, определяющем верное направление. Компонент типизированного файла имеет тип "запись", где отводится поле для хранения вопроса по теме курса и поле - массив предлагаемых ответов.

Для структуры помещений, приведенной на рис. 1, содержание файла-подсказки может быть следующим:

Какая файловая переменная является стандартной?

Con

Prn

F

Input

Какая файловая переменная текстового типа описана правильно?

F4: text80;

F1: File of String;

F: File of String [80];

F2: Text;

Какая процедура предназначена для связи файловой переменной с физическим файлом на диске?

Reset

Rewrite

Assign

Read

Какая процедура предназначена для чтения данных из типизированного файла?

Readln

ReadBlock

Input

Read

2.6 Написание программы

Программную реализацию разработанных алгоритмов содержат обработчики событий. На этапе разработки рабочего проекта необходимая степень детализации алгоритмов обычно выбирается такой, чтобы предписания разработанных алгоритмов могли записываться на языке программирования, выбранном для составления текста программы. При детализации алгоритма необходимо перейти к обозначениям, принятым для разработки программ на алгоритмическом языке. При этом имена следует выбирать таким образом, чтобы они отражали сущность используемых параметров.

Кодирование должно быть простым. Изощренное программирование может обойтись слишком дорого при отладке и модификации программы. Необычное кодирование (например, использование скрытых возможностей машины) часто препятствует отладке программы и затрудняет ее использование другими программистами. Программа должна быть по возможности универсальной. Универсальные программы обеспечивают независимость программы от конкретного набора данных. Например, универсальная программа использует в качестве параметров переменные, а не константы, обрабатывает вырожденные случаи и т. д. Универсальность программы экономит время по дальнейшей работе над ней и обеспечивает широкие возможности по использованию.

Разрабатывая такие программы, можно предвидеть будущие изменения в спецификациях этой программы.

Входные форматы должны быть разработаны с учетом максимального удобства для пользователя и минимальной возможности ошибок. Порядок переменных и форматы данных, привычные для пользователя, помогут избежать ошибок и облегчат использование программ.

При написании программы следует применять операторы, позволяющие использовать основные алгоритмические структуры. Оператор `Goto` желательно не использовать.

При написании программ не следует забывать о хорошем стиле программирования. После заголовка процедуры или функции записывается комментарий, содержащий поясняющий текст, а именно: назначение подпрограммы; перечень и назначение формальных параметров, их тип. Комментариями должны быть снабжены и основные смысловые блоки программы или подпрограммы. Для облегчения чтения текста программы отдельные операторы программы записываются с отступом.

2.7 Спецификация программы

В разделе спецификация приводится точное название программы и ее

состав. Форма спецификации приведена в приложении 2. Графы спецификации заполняют следующим образом:

в графе "Обозначение" указывают обозначение основных программных компонентов;

в графе "Наименование" указывают полное наименование соответствующего компонента;

в графе "Примечание" — дополнительные сведения, относящиеся к записанным в спецификации программам.

Пример.

Исполняемый файл программы "Игра "Лабиринт"" имеет название Labirint.exe и расположен на диске в каталоге Student\Kurosov\Labirint.

Таблица1 - Состав проекта

Наименование	Обозначение	Примечание
Ad	Файл плана	Создается в приложении
Help1	Файл помощи	Создается в приложении
Labirint.Dof	Файл параметров проекта	Содержит текущие установки проекта: настройки компилятора и компоновщика, имена служебных каталогов, условные директивы
Labirint.Dpr	Файл проекта	Связывает все файлы, из которых состоит приложение
Labirint.Dsk	Файл, содержащий Desktop—настройки проекта	Содержит информацию о том, какие окна открыты и в каких позициях они расположены
Labirint.Res	Файл ресурсов	Содержит пиктограммы, графические изображения
Unit 1.Pas	Файл программного модуля для формы № 1	Определяет функциональность формы № 1
Unit2.Pas	Файл программного модуля для формы №2	Определяет функциональность формы №2
Unit3.Pas	Файл программного модуля для формы №3	Определяет функциональность формы №3

Unit1.Dfm	Файл формы № 1	Содержит список свойств всех компонентов, включенных в форму № 1
Unit2.Dfm	Файл формы №2	Содержит список свойств всех компонентов, включенных в форму №2
Unit3.Dfm	Файл формы №3	Содержит список свойств всех компонентов, включенных в форму №3
Unit1.Dcu	Объектный файл для Unit 1.Pas	Откомпилированная версия Unit 1.Pas
Unit2.Dcu	Объектный файл для Unit2.Pas	Откомпилированная версия Unit2.Pas
Unit3.Dcu	Объектный файл для Unit3.Pas	Откомпилированная версия Unit3.Pas

2.8 Текст программы

Приводится полный листинг всех модулей программы.

2.9 Тестирование программы

Кратко описывается среда программирования. Приводятся основные команды, выполняемые при вводе и редактировании программы, команды записи программы на диск, чтения с диска. Перечисляются и описываются средства отладки.

Перечисляются требования, подлежащие проверке при испытании программы, а также порядок и методика их контроля. Приводятся исходные данные для решения контрольного примера и ожидаемые результаты.

Прилагаются окна приложения на этапе выполнения с обработкой исключительных ситуаций.

Пример.

При разработке программы "Лабиринт" в качестве контрольного примера может быть использован пример, приведенный в разделе "Постановка задачи". Должны быть проверены ситуации с отсутствием файлов, необходимых для работы программы.

2.9 Список использованных источников

В разделе перечисляется литература, использованная при выполнении курсовой работы, Интернет-ресурсы. При оформлении литературы необходимы следующие данные:

- фамилии и инициалы авторов через запятую;
- название книги или статьи;
- место издания (сокращенно);
- издательство или название журнала;
- год издания (для журнала добавляется номер журнала). Источники нумеруются для того, чтобы на них можно было ссылаться из текста пояснительной записки.

Например,

- 1 С. Окулов Программирование в алгоритмах, Бином. Лаборатория знаний, 2013 г.
- 2 С. Окулов Основы программирования, Бином. Лаборатория знаний, 2015 г.
- 3 учебник с практикумом <http://inform-school.narod.ru/index.htm>

УТВЕРЖДЕНО
цикловой методической комиссией
информационных технологий
«__» _____ Н.Н Шутова

ЗАДАНИЕ

Для курсового проектирования по ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем студенту(ке) «2» курса группы «_____»
ГАПОУ СО «Уральский радиотехнический колледж им А.С. Попова»
Тема задания _____

Курсовой проект на указанную тему выполняется студентом в следующем объеме:

1) Пояснительная записка состоит из следующих разделов:

Введение

1 Постановка задачи

2 Системный проект

2.1 Описание алгоритма

2.2 Организация входных и выходных данных

3 Технический проект

3.1 Спецификация программы

3.2 Текст программы

3.3 Тестирование программы

3.4 Руководство пользователя

Список использованных источников

2) Программа

Дата выдачи «__» _____ 20__ г

Срок окончания «__» _____ 20__ г

Преподаватель _____ Е.В.Тыщенко