

ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Т.Г. ШЕВЧЕНКО
Рыбницкий филиал
Кафедра физики, математики и информатики

ПРОГРАММИРОВАНИЕ

Методические рекомендации

Рыбница, 2014

УДК 681.3.06
ББК 32.973-01
П78

Составители:

Е.С. Гарбузняк, преп.

Н.В. Чернега, преп.

О.В. Шестопал, ст. преп.

Рецензенты:

В.Е. Фёдоров, канд. экон. наук, доц. каф. прикладной информатики филиала ПГУ им. Т.Г. Шевченко в г. Рыбница

Л.А. Тягульская, канд. экон. наук, доц. каф. физики, математики и информатики филиала ПГУ им. Т.Г. Шевченко в г. Рыбница

П78 Программирование: Методические рекомендации / Сост.: Е.С. Гарбузняк, Н.В. Чернега, О.В. Шестопал. – Рыбница, 2014. – 100 с. – (в обл.)

Методические рекомендации содержат основные положения и требования по подготовке, написанию, оформлению и защите курсовых работ по дисциплине «Программирование», а также образец курсовой работы.

Методические рекомендации предназначены для студентов I курса направления «Педагогическое образование», профиля «Информатика», дополнительного профиля «Иностранный язык». Могут быть использованы студентами других направлений.

УДК 681.3.06
ББК 32.973-01

Рекомендовано Научно-методическим советом ПГУ им. Т.Г. Шевченко

© Е.С. Гарбузняк, Н.В. Чернега, О.В. Шестопал
составление, 2014

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ.....	4
1. ОБЩИЕ ПОЛОЖЕНИЯ О КУРСОВЫХ РАБОТАХ	5
1.1. Задачи курсовой работы.....	5
1.2. Порядок подготовки и защиты курсовой работы	5
1.3. Общие требования к выполнению курсовой работы.....	7
1.4. Общие требования к структуре, содержанию и оформлению курсовой работы.....	7
1.5. Оценка курсовой работы.....	8
2. ПОНЯТИЙНЫЙ АППАРАТ ИССЛЕДОВАНИЯ.....	10
2.1. Актуальность, степень изученности и научной разработанности темы исследования.....	10
2.2. Объект и предмет исследования	11
2.3. Цель и задачи исследования	11
2.4. Новизна и практическая значимость	12
3. ВЫБОР ТЕМЫ И ПЛАНИРОВАНИЕ КУРСОВОЙ РАБОТЫ	13
3.1. Выбор темы исследования.....	13
3.2. Составление рабочего плана.....	14
3.3. Работа с литературой.....	15
4. НАПИСАНИЕ КУРСОВОЙ РАБОТЫ	17
4.1. Разработка структуры работы	17
4.2. Подготовка введения.....	19
4.3. Работа над основной частью.....	20
4.4. Подготовка заключения и формулирование выводов	23
5. ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ	25
5.1. Правила оформления текстового материала	25
5.2. Правила оформления иллюстраций	26
5.3. Правила оформления формул.....	27
5.4. Правила оформления таблиц	28
5.5. Правила оформления библиографического списка	31
5.6. Правила оформления библиографических ссылок.....	39
5.7. Правила оформления приложений.....	41
5.8. Правила оформления блок-схем	42
6. ЗАЩИТА КУРСОВОЙ РАБОТЫ	45
6.1. Подготовка доклада.....	45
6.2. Подготовка презентации	46
6.3. Защита курсовой работы.....	48
СПИСОК ЛИТЕРАТУРЫ	51
ПРИЛОЖЕНИЕ	

ПРЕДИСЛОВИЕ

Курсовая работа – это одна из форм учебно-исследовательской работы, её выполнение является обязательным для всех студентов.

Выполнение курсовой работы представляет собой самостоятельное решение студентом под руководством преподавателя частной задачи или проведение исследования по одному из вопросов, изучаемых в курсе общепрофессиональных и специальных дисциплин.

Дисциплина «Программирование» относится к вариативной части профессионального цикла (БЗ.В.11) подготовки студентов по направлению 050100 «Педагогическое образование» (профиль «Информатика», дополнительный профиль «Иностранный язык»). Целью освоения дисциплины «Программирование» является освоение теоретических и практических основ программирования.

Курсовая работа по программированию позволяет выполнить основные задачи дисциплины – получение представления об основах программирования и приобретение навыков практического использования инструментальных средств программирования в информационных системах.

Методические рекомендации содержат основные положения и требования по подготовке, написанию, оформлению и защите курсовых работ по дисциплине «Программирование».

Понятийный аппарат научного исследования и лексико-синтаксические конструкции, рекомендуемые для употребления при обосновании актуальности темы исследования, описании степени её изученности и научной разработанности, при изложении введения, основной части работы и заключения, позволят студентам выполнить требования к научному стилю написания курсовой работы.

Образец курсовой работы по дисциплине «Программирование», приведенный в приложении методических рекомендаций, позволит студентам избежать типичных ошибок при выполнении курсовых работ и грамотно их оформить.

Методические рекомендации могут быть использованы студентами других направлений при подготовке курсовых работ по изучаемым дисциплинам.

1. ОБЩИЕ ПОЛОЖЕНИЯ О КУРСОВЫХ РАБОТАХ

1.1. Задачи курсовой работы

Курсовая работа является неотъемлемой частью образовательного процесса.

Курсовая работа предназначена для углубления знаний студентов по изучаемым дисциплинам, развития навыков исследовательской деятельности, а также обучения студентов подбору, изучению и обобщению материалов, являющихся источниками информации в выбранной предметной области.

Выполнение курсовой работы осуществляется под руководством преподавателя – научного руководителя работы – и предполагает:

- развитие навыков сбора, систематизации, анализа и обобщения теоретического и практического материала по выбранной проблематике;
- развитие навыков работы с различными первоисточниками информации;
- углубление и систематизацию знаний по выбранной проблематике;
- формирование практических навыков проведения исследования;
- развитие умения применять получаемые в процессе обучения знания, умения, навыки для решения конкретных научных и прикладных задач;
- творческое применение полученных знаний, умений и навыков для обоснованного изложения подхода к решению задач, формулирования выводов.

1.2. Порядок подготовки и защиты курсовой работы

Графики подготовки, сдачи и защиты курсовых работ составляются и утверждаются в соответствии с общим положением о курсовой работе в вузе, методическими рекомендациями по подготовке и защите курсовых работ.

Порядок подготовки и защиты курсовой работы представлен на рисунке (рис. 1.1).



Рис. 1.1. Порядок подготовки и защиты курсовой работы

Требования, предъявляемые к курсовым работам, и критерии их оценки рассмотрены в следующих разделах.

1.3. Общие требования к выполнению курсовой работы

Основной целью выполнения курсовой работы является расширение, углубление знаний студента и формирование у него навыков научно-исследовательской деятельности.

Тематика курсовых работ ежегодно разрабатывается и утверждается профессорско-преподавательским составом кафедры на заседаниях кафедр. Она должна соответствовать задачам изучения данной дисциплины и подготовки специалистов по данному профилю, предусмотренным в государственном образовательном стандарте, соответствовать современному уровню развития данной отрасли науки и опыту педагогической (производственной) деятельности.

Руководителем курсовой работы является, как правило, лектор, ведущий данную дисциплину. Заведующий кафедрой может назначить руководителем курсовой работы по дисциплине учебного плана преподавателя, ведущего практические занятия или иного преподавателя кафедры.

Для правильного выбора темы студент консультируется с научным руководителем, который поможет определить тему, поставить цели и задачи курсовой работы, даст советы по методике выполнения курсовой работы. Студент вправе предложить свою тему с обоснованием целесообразности её исследования.

Изменение темы курсовой работы допускается в исключительных случаях по обоснованному ходатайству самого студента или по инициативе научного руководителя. Тема курсовой работы дается одному студенту учебной группы.

Студент обязан написать, сдать и защитить курсовую работу в соответствии с предъявляемыми к ней требованиями на основании методических рекомендаций по подготовке курсовых работ, а также в соответствии с графиком выполнения курсовой работы, составленным совместно с научным руководителем.

1.4. Общие требования к структуре, содержанию и оформлению курсовой работы

Общие требования к структуре, содержанию и оформлению курсовой работы по направлению подготовки «Педагогическое образование» (профиль «Информатика», дополнительный профиль «Иностранный язык») представлены в табл. 1.1.

Таблица 1.1

**Общие требования к структуре, содержанию
и оформлению курсовой работы**

Группа требований	Описание основных требований
Структура	Типовая структура курсовой работы: – титульный лист; – содержание; – введение; – основная часть; – заключение; – список использованных литературных источников; – приложения.
Содержание	Содержание курсовой работы должно: – включать обоснование актуальности темы; – соответствовать сформулированной теме; – включать постановку задачи исследования; – объективно освещать, анализировать и обобщать известные достижения и факты в выбранной предметной области; – отражать знание современного состояния выбранной проблематики; – аргументировано, обоснованно и чётко излагать полученные результаты и обобщенные выводы. Используемая информационная база курсовой работы должна быть полной и достаточной для решения поставленных задач.
Объём	Не менее 20 и не более 35 страниц печатного текста без приложений.
Оформление	Стиль изложения – научный. Требования к оформлению курсовой работы рассмотрены в соответствующем разделе.

1.5. Оценка курсовой работы

Курсовая работа может быть оценена на «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Оценка проставляется на титульном листе с подписью научного руководителя.

Оценка вносится в экзаменационную ведомость и зачетную книжку. Отрицательная оценка в зачетную книжку не вносится. Полные названия курсовых работ вносятся в приложение к диплому в соответствии с распоряжением УАП ПГУ им. Т.Г. Шевченко.

Классификация и описание критериев оценки курсовых работ приведены в табл. 1.2.

Таблица 1.2

Критерии оценки курсовой работы

Группа критериев	Описание критериев оценки
Структура	Логичность построения курсовой работы. Логичность и структурированность изложения материала.
Содержание	Актуальность темы и содержания. Ясность постановки цели и задач. Соответствие содержания курсовой работы поставленной цели и задачам. Логичность и структурированность изложения материала, умение систематизировать материал. Умение выбирать методы для реализации поставленных задач. Самостоятельность подхода к решению задач. Аргументированность, достоверность и обоснованность полученных результатов и выводов. Оригинальность изложения. Отсутствие лишнего материала.
Объём	Объём в пределах установленной нормы.
Оформление	Выполнение требований к оформлению. Отсутствие грамматических ошибок. Научный стиль изложения.
Выполнение	Своевременность сдачи материалов.
Защита	Содержание доклада на защите. Качество презентации. Полнота и аргументированность ответов на поставленные вопросы. Умение вести дискуссию.

2. ПОНЯТИЙНЫЙ АППАРАТ ИССЛЕДОВАНИЯ

2.1. Актуальность, степень изученности и научной разработанности темы исследования

Актуальность (от лат. *actualis* – фактически существующий, настоящий, современный) темы исследования является одним из основных требований, предъявляемых к курсовой работе.

Обоснование актуальности темы излагается во введении работы и заключается в аргументации необходимости проведения исследования по выбранной тематике.

Актуальность темы предполагает её увязку со степенью изученности и научной разработанности.

Степень изученности и научной разработанности темы представляет собой краткий обзор и обобщенный анализ научных достижений в выбранной области.

При обосновании актуальности и описании степени изученности и научной разработанности темы рекомендуется использовать типовые лексико-синтаксические конструкции, приведенные в табл. 2.1.

Таблица 2.1

Лексико-синтаксические конструкции, рекомендуемые для употребления при обосновании актуальности темы исследования и описании степени её изученности и научной разработанности

Группа лексико-синтаксических конструкций	Примеры
Обоснование актуальности темы исследования	Многочисленные исследования ... показали, что одной из основных проблем в области ... является ... , поэтому необходимо ... Необходимость создания (подготовки, разработки) ... обусловлена отсутствием ... Актуальность темы исследования определяется следующими факторами: 1. ... 2. ... 3. ... Необходимость проведения исследования на выбранную тему обусловлена ... Для решения проблемы... необходимо ... , что определяет актуальность темы исследования ...
Описание степени изученности и научной разработанности темы исследования	В научных трудах и практических рекомендациях не нашли отражения вопросы, поэтому существует потребность в подготовке (разработке, создании) ... В настоящее время вопросы ... мало освещены в теоретических исследованиях и/или практических разработках, что определяет необходимость проведения исследований в области ...

2.2. Объект и предмет исследования

В каждом научном исследовании должны быть чётко определены объект и предмет исследования.

Объект исследования – это процесс или явление, порождающее проблемную ситуацию и выбранное для изучения. Это та часть практики или научного знания, с которой работает исследователь.

Предмет исследования – целостная составляющая объекта исследования, определенный аспект его рассмотрения, одна или несколько сторон, та точка зрения, с которой исследователь познает целостный объект, выделяя при этом наиболее существенные свойства, признаки, отношения, характеризующие объект исследования. Предмет исследования часто включается в формулировку темы работы.

Объект и предмет исследования как категории научного познания соотносятся между собой как общее и частное. В объекте выделяется та часть, которая служит предметом исследования. Например, если объектом исследования является семейство объектно-ориентированных языков программирования, то предметом исследования может быть решение задач с помощью средств языка *Object Pascal*.

Следует отметить некоторую особенность понятий объект и предмет: то, что выступает предметом в одних исследованиях, в других может рассматриваться как объект.

Формулировки объекта и предмета исследования обычно начинаются словами: «Объектом (предметом) курсовой работы является...».

2.3. Цель и задачи исследования

На основе задания курсовой работы, определенных объекта и предмета устанавливается цель курсовой работы.

Цель направлена на выполнение задания, от неё зависит весь ход дальнейшего исследования.

Цель исследования – это то, что должно быть получено в конечном итоге работы.

Формулировка цели обычно начинается словами: «Цель курсовой работы – ...».

Сформулированная цель определяет *задачи исследования*, которые чаще всего являются частными подцелями в некоторых условиях. Задачи исследования обеспечивают достижение общей цели исследования.

В работе обычно формулируется несколько задач, рекомендуемое количество 4-5. Задачи излагаются списком, в виде перечисления.

Перечисление задач может определяться либо временной последовательностью проведения исследования, либо логикой процесса исследования.

Поставленные задачи определяют структуру работы, описание их решения составляет содержание разделов работы. Из формулировки задач вытекают названия разделов и подразделов работы.

Задачи часто формулируются с помощью глаголов: выявить, исследовать, обосновать, определить, проанализировать, разработать, установить, или соответствующих существительных.

2.4. Новизна и практическая значимость

Под *новизной* результатов понимаются теоретические положения, которые сформулированы и обоснованы впервые, ранее в такой форме для автора не были известны, либо научно обоснованные технические, технологические или иные решения, имеющие важное значение для практики и которые ранее не применялись.

При формулировании результатов исследования необходимо показать их практическую значимость.

Практическая значимость результатов исследования:

- отражает их вклад в практику;
- показывает, что могут дать результаты для практики или что уже дали;
- характеризуется оценкой эффективности.

3. ВЫБОР ТЕМЫ И ПЛАНИРОВАНИЕ КУРСОВОЙ РАБОТЫ

3.1. Выбор темы исследования

Первоочередная задача при выборе темы курсовой работы состоит в определении направления исследования, объекта и предмета исследования.

При выборе направления исследований и определении темы курсовой работы желательно использовать имеющиеся накопленные материалы и практический опыт решения определенных проблем (при наличии таковых). На основании изученных источников определяется сфера научных интересов.

Рекомендуется также ознакомиться с ранее выполненными студенческими работами.

Формулирование названия темы исследования имеет важное значение, поскольку это определяет структуру, содержание, цели и задачи будущей курсовой работы. В названии работы рекомендуется использовать сочетание различных информационных блоков, например, новые особенности предмета исследования, цель исследования. Схема сочетания возможных информационных блоков приведена на рис. 3.1.

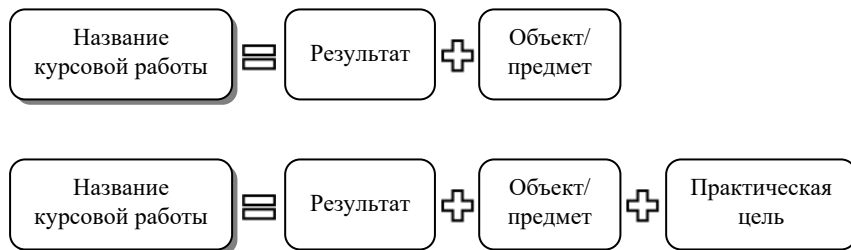


Рис. 3.1. Схема сочетания возможных информационных блоков названия темы курсовой работы

Примеры формулирования тем курсовых работ в зависимости от направлений и информационных блоков приведены в табл. 3.1.

Таблица 3.1

Примеры формулирования тем курсовых работ

Наименование направлений	Информационные блоки	Примеры формулировки темы
Решение задач	Результат – разработка программы. Объект – двоичная система счисления. Предмет – основные операции над двоичными числами.	Программа реализации основных операций над двоичными числами.
Защита информации	Результат – разработка программы. Объект – графические файлы Практическая цель – хранение сообщений.	Хранение текстовых сообщений в графическом файле bmp.

Следует отметить, что при выборе темы курсовой работы важно обосновать её актуальность, новизну, оценить возможности реализации.

3.2. Составление рабочего плана

При выполнении курсовой работы необходимо составить рабочий план, содержащий этапы и примерные сроки выполнения каждого этапа, результаты работы.

Процесс выполнения курсовой работы в общем случае состоит из следующих этапов:

1. Выбор направления исследования.
2. Теоретические и экспериментальные исследования.
3. Обобщение и оценка результатов исследований, выпуск отчетной документации.
4. Предъявление работы к проверке, сдача печатного и электронного вариантов и защита курсовой работы.

Выбор направления исследования проводят с целью определения оптимального варианта направления исследований на основе анализа состояния исследуемой проблемы и сравнительной оценки вариантов возможных решений с учетом результатов исследований, проводившихся по аналогичным проблемам.

Теоретические и экспериментальные исследования проводят с целью получения достаточных теоретических и достоверных экспериментальных результатов исследований для решения поставленных задач.

Обобщение и оценку результатов исследований, выпуск отчетной документации проводят с целью оценки эффективности полученных результатов в сравнении с современным научно-техническим.

План является документом, определяющим содержание, объём, сроки подготовки работы, формы отчётного материала, время проведения защиты.

План работы имеет произвольную форму. В табл. 3.2 представлен пример план-графика работ.

Таблица 3.2

Пример плана-графика работ

Вид документа, отчетной работы	Срок сдачи	Принимающее лицо / подразделение
План работы, проект введения	15 февраля	Научный руководитель
Материалы 1 раздела (рабочий вариант)	10 марта	Научный руководитель
Материалы 1 раздела (законченный вариант)	31 марта	Научный руководитель
Материалы 2 раздела (рабочий вариант)	20 апреля	Научный руководитель
Материалы 2 раздела (законченный вариант)	10 мая	Научный руководитель
Окончательный вариант курсовой работы	26 мая	Научный руководитель
Защита курсовой работы	8 июня	Кафедра физики, математики и информатики, научный руководитель

Перед составлением плана работы рекомендуется уточнить формулировку темы, обосновать актуальность темы, определить степень её изученности, цель и задачи исследования, обосновать выбор среды программирования.

3.3. Работа с литературой

Библиографический поиск литературных источников рекомендуется начинать с выделения ключевых слов исследования, далее продумать порядок поиска и формы хранения информации.

Для оценки состояния разработки (изученности) выбранной темы рекомендуется просмотр информационных изданий, цель выпуска

которых – представить оперативную информацию о самой публикации и наиболее существенных вопросах её содержания.

Информационное издание содержит либо систематизированные сведения об опубликованных, непубликуемых или неопубликованных документах, либо результат анализа и обобщения сведений, представленных в первоисточниках, и включают новизну сообщаемой информации, наличие справочного аппарата, позволяющего быстро систематизировать и отыскивать документы.

Поиск информации предполагает работу с различными видами источников информации. Информационные издания подразделяются на три вида: библиографические, реферативные и обзорные.

Библиографические издания содержат упорядоченную совокупность библиографических описаний, из которых формируются библиографические указатели и библиографические списки, и обеспечивают быстрое информирование специалистов о новых публикациях науки и техники.

Реферативное издание – информационное издание, содержащее упорядоченную совокупность библиографических записей, включающих рефераты. К реферативным изданиям относят: экспресс-информацию, реферативные журналы, реферативные сборники, информационные листки.

Обзорное издание содержит публикацию одного или нескольких обзоров, включающих результаты анализа и обобщения сведений по актуальным проблемам науки и техники. Обзорные издания представляют собой критически оцененную и обобщенную информацию проблемно-ориентированного характера.

Рекомендуемое число источников для бакалаврских курсовых работ составляет 15-20 источников.

4. НАПИСАНИЕ КУРСОВОЙ РАБОТЫ

4.1. Разработка структуры работы

При написании курсовой работы особое внимание следует уделять логике и композиции научного исследования.

Пример композиционной структуры курсовой работы, выполняемой в процессе обучения, представлен ниже.

Титульный лист
СОДЕРЖАНИЕ
ВВЕДЕНИЕ
1. НАЗВАНИЕ РАЗДЕЛА
1.1. Название подраздела
1.1.1. Название подподраздела
...
1.2. Название подраздела
...
Выводы по разделу 1
2. НАЗВАНИЕ РАЗДЕЛА
2.1. Название подраздела
2.1.1. Название подподраздела
...
2.2. Название подраздела
...
Выводы по разделу 2
ЗАКЛЮЧЕНИЕ
СПИСОК ЛИТЕРАТУРЫ
ПРИЛОЖЕНИЯ

Рекомендованная структура курсовой работы по программированию для студентов I курса по направлению подготовки «Педагогическое образование» (профиль «Информатика», дополнительный профиль «Иностранный язык») представлена ниже.

Титульный лист
СОДЕРЖАНИЕ
ВВЕДЕНИЕ
1. ИССЛЕДОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ
1.1. Современное состояние исследуемой задачи.
1.2. Описание применяемых алгоритмов.
1.3. Обзор методов решения подобных задач.
Выводы по разделу 1

2. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА, ОПИСАНИЕ РЕЗУЛЬТАТОВ

2.1. Постановка задачи, системные требования, требования к входным данным и выходным формам.

2.2. Структура программного продукта.

2.3. Описание пользовательских подпрограмм.

2.4. Тестирование и анализ полученных результатов.

Выводы по разделу 2

ЗАКЛЮЧЕНИЕ

СПИСОК ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЯ

Титульный лист является первой страницей курсовой работы и заполняется по определенным правилам. В верхней части страницы указывается полное наименование вуза и кафедры, на которой выполнена курсовая работа. В средней части страницы – вид (курсовая) и тема работы. Далее, ближе к правому краю титульного листа приводятся фамилия и инициалы исполнителя, фамилия и инициалы научного руководителя, а также его учёное звание и учёная степень. В нижней части страницы – место выполнения работы и год ее написания без слова «год», например, Рыбница, 2014. Образец титульного листа приведен в приложении А.

В содержании приводятся заголовки работы и указываются страницы, с которых они начинаются. Заголовки содержания должны точно повторять заголовки текста, поэтому рекомендуется использовать возможности текстовых редакторов по автоматическому сбору оглавления.

Заголовки разделов и подразделов должны точно отражать содержание относящегося к ним текста. Рекомендуется в заголовок включать отглагольные существительные, которые отражали бы действия автора с изложенным в разделе (подразделе) материалом. Например, анализ, исследование, выбор, определение, разработка и т.д. Не рекомендуется в заголовок включать слова, отражающие общие понятия или не вносящие ясность в смысл заголовка, а также слова являющимися терминами узкоспециального профиля, сокращённые слова, аббревиатуры и формулы. Следует помнить, что разделы не должны существенно отличаться по объёму текста. Для нумерации текста используются арабские цифры.

Абзац – структурно-композиционная часть текста, которая состоит из одной или нескольких фраз, выражает логически завершённую мысль. Разделение текста на абзацы является приёмом, используемым для объединения ряда предложений, имеющих общий предмет изучения. Правильная разбивка текста курсовой работы на абзацы существенно

облегчает её чтение и осмысление. Абзацы одного подраздела или раздела по смыслу должны быть последовательно связаны друг с другом.

Вопросы подготовки введения, основной части работы, заключения рассмотрены в следующих разделах.

4.2. Подготовка введения

При подготовке введения используется понятийный аппарат исследования, рассмотренный ранее в соответствующем разделе, который применяется в определенной логической последовательности.

Как правило, во введении:

- обосновывается актуальность темы исследования;
- освещается степень изученности и научной разработанности темы;
- определяется объект, предмет, цели, задачи работы;
- определяется область применения, значимость результатов;
- описывается структура работы.

В таблице 4.1 приведена примерная структура введения для курсовых работ, выполняемых в процессе обучения, представлена последовательность размещения, названия разделов, а также рекомендации по объёму отдельных разделов введения.

Таблица 4.1

Примерная структура введения

Название раздела	Курсовая работа объёмом 20-35 стр.
Актуальность	Не более 1 стр.
Степень изученности и научной разработанности темы	Не более 1 стр.
Объект исследования	1 фраза
Предмет исследования	1 фраза
Цель работы	Одна развернутая фраза
Задачи работы	Ряд перечислений
Результаты	Несколько абзацев
Область применения, практическая значимость результатов	Не более 0,3 стр.
Описание структуры работы	Не более 0,5 стр.

Лексико-синтаксические конструкции, рекомендуемые для употребления во введении следующие:

Обоснована (что?) ...

Освещается (что?) ...

Определяются (что?) ...

Характеризуются (что?) ...
Указываются (что?) ...
Отмечается, что ...

4.3. Работа над основной частью

Основная часть – это содержательная часть работы, которая обычно состоит из двух разделов, в которых представлена постановка проблемы, предлагаемый путь и результаты её решения.

Содержание разделов должно точно соответствовать теме работы и полностью её раскрывать. Разделы работы должны показать умение автора сжато, логично и аргументированно излагать материал. Каждый раздел работы должен завершаться краткими выводами, которые подводят итоги исследований, приведённых в разделе.

При подготовке основного текста курсовой работы автор обязан делать ссылки на литературные источники, из которых он берет материалы или отдельные результаты.

Материалы, не влияющие на объяснение решения задачи, выносятся в приложения.

В процессе написания курсовой работы студент сталкивается с необходимостью структурирования больших объёмов информации. Одним из эффективных способов построения основной части, когда структура работы строится по задачам, является способ, основанный на принципе пирамиды, предложенный Барбарой Минто, консультантом компании McKinsey&Company [10]. Принцип пирамиды предполагает, что задачи, поставленные автором для достижения цели работы, выстраиваются в иерархию (пирамиду), что позволяет устанавливать между ними логические взаимосвязи.

При построении пирамиды предлагается придерживаться следующих правил:

1. Задачи любого уровня должны суммировать задачи, находящиеся на более низком уровне, и задачи любого уровня должны детализироваться на более мелкие на уровне ниже.
2. Задачи каждого раздела должны быть задачами одного порядка, то есть должны быть связаны между собой, что позволяет объединить их в группу (раздел).
3. Задачи раздела должны быть выстроены в определенном порядке, согласно внутренней логике.

На рис. 4.1 графически изображена основная часть курсовой работы, представленная в виде пирамиды.

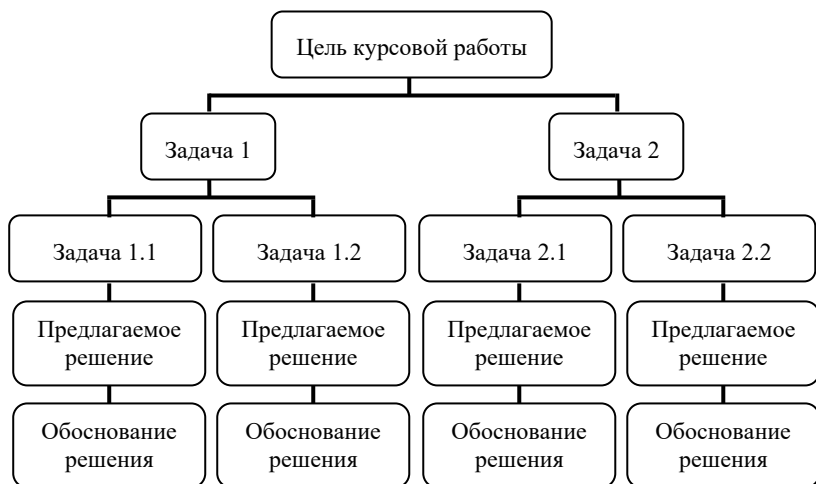


Рис. 4.1. Основная часть работы, представленная в виде пирамиды

Цель – стержень курсовой работы. Каждая задача (разделы) может быть детализирована на подзадачи. Для каждой задачи/подзадачи предлагается подход к решению и излагается обоснование решения (подразделы).

Основная часть исследовательской работы должна составлять не менее 70% её полного объёма. Язык и стиль изложения их изложения должны учитывать особенности научной речи.

Для научного текста характерны смысловая законченность, целостность и связность, которые достигаются специальными средствами организации связанного текста и использованием определений оценочного характера.

Одной из отличительных черт научной лексики является специальная терминология. Поскольку научный термин – это выражение сущности некоторого явления, к выбору терминов и определений следует относиться с большим вниманием. При написании работы недопустимо вместо терминов употреблять профессионализмы – условные понятия, употребляемые в среде узких специалистов. Рекомендуется завести словарь используемой терминологии, с указанием значений каждого термина и источника его определения.

Примеры лексико-синтаксических конструкций, рекомендуемые для изложения основной части и выводов курсовой работы, приведены в табл. 4.2.

Таблица 4.2

**Список лексико-синтаксических конструкций, рекомендуемых
при изложении основной части**

Информационный блок	Лексико-синтаксические конструкции
Текст основной части	Усиление внимания к проблеме ... связано в первую очередь с разработкой ... Теоретический анализ литературы позволяет выделить перспективное направление разработки ... Весьма полезными оказались результаты исследований ... , которые рассматривают ... В результате изучения был получен материал, анализ которого позволил заключить, что ... Чтобы обосновать ... , необходимо, ... прежде всего, выяснить ... Изложение названных проблем приведено в работах ... Собственные наблюдения и специальные исследования показали, что ... Поставленные задачи определяют следующие подходы к их решению... Выше изложенное подчеркивает необходимость рассмотрения вопроса о ... В работе предлагается ... Рассмотрим ... на примере ...
Выводы в разделах	Результаты проведенного анализа позволяют сделать следующие выводы ... Изложенное позволяет заключить, что ... В итоге следует подчеркнуть, что ... Наряду с этим необходимо отметить следующее ... В итоге рассмотрения данного вопроса можно утверждать, что ... Анализ ... позволяет сделать вывод о ...

Особую информационную нагрузку в тексте работы несут глаголы и глагольные формы. Основное место занимают глаголы несовершенного вида и формы настоящего времени, поскольку они не выражают отношение описываемого действия к моменту высказывания. Часто употребляется изъявительное наклонение глагола, редко – сослагательное наклонение, почти совсем не употребляется повелительное наклонение.

4.4. Подготовка заключения и формулирование выводов

Заключение подводит итог выполненному исследованию и должно содержать доказательство достижения поставленной цели курсовой работы, определённой во введении. В нём в сжатом виде раскрываются полученные результаты решения поставленных задач. Переходя от описания выделенных во введении проблем, определивших актуальность темы исследования, к описанию результатов решения поставленных задач, студент должен продемонстрировать устранение этих проблем. В этом случае работа будет иметь логически законченный результат. При этом заключение не должно повторять выводы и обобщения по разделам, выводы делаются по всей работе в целом.

В заключении также необходимо изложить собственный вклад в решение установленной проблемы, то есть отразить то существенное и новое, что отличает работу от ранее выполненных работ по рассматриваемой тематике. Кроме того, следует привести предложения по практическому использованию полученных результатов.

При разработке заключительной части курсовой работы рекомендуется:

1. Начинать заключение с вступительного слова, объемом 3-5 предложений по теоретической части, и только потом излагать выводы и предложения.
2. Обобщать основные результаты, полученные в ходе проведенного исследования.
3. Связывать приводимые результаты с поставленной целью и задачами.
4. Выделять собственный вклад автора.
5. Отмечать элементы новизны и практической значимости полученных результатов.
6. Приводить в краткой форме результаты практической апробации.
7. Излагать предложения по внедрению решений.
8. Обосновывать основные направления для дальнейшего развития исследования.

Рекомендуемый объём заключения курсовой работы – 2-3 страницы.

Одной из типовых ошибок при разработке заключения является механическое объединение выводов по отдельным разделам.

В табл. 4.3 приведены лексико-синтаксические конструкции, рекомендуемые для употребления в заключении.

Таблица 4.3

**Лексико-синтаксические конструкции, рекомендуемые
для употребления в заключении**

Информационный блок	Примеры лексико-синтаксических конструкций
Содержание заключения	В работе получены следующие результаты ... В работе поставленная цель ... достигнута путем ... В работе проведен анализ ... В работе обоснован выбор ... Исследование показало, что ... Полученные результаты базируются на ... Проведенное исследование позволяет сделать вывод, что... Анализ результатов исследования показал, что ... Основной собственный вклад в ... заключается в ... Новизна полученных результатов заключается в ... Преимущества разработанного ... состоят в ... Применение ... позволяет сократить (улучшить) ... Практическая ценность полученных результатов заключается в ... Направление ... представляется перспективным для дальнейшего исследования

После заключения в курсовой работе приводят библиографический список и приложения.

5. ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ

5.1. Правила оформления текстового материала

При оформлении текстового и иллюстративного материала курсовой работы рекомендуется руководствоваться соответствующими документами [1-7].

Текст курсовой работы разделяется на разделы и подразделы, которые нумеруют арабскими цифрами.

Каждый новый раздел начинается с новой страницы. Это относится и к другим основным структурным частям работы (введению, заключению, списку использованных источников, приложениям).

Разделам и подразделам дают заголовки, отражающие их содержание, при этом заголовки в оглавлении должны точно повторять заголовки в тексте. Не допускается сокращать или давать заголовки в другой формулировке.

Правила оформления текстового материала курсовой работы приведены в табл. 5.1.

Таблица 5.1

Правила оформления текстового материала курсовой работы

Группа правил	Содержание
Оформление основного текста	Формат страницы – А4 (210х297 мм), без рамки, книжная ориентация, левое поле – 30 мм, верхнее и нижнее поля – 20 мм, правое – 15 мм. Шрифт – Times New Roman, размер – 14 pt, строчные буквы, выравнивание – по ширине, межстрочный интервал – полуторный, абзацный отступ 1,25, цвет шрифта – чёрный.
Нумерация страниц	Арабские цифры, сквозная нумерация по всему тексту работы. Первая страница – титульный лист, на котором нумерация страниц не ставится, на следующей странице ставится цифра 2. Порядковый номер страницы – нижнее поле страницы, справа.
Оформление заголовков разделов и подразделов	Выравнивание – по центру. Все буквы заголовка раздела прописные. Точка в конце заголовка не ставится. Заголовок не подчеркивается. Переносы слов не допускаются. Если заголовок состоит из нескольких строк, то интервал между ними – одинарный.
Нумерация разделов и подразделов	Нумерация разделов – арабские цифры, в пределах всей работы. Номер подраздела состоит из номеров раздела и подраздела, разделенных точкой.

Каждый абзац должен начинаться с красной строки, содержать законченную мысль и состоять, как правило, из 4-5 предложений. Слишком крупный абзац затрудняет восприятие смысла и свидетельствует о неумении четко излагать мысль.

При печати работы необходимо установить запрет «висячих строк», то есть не допускается перенос на новую страницу или оставление на предыдущей странице одной строки абзаца, состоящего из нескольких строк. Следует избегать также оставления на последней строке абзаца части слова или даже одного целого слова. В этом случае лучше изменить формулировку предложения так, чтобы на последней строке абзаца оставалось не менее трех-четырех слов, либо использовать уплотненный текст, но не более чем на 0,3 pt.

Числовые значения величин с обозначением единиц физических величин и единиц счета следует писать цифрами, а числа без обозначения от единицы до девяти – словами, например, 3 км, но три программы. Если приводится ряд или диапазон числовых значений, выраженных в одной и той же единице физической величины, то ее указывают только после последнего числового значения, например; 1,50; 1,75; 2,00 В; от 10 до 100 Ом. Недопустимо отделять единицу физической величины от числового значения (переносить их на разные строки или страницы).

Размер символов в математических выражениях не должен превышать размер символов основного текста и должен выделяться курсивом.

Все используемые наименования на иностранных языках должны быть выделены курсивом, например, *Internet*.

Нельзя использовать аббревиатуры (сокращения) без первого полного упоминания в тексте.

5.2. Правила оформления иллюстраций

Иллюстративный материал может быть представлен рисунками, графиками, схемами, моделями, диаграммами и другим подобным материалом. Правила оформления рисунков следующие:

1. Иллюстрации нумеруют арабскими цифрами в пределах раздела.
2. Иллюстрация должна иметь название, которое приводится под рисунком, посередине строки, вслед за номером иллюстрации, шрифт – 12 pt.
3. На все иллюстрации должны быть приведены ссылки в тексте работы. При ссылке следует писать слово «рисунок» с указанием его номера, например, «... в соответствии с рисунком 1.1».

4. Иллюстрации размещают под текстом, в котором впервые дана ссылка на них, или на следующей странице, а при необходимости – в приложении.
5. В случае заимствования рисунка из какого-либо источника обязательна ссылка на этот источник, которая размещается после названия иллюстрации.

Подпись под иллюстрацией обычно имеет четыре основных элемента:

- наименование графического элемента, обозначаемое сокращенным словом «Рис.»;
- порядковый номер иллюстрации без значка №;
- тематический заголовок (после точки с прописной буквы);
- расшифровку рисунка, перед которой ставится знак двосточия, а между её элементами – точка с запятой.

Пример оформления иллюстрации приведен ниже (рис. 5.1).

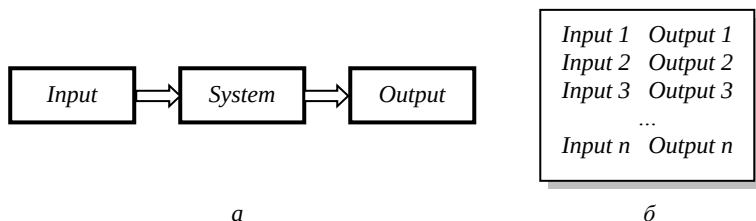


Рис. 5.1. Система ввода-вывода: *a* – взаимодействие системы с внешним миром;
b – общий вид экспериментальных данных

Если в курсовой работе только один рисунок, то его не нумеруют.

5.3. Правила оформления формул

При использовании в тексте курсовой работы формул рекомендуется придерживаться основных правил оформления (табл. 5.2).

В формулах в качестве символов следует применять обозначения, установленные соответствующими стандартами. Латинские буквы и символы должны быть выделены курсивом. Русские обозначения и цифры набираются прямым шрифтом. Пояснения символов и числовых коэффициентов, входящих в формулу, если они не пояснены ранее, должны быть приведены непосредственно под формулой. Пояснения каждого символа следует давать с новой строки в той последовательности, в которой они приведены в формуле. Первая строка

пояснения должна начинаться со слова «где», без двоеточия после него, без абзацного отступа.

Выше и ниже каждой формулы или уравнения должно быть оставлено не менее одной свободной строки одинарного интервала, размером шрифта 12 pt.

Таблица 5.2

Основные правила оформления формул

Группа правил	Содержание
Короткие формулы, а также формулы, не имеющие самостоятельного значения	Внутри текстовых строк. Формулы, не имеющие самостоятельного значения, не нумеруют.
Наиболее важные, а также длинные и сложные формулы	На отдельной строке по центру. Допускается переносить формулу на следующую строку на знаках выполняемых операций, причем знак в начале следующей строки повторяют. Важные формулы следует нумеровать арабскими цифрами сквозной нумерацией или в пределах раздела. Номер заключают в круглые скобки и записывают на уровне формулы справа.
Ссылки на формулы в тексте работы	Ссылки в тексте на порядковые номера формул дают в скобках, например, «... в формуле (5.1)».

Пример оформления формулы приведен ниже.

Плотность каждого образца ρ , кг/м, вычисляют по формуле

$$\rho = \frac{m}{V} \quad (5.1)$$

где m – масса образца, кг;

V – объём образца, м.

Набор формул необходимо осуществлять в соответствующем редакторе. Не допускается вставка формул в виде рисунков.

Если в курсовой работе только одна формула или уравнение, то их не нумеруют.

5.4. Правила оформления таблиц

В курсовой работе таблицы применяют для структурирования, лучшей наглядности материала, представления результатов сравнения показателей, характеристик, критериев и т.д.

Обычно таблица состоит из следующих элементов (рис. 5.2):

1. Тематический заголовок.
2. Нумерационный заголовок.
3. Заголовочная часть.
4. Хвостовая часть.

Характеристики целочисленных типов			Таблица 5	Нумерационный заголовок
				Тематический заголовок
Тип	Диапазон значений	Требуемая память (байт)		Головка
Integer	– 32768 .. 32767	2		Хвост
ShortInt	– 128 .. 127	1		
LongInt	– 2147483648 .. 2147483647	4		
Byte	0 .. 255	1		
Word	0 .. 65535	2		
Боковик		Прографка		

Рис. 5.2. Элементы таблицы

Тематический заголовок определяет цель и содержание таблицы. Он может быть опущен, когда связь таблицы с основным текстом очевидна).

Нумерационный заголовок (слово «таблица» и номер арабскими цифрами) необходим, когда в публикации несколько таблиц.

Заголовочная часть таблицы состоит из заголовков, определяющих содержание каждой графы.

Хвостовая часть таблицы – всё, что расположено под заголовочной частью. Сюда входят боковик и прографка.

Боковик – первая слева графа, включающая сведения, в которых обобщается содержание горизонтальных строк таблицы. Общие для нескольких строк понятия выносят в дополнительную строку.

Основные формулировки боковика пишутся с прописной буквы, формулировки подчиненных ступеней – со строчной. Точка в конце их не ставится. Если в нижестоящей строке повторяются слова вышестоящей, их заменяют кавычками;

Прографка – графы, данные которых выражают взаимозависимость между данными боковика и заголовочной части. Однородные числовые данные располагают так, чтобы в графе единицы находились под единицами, десятки – под десятками и т. д. Если в графе есть десятичные дроби, рекомендуется для удобства сравнения к цифровому обозначению целых чисел добавить нули после запятой.

Текстовые элементы прографки следует начинать с прописных букв. Вместо слов «более» и «менее» применяют условные обозначения (> <). Вместо слов «от» и «до» между цифрами следует ставить тире.

Заменять кавычками повторяющиеся в таблице цифры, математические знаки, знаки процента, обозначения марок материала, обозначения нормативных документов не допускается.

Таблицы, как и рисунки, имеют названия и порядковую нумерацию. Название должно отражать содержимое таблицы, быть точным и кратким. Правила оформления таблиц приведены в табл. 5.3.

Таблица 5.3

Правила оформления таблиц

Группа правил	Содержание
Размещение таблиц	Таблицы должны располагаться непосредственно за текстом, где они упоминаются, или на следующей странице.
Нумерация таблиц	Таблицы нумеруют арабскими цифрами сквозной нумерацией или в пределах раздела.
Оформление заголовков	Слово «Таблица» и её номер помещается сверху над таблицей справа, ниже по центру размещается название таблицы. Заголовки граф (столбцов) и строк следует писать с прописной буквы в именительном падеже (без сокращения слов), а подзаголовки граф со строчной буквы, если они составляют одно предложение с заголовком или с прописной буквы, если они имеют самостоятельное значение. Заголовки и подзаголовки указывают в единственном числе, через одинарный межстрочный интервал, без красной строки, выравнивание по центру. Шрифт – Times New Roman, кегль 12. В конце заголовков и подзаголовков таблиц точки не ставят.
Оформление основной части таблицы	Одинарный межстрочный интервал, без отступа, шрифт – Times New Roman, кегль 12.
Деление таблицы на части	Таблицу с большим количеством строк допускается переносить на следующие страницы. Название таблицы и её номер размещают один раз над первой частью таблицы. Над остальными частями таблицы указывают «Продолжение таблицы» и её номер. Допускается заменять головку или боковик соответственно номерами граф и строк, при этом нумеруют арабскими цифрами графы и (или) строки первой части таблицы.
Оформление заимствованных таблиц	Если таблица заимствована из литературных источников, то обязательна ссылка на источник данных. Ссылка помещается сразу после названия таблицы

Ссылка на таблицу в тексте обязательная и должна находиться до расположения самой таблицы. При ссылке следует писать сокращенно слово «таблица» с указанием номера, например, «табл. 5.1».

Если в курсовой работе только одна таблица, её не нумеруют.

5.5. Правила оформления библиографического списка

Библиографический аппарат курсовой работы представлен библиографическим списком и библиографическими ссылками.

Библиографический список является обязательной составляющей всех исследовательских работ и представляет собой нумерованный перечень источников, использованных при работе над темой: литературных, статистических и других видов.

Библиографический список размещается после основного текста работы. Он содержит библиографические записи использованных источников. Каждому источнику присваивается порядковый номер.

Библиографическая запись документа состоит из заголовка и библиографического описания.

Заголовок располагается перед библиографическим описанием и служит для упорядочения и поиска документов. Наиболее часто используют заголовок, содержащий имя автора произведения. В конце заголовка ставят точку.

Библиографическое описание содержит библиографические сведения о документе, приведенные по определенным правилам, устанавливающим наполнение и порядок следования областей и элементов, и предназначенные для идентификации и общей характеристики документа. В состав библиографического описания входят следующие области:

- область заглавия и сведений об ответственности;
- область издания;
- область специфических сведений;
- область выходных данных;
- область физической характеристики;
- область серии;
- область примечания;
- область стандартного номера и условий доступности.

Источником информации для составления библиографического описания является документ в целом. Наиболее часто используются сведения, приведенные на титульном листе. Библиографические сведения указывают в описании в том виде, в каком они даны в источнике информации.

Список должен содержать перечень источников, использованных при выполнении работы, как правило, список содержит не менее 15 источников.

Рекомендуется представить единый список литературы к работе в целом. Наиболее удобным является алфавитное расположение материала без разделения на части по видовому признаку (например, книги, статьи).

Произведения одного автора расставляются в списке по алфавиту заглавий или по годам публикации, в прямом хронологическом порядке. Затем все библиографические записи в списке последовательно нумеруются.

Список использованных источников размещается после текста работы и предшествует приложениям. Сведения о наличии списка литературы отражаются в «Содержании» [12, 13].

Нормативно-правовые документы

Заглавие официального документа (закон, постановление, указ и др.): сведения, относящиеся к заглавию, дата принятия документа // Название издания. – Год издания. – Номер (для журнала), Дата и месяц для газеты. – Первая и последняя страницы.

Примеры:

О военном положении [Текст]: Федеральный конституционный закон от 30 янв. 2002 г. № 1-ФКЗ // Собрание законодательства. – 2002. - № 5, (4 февр.). – С. 1485 – 1498 (ст. 375).

О правительственной комиссии по проведению административной реформы [Текст]: постановление Правительства РФ от 31 июля 2003 г. № 451 // Собрание законодательства. – 2003. – № 31. – Ст. 3150.

Нормативно-технические документы

Заглавие нормативно-технического документа: сведения, относящиеся к заглавию, обозначения ранее действующего документа, дата введения. – Год издания. – Объём.

Примеры:

ГОСТ 7.9 – 77. Реферат и аннотация. – М.: Изд-во стандартов, 1981. – 6 с.

ГОСТ 7.53 – 2001. Издания. Международная стандартная нумерация книг [Текст]. – Взамен ГОСТ 7.53 – 86; введ. 2002 – 07 – 01. – Минск: Межгос. Совет по стандартизации, метрологии и сертификации; Москва: Изд-во стандартов, 2002. – 3 с.

ГОСТ 1759. 5 – 87. Гайки. Механические свойства и методы [Текст]. – Взамен ГОСТ 1759 – 70; Введ. с 01.01.89 по 01.01.94. – М.: Изд-во стандартов, 1988. – 14 с.

Строительные нормы и правила: СНиП 2.01.07 – 85. Нагрузки и воздействия [Текст]: нормативно-технический материал. – М., 1987. – 36 с.

Авторские свидетельства, патенты

Примеры:

А.с. 1007970 СССР, МПК В 25 J 15/00. Устройство для захвата деталей [Текст] / Ваулин В.С., Калов В.К. (СССР). – 3350585/25-08; заявлено 23.11.81; опубл. 30.03.83, Бюл. 12. – С. 2.

Пат. 2187888 Российская Федерация, МПК Н 04 В 1/38, Н 04 J 13/00. Приемопередающее устройство [Текст] / Чугаева В.И.; заявитель и патентообладатель Воронеж. науч.-исслед. ин-т связи. – № 2000131736/09; заявл. 18.12.00; опубл. 20.08.02, Бюл. № 23 (II ч.). – 3 с.

Информационные листки

Примеры:

Барабин, А.И. Прогнозирование урожая семян ели методом подсчета числа женских почек [Текст] / А.И. Барабин. – Архангельск, 1971. – [4] с. – (Информ. листок о науч.-техн. достижении / АрхЦНТИ; N 71-62).

Мурманская, Н.П. Опыт хранения семян сосны и ели [Текст] / Н.П. Мурманская, Г.С. Тутыгин. – Архангельск, 1976. – [4] с. – (Информ. листок о науч.-техн. достижении / АрхЦНТИ; N 160-76).

Книги

Однотомное издание

Автор. Заглавие: сведения, относящиеся к заглавию (см. на титуле) / сведения об ответственности (авторы); последующие сведения об ответственности (редакторы, переводчики, коллективы). – Сведения об издании (информация о переиздании, номер издания). – Место издания: Издательство, Год издания. – Объем. – (Серия).

Примеры:

Если у издания один автор, то описание начинается с фамилии и инициалов автора. Далее через точку «.» пишется заглавие. За косой чертой «/» после заглавия имя автора повторяется, как сведение об ответственности.

Лукаш, Ю.А. Индивидуальный предприниматель без образования юридического лица [Текст] / Ю.А. Лукаш. – Москва: Книжный мир, 2002. – 457 с.

Если у издания два автора, то описание начинается с фамилии и инициалов первого автора. За косой чертой «/» после заглавия сначала указывается первый автор, а потом через запятую – второй автор.

Бычкова, С.М. Планирование в аудите [Текст] / С.М. Бычкова, А.В. Газорян. – Москва: Финансы и статистика, 2001. – 263 с.

Если у издания три автора, то описание начинается с фамилии и инициалов первого автора. За косой чертой «/» после заглавия сначала указывается первый автор, а потом через запятую – второй и третий авторы.

Краснова, Л.П. Бухгалтерский учет [Текст]: учебник для вузов / Л.П. Краснова, Н.Т. Шалашова, Н.М. Ярцева. – Москва: Юрист, 2001. – 550 с.

Если у издания четыре автора, то описание начинается с заглавия. За косой чертой указываются все авторы.

Лесоводство [Текст]: учебное пособие к курсовому проектированию/З.В. Ерохина, Н.П. Гордина, Н.Г. Спицына, В.Г. Атрохин. – Красноярск: Изд-во СибГТУ, 2000. – 175 с.

Если у издания пять авторов и более, то описание начинается с заглавия. За косой чертой указываются три автора и др.

Логика [Текст]: учебное пособие для 10-11 классов / А.Д. Гетманова, А.Л. Никифоров, М.И. Панов и др. – Москва: Дрофа, 1995. – 156 с.

Если у издания есть один или несколько авторов и также указаны редакторы, составители, переводчики и т.п., то информация о них указывается в сведении об ответственности, после всех авторов перед точкой с запятой «;».

Ашервуд Б. Азбука общения [Текст] / Б. Ашервуд; пер. с англ. И.Ю.Багровой и Р.З. Пановой, науч. ред. Л.М. Иньковой. – Москва: Либеря, 1995. – 175 с.

Если у издания нет автора, но указаны редакторы, составители, переводчики и т.п., то описание начинается с заглавия. За косой чертой после заглавия сразу пишутся редакторы, составители и т.п. с указанием функции.

Логопедия [Текст]: учебник для студ. дефектолог. фак. пед. вузов / ред. Л.С. Волкова, С.Н. Шаховская. – 3-е изд., перераб. и доп. – Москва: Гуманит. изд. центр. ВЛАДОС, 2002. – 680 с.

Если у издания нет автора, редакторов и т.п., то после заглавия сразу идет информация об издании после точки и тире «. –».

Иллюстрированный словарь английского и русского языка с указателями [Текст]. – Москва: Живой язык, 2003. – 1000 с.

Многотомные издания

Автор. Заглавие издания: сведения, относящиеся к заглавию (см. на титуле) / Сведения об ответственности (авторы); последующие сведения об ответственности (редакторы, переводчики, коллективы). – Город издания: Издательство, Год начала издания – год окончания издания. – (Серия).

Обозначение и номер тома: Заглавие тома: сведения, относящиеся к заглавию. – Год издания тома. – Объём;

Обозначение и номер тома: Заглавие тома: сведения, относящиеся к заглавию. – Год издания тома. – Объём. и т.д.

или:

Автор. Заглавие издания: сведения, относящиеся к заглавию (см. на титуле) / Сведения об ответственности (авторы); последующие сведения об ответственности (редакторы, переводчики, коллективы). – Город издания: Издательство, Год начала издания – год окончания издания. – Количество томов. – (Серия).

Примеры:

Горожанин, А.В. Российская полиция на страже имперской государственности: монография [Текст]: в 2-х т. / А.В. Горожанин; Мин-во юстиции РФ, Самар. юрид. ин-т. – Самара, 2004. – 91 с.

Т.1: Полиция как столп российской имперской государственности (XVIII – первая половина XIX в.) – 258 с.

Т.2: Российская империя и ее полиция: рассвет и закат – 166 с.

или

Горожанин, А.В. Российская полиция на страже имперской государственности: монография [Текст]: в 2-х т. / А.В. Горожанин; Мин-во юстиции РФ, Самар. юрид. ин-т. – Самара, 2004. – 91 с. – 2 т.

Неопубликованные документы

Диссертации

Автор. Заглавие: сведения, относящиеся к заглавию (см. на титуле): шифр номенклатуры специальностей научных работников: дата защиты: дата утверждения / сведения об ответственности (автор); последующие сведения об ответственности (коллектив). – Место написания, Дата написания. – Объём.

Примечания:

В сведениях, относящихся к заглавию, приводят сведения о том, что данная работа представлена в качестве диссертации, а также сведения об ученой степени, на соискание которой представлена диссертация. Сведения приводят в сокращенном виде.

Например:

дис. ... канд. пед. наук

дис. ... д-ра техн. наук

Примеры:

Белозеров, И.В. Религиозная политика Золотой Орды на Руси в XIII-XIV вв. [Текст]: дис ... канд. ист. наук: 07.00.02: защищена 22.01.02: утв. 15.07.02 / Белозеров Иван Валентинович. – Москва, 2002. – 215 с.

Автореферат диссертации

Автор. Заглавие: сведения, относящиеся к заглавию (см. на титуле): шифр номенклатуры специальностей научных работников: дата

защиты: дата утверждения / сведения об ответственности (коллектив). – Место написания. – Объём.

Примечания:

В сведениях, относящихся к заглавию, приводят сведения о том, что данная работа представлена в качестве автореферата диссертации на соискание учёной степени. Сведения приводят в сокращённом виде.

Например:

автореф. дис. ... канд. физ. наук

автореф. дис. ... д-ра пед. наук

Примеры:

Александров, А.А. Анализ и оценка оперативной обстановки в республике, крае, области (правовые и организационные аспекты) [Текст]: автореф. дис. на соиск. учен. степ. канд. юрид. наук (12.00.11) / Александров Александр Александрович; Акад. упр. МВД России. – Москва, 2004. – 26 с.

Электронные ресурсы

Электронный ресурс локального доступа (CD)

Автор. Заглавие [Электронный ресурс]: сведения, относящиеся к заглавию / сведения об ответственности (авторы); последующие сведения об ответственности (редакторы, переводчики, коллективы). – Обозначение вида ресурса («электрон. дан.» и/или «электрон. прогр.»). – Место издания: Издательство, Год издания. – Обозначение материала и количество физических единиц. – (Серия).

Примечания:

Описание электронного ресурса в области «Автор» и «Сведения об ответственности» осуществляется по правилам описания книжного издания.

Обозначение материала приводят сразу после заглавия в квадратных скобках: [Электронный ресурс]

Примеры:

Родников, А.Р. Логистика [Электронный ресурс]: терминологический словарь. – / А.Р. Родников. – Электронные данные. – Москва: ИНФРА-М, 2000. – 1 эл. опт. диск (CD-ROM).

Энциклопедия классической музыки [Электронный ресурс]. – Электрон. дан. – Москва: Коминфо, 2000. – 1 эл. опт. диск (CD-ROM).

Электронный ресурс удаленного доступа (Internet)

Автор. Заглавие [Электронный ресурс]: сведения, относящиеся к заглавию / сведения об ответственности (авторы); последующие сведения об ответственности (редакторы, переводчики, коллективы). – Обозначение вида ресурса («электрон. текст. дан.»). – Место издания:

Издательство, Дата издания. – Режим доступа: URL. – Примечание («Электрон. версия печ. публикации»).

Примечания:

Описание электронного ресурса в области «Автор» и «Сведения об ответственности» осуществляется по правилам описания книжного издания.

Обозначение материала приводят сразу после заглавия в квадратных скобках: [Электронный ресурс].

Если описывается сайт в целом, то область «Дата издания» будет выглядеть следующим образом: Год начала издания – год окончания издания.

Примеры:

Исследовано в России [Электронный ресурс]: многопредмет. науч. журн. / Моск. физ.-техн. ин-т. – Электрон. журн. – Долгопрудный: МФТИ, 1998. – Режим доступа к журн.: <http://zhurnul.milt.riss.ru>

Шпринц, Лев. Книга художника: от миллионных тиражей – к единичным экземплярам [Электронный ресурс] / Л. Шпринц. – Электрон. текстовые дан. – Москва: [б.и.], 2000. – Режим доступа: <http://atbook.km.ru/news/000525.html>, свободный.

Составные части документов

Сведения о статье // Сведения об источнике статьи. – Сведение о местоположении статьи в документе.

Статья из книги

Автор. Заглавие статьи: сведения, относящиеся к заглавию / сведения об ответственности (авторы статьи) // Заглавие книги: сведения, сведения, относящиеся к заглавию / сведения об ответственности (авторы книги); последующие сведения об ответственности (редакторы, переводчики, коллективы). – Место издания: Издательство, год издания. – Местоположение статьи (страницы).

Примечание:

Сведения об издательстве в области выходных данных книг можно упустить.

Иванов, С.А. Маркетинг и менеджмент [Текст] / С.А. Иванов // Статьи о классиках. – Москва, 2002. – С. 12-34.

Статья из сборника

Думова, И.И. Инвестиции в человеческий капитал [Текст] / И.И. Думова, М.В. Колесникова // Современные аспекты регионального развития: сб. статей. – Иркутск, 2001. – С. 47-49.

Баданина, Л.А. Расчет процесса фильтрации жидкости в древесине при автоклавной пропитке [Текст] / Л.А. Баданина // Наука –

Северному региону: сб. науч. тр. / АГТУ. – Архангельск, 2005. – Вып. 62. – С. 8-12.

Статья из газеты

Автор. Заглавие статьи: сведения, относящиеся к заглавию / сведения об ответственности (авторы статьи) // Название газеты. – Год выпуска. – Число и месяц выпуска. – Местоположение статьи (страницы).

Николаева, С. Будем читать. Глядишь, и кризис пройдет...[Текст] / С. Николаева // Северный комсомолец. – 2009. - № 13. – С. 9.

Рысев, В. Приоритет – экология [Текст] / В. Рысев // Волна. – 2004. – 4 марта. – С. 13.

Статья из журнала

Автор. Заглавие статьи: сведения, относящиеся к заглавию / сведения об ответственности (авторы статьи) // Название журнала. – Год выпуска. – Номер выпуска. – Местоположение статьи (страницы).

Примечание:

Если статья размещены в двух и более журналах, то сведения о ее местоположении в каждом из номеров отделяют точкой с запятой.

Примеры:

Тарасова, Н.Г. Смена парадигм в развитии теории и практики градостроительства [Текст] / Н.Г. Тарасова // Архитектура и строительство России. – 2007. – № 4. – С. 2-7.

Казаков, Н.А. Запоздалое признание [Текст] / Н.А. Казаков // На боевом посту. – 2000. – № 9. – С. 64-67; № 10. – С. 58-71.

Статья из продолжающихся изданий

Автор. Заглавие статьи: сведения, относящиеся к заглавию / сведения об ответственности (авторы статьи) // Заглавие издания. Название серии. – Год издания. – Номер выпуска: Заглавие выпуска. – Местоположение статьи (страницы).

Примеры:

Белох, Н.В. Доходы, предложение и цены – проблема сбалансированности [Текст] / Н.В. Белох, Н.Я. Петраков, В.П. Русаков // Известия АН СССР. Сер. экономическая. – 1982. – № 2. – С. 71-77.

Белова, Г.Д. Некоторые вопросы уголовной ответственности за нарушение налогового законодательства [Текст] / Г.Д. Белова // Актуальные проблемы прокурорского надзора /Ин-т повышения квал. рук. кадров Генер. прокуратуры Рос. Федерации. – 2001. – Вып. 5: Прокурорский надзор за исполнением уголовного и уголовно-процессуального законодательства. Организация деятельности прокуратуры. – С. 46-49.

Рецензия

Автор рецензии. Заглавие статьи: сведения, относящиеся к заглавию / сведения об ответственности (авторы статьи) // Название журнала. – Год выпуска. – Номер выпуска. – Местоположение статьи (страницы). – Рец. на кн.: Описание книги.

или

Описание книги. – Рец. Заглавие статьи: сведения, относящиеся к заглавию / сведения об ответственности (авторы статьи) // Название журнала. – Год выпуска. – Номер выпуска. – Местоположение статьи (страницы).

Примечание:

Описание издания, на которое написана рецензия, осуществляется по правилам описания книжного издания.

Примеры:

Воскресенский, С.В. В помощь учителю и ученику [Текст] // Северный край. – 1999. – 30 сент. – Рец. на кн.: Карта Ярославской области. География. История [Карты] / отв. Ред. Е.Ю. Колобовский. – Ярославль, 1999.

Пономаренков, В.А. Особенности расследования «цыганских» преступлений: учебное пособие / В.А. Пономаренков, И.А. Пономаренкова. – Москва: Изд-во МГПУ, 2002. – 76 с. – Рец. Наумова, Н.А. О необычном пособии для правоохранительных органов [Текст] / Е.А. Наумова // Вестник Моск. гор. пед. ун-та. – 2003. – № 2. – С. 273.

5.6. Правила оформления библиографических ссылок

Библиографическая ссылка является частью текста курсовой работы и служит источником библиографической информации об использованных литературных и других источниках. Библиографическая ссылка содержит библиографические сведения о цитируемом, рассматриваемом или упоминаемом в тексте документа другом документе (его составной части или группе документов), необходимые и достаточные для его идентификации, поиска и общей характеристики.

По составу элементов библиографическая ссылка может быть полной или краткой, в зависимости от вида ссылки, её назначения, наличия библиографических сведений о документе.

Полная ссылка содержит совокупность библиографических сведений о документе.

Краткая ссылка предназначена только для поиска документа.

По месту расположения библиографические ссылки классифицируют на следующие группы:

1. Внутритекстовые, помещенные в текст документа.

2. Подстрочные, вынесенные из текста вниз страницы документа (в сноску).
3. Затекстовые, вынесенные за текст документа или его части (в выноску).

Внутритекстовые ссылки обычно включают в текст, если в работе отсутствует библиографический перечень или не могут быть использованы подстрочные ссылки.

Подстрочные ссылки используются, когда в процессе чтения основного текста необходимо получить справочную информацию о литературном источнике, а размещение ссылки внутри текста приводит к усложнению чтения.

Правила оформления библиографических ссылок приведены в табл. 5.4.

Таблица 5.4

Правила оформления библиографических ссылок

Вид ссылки	Содержание
Внутритекстовая	Заключают в круглые скобки, а предписанный знак точку и тире, разделяющий области библиографического описания, как правило, заменяют точкой.
Подстрочная	Оформляется как примечание, вынесенное из текста документа вниз страницы. При нумерации применяют единообразный порядок: сквозную нумерацию по всему тексту, в пределах каждого раздела, части и т. п. Для связи с текстом документа используют знак сноски. При наличии в тексте библиографических сведений, идентифицирующих электронный ресурс удаленного доступа, допускается указывать только его электронный адрес. Для обозначения электронного адреса используют аббревиатуру "URL" (<i>Uniform Resource Locator</i> – унифицированный указатель ресурса).
Затекстовая	Оформляется как перечень библиографических записей, помещенный после текста документа или его составной части. Используется сплошная нумерация для всего текста работы или для отдельных разделов, частей и т.п. Для связи с текстом порядковый номер библиографической записи указывают в знаке выноски, который набирают надстрочным шрифтом, или в отсылке, которую приводят в квадратных скобках в строку с текстом документа. Если ссылку приводят на конкретный фрагмент текста документа, в отсылке указывают порядковый номер библиографической записи и страницы, на которых помещен объект ссылки. Сведения разделяют запятой.

Совокупность затекстовых библиографических ссылок не является библиографическим списком, помещаемым после основного текста курсовой работы.

Примеры оформления библиографических ссылок приведены ниже.

Внутритекстовая ссылка

В тексте: (Меженный, О.А. Самоучитель *Turbo Pascal* [Текст]. – М.: Диалектика, 2008. – 330 с.);

Ссылка на цитату: (Меженный, О.А. Самоучитель *Turbo Pascal* [Текст]. – М.: Диалектика, 2008. – С. 40).

Подстрочная ссылка

Пример 1:

В тексте: О.А. Меженный в своей работе «Самоучитель *Turbo Pascal*»¹ отмечает, что вопросы ...

В ссылке: ¹ Меженный О.А., Самоучитель *Turbo Pascal* [Текст]. – М.: Диалектика, 2008. – 330 с.

Пример 2:

В тексте: Издание о высоких технологиях ¹.

В ссылке: ¹URL: <http://www.cnews.ru/>

Затекстовая ссылка

Пример 1:

В тексте: В своей работе «Самоучитель *Turbo Pascal*» О.А. Меженный рассматривает...¹

В ссылке: ¹ Меженный, О.А. Самоучитель *Turbo Pascal* [Текст]. – М.: Диалектика, 2008. – 330 с.

Пример 2:

В тексте: Операции и стандартные подпрограммы для типов данных рассмотрены в работе О.А. Меженного [1].

В ссылке: 1. Меженный, О.А. Самоучитель *Turbo Pascal* [Текст]. – М.: Диалектика, 2008. – 330 с.

5.7. Правила оформления приложений

Материал, дополняющий основной текст курсовой работы допускается помещать в приложениях. Примерами таких материалов служат графический материал, таблицы, формулы, блок-схемы, рисунки, инструкции, разработанные в ходе выполнения исследований, техническое задание, акты внедрения, образцы документов и пр.

Приложение оформляют как продолжение текста курсовой работы на последующих его листах или выпускают в виде самостоятельного документа.

В тексте на все приложения должны быть даны ссылки. Приложения располагают в порядке ссылок на них в тексте работы.

Каждое приложение должно начинаться с новой страницы, в правом верхнем углу которой указывается обозначение приложения, а в отдельной строке – заголовок приложения. На все приложения в тексте работы обязательно должны быть ссылки. Например, см. Приложение А. Приложения обозначаются:

- прописными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ь. Например, Приложение А;
- буквами латинского алфавита, за исключением букв I и O;
- арабскими цифрами.

Если приложений много, они оформляются отдельной книгой, титульный лист которой должен быть оформлен так же, как титульный лист курсовой работы, дополнительно ниже темы должно быть написано прописными буквами слово «ПРИЛОЖЕНИЯ». Нумерация приложения начинается с титульного листа приложения, но номер на титульном листе не проставляется. Нумерация приложений, если их более одного, может быть сквозной или для каждого приложения отдельной.

Листы работы, имеющие формат более А4, помещаются в качестве приложений и складываются по формату листов курсовой работы.

В содержании курсовой работы должно присутствовать указание на наличие приложений. Список приложений размещается после библиографического списка.

5.8. Правила оформления блок-схем

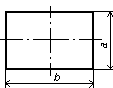
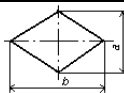
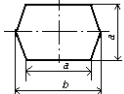
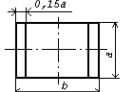
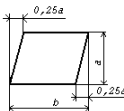
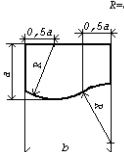
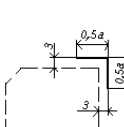
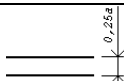
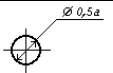
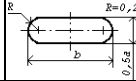
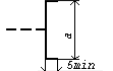
Блок-схема – распространенный тип схем (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности.

При выполнении блок-схем алгоритмов, программ, данных и систем следует руководствоваться ГОСТ 19.102-77.

Наименование, функции и обозначение основных элементов блок-схем приведены в табл. 5.5.

Таблица 5.5

Наименование, функции и обозначение основных элементов блок-схем

Наименование	Функция	Обозначение и размеры
Процесс	Выполнение операций или группы операций, в результате которых изменяется значение, форма представления или расположение данных.	
Решение	Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий.	
Модификация	Выполнение операций, меняющих команды или группу команд, изменяющих программу.	
Предопределенный процесс	Использование ранее созданных и отдельно описанных алгоритмов или программ.	
Ввод-вывод	Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод).	
Документ	Ввод-вывод данных, носителем которых служит бумага.	
Файл	Представление организованных на основе общих признаков данных, характеризующих в совокупности некоторый объект обработки данных. Символ используется в сочетании с символами конкретных носителей данных, выполняющих функции ввода-вывода.	
Линия потока	Указание последовательности между символами	
Параллельные действия	Начало или окончание двух и более одновременно выполняемых операций.	
Соединитель	Указание связи между прерванными линиями потока, связывающими символами.	
Пуск-останов	Начало, конец, прерывание процесса обработки данных или выполнения программы.	
Комментарий	Связь между элементом схемы и пояснением.	

Размер a для изображения элементов блок-схем должен выбираться из ряда 10, 15, 20 мм. Допускается увеличивать размер a на число, кратное 5. Размер b равен $1,5a$.

При ручном выполнении схем алгоритмов и программ для символов процесс, решение, модификация, предопределенный процесс, ввод-вывод, и пуск-останов допускается устанавливать b равным $2a$.

При выполнении условных графических обозначений автоматизированным способом размеры геометрических элементов символов округляются до значений, определяемых техническими возможностями используемых устройств.

Блок-схемы, в основном, используют для того, чтобы облегчить процесс программирования, поэтому всегда при решении любой задачи следует сначала составлять алгоритм программы, а потом уже программировать решение задачи на выбранном языке.

На языке блок-схем каждый шаг алгоритма описывается с помощью соответствующей фигуры, а последовательность выполнения шагов определяется линиями-связями. Блок схемы читаются сверху вниз и слева направо.

Блок-схемы полезны тем, что обеспечивают легкую «читаемость» алгоритма. Однако это не всегда так: стоит попытаться нарисовать блок-схему для более-менее сложного алгоритма, как она разрастается до невероятных размеров и теряет все свое наглядное преимущество. Поэтому блок-схемы хороши в структурном программировании для описания коротких алгоритмов и наиболее удобны для записи алгоритмов на начальных стадиях обучения программированию.

6. ЗАЩИТА КУРСОВОЙ РАБОТЫ

6.1. Подготовка доклада

Доклад представляет собой краткое изложение сути проведённого исследования, полученных результатов, их теоретической и практической значимости. Его подготовка включает:

1. Обдумывание структуры и содержания.
2. Разработку плана.
3. Написание текста доклада.
4. Репетицию выступления.

В структурном отношении доклад обычно делится на три части: введение, основную часть, заключение.

В совокупности эти части должны составлять единое целое и каждая часть должна быть логическим продолжением предыдущей. Принцип построения доклада следующий: сначала приводится общая информация об исследовании, затем излагается ход и содержание проведенного исследования и в заключении подводятся итоги. Рекомендации по содержанию частей доклада приведены в табл. 6.1.

Таблица 6.1

Рекомендации по содержанию доклада

Раздел доклада	Содержание
Введение	Основная цель введения доклада информировать о содержании исследования и вызвать интерес к проделанной работе. В нём в сжатой форме повторяется введение курсовой работы: обосновывается актуальность темы, описывается задание, даётся оценка степени изученности и научной проработанности темы, определяется объект, предмет и цель исследования, комплекс задач, которые необходимо было решить, чтобы цель была достигнута. Введение должно быть кратким и исчерпывающе информативным.
Основная часть	Вторая часть доклада – самая большая по объёму. В ней, в последовательности, установленной логикой проведенного исследования, излагается суть выполненной работы: постановка и решение задач, аргументация полученных результатов. В этой части необходимо подчеркнуть собственный вклад в проведенном исследовании, определить новизну полученных результатов.
Заключение	Завершающая часть аналогична по построению заключению курсовой работы. Здесь приводятся общие выводы, основные рекомендации, характеризуется новизна полученных результатов, определяются перспективы дальнейшего развития темы и полученных результатов.

Каждый доклад имеет свою специфику, отражающую особенности проведенного исследования. Вместе с тем, структура доклада имеет общий характер. Ниже приведен примерный план доклада.

1. Обоснование актуальности темы.
2. Постановка задания курсовой работы.
3. Обзор и анализ известных решений проблемы, их недостатки.
4. Объект и предмет исследования.
5. Цель и задачи исследования.
6. Теоретическая база, методы и инструменты исследования.
7. Основные положения, выносимые на защиту.
8. Предлагаемое решение задач исследования с обоснованием.
9. Анализ достигнутых результатов. Новизна, практическая значимость полученных результатов.
10. Общее заключение и выводы.

Для подготовки к выступлению доклад рекомендуется оформить письменно. Содержание доклада необходимо согласовать с научным руководителем.

Доклад следует прорепетировать перед коллегами, друзьями, родственниками. В процессе репетиции рекомендуется осуществить хронометраж выступления, чтобы не выходить за рамки установленного времени доклада, отметить в докладе ориентиры, чтобы можно было следить за временем по ходу выступления. Рекомендации по распределению времени выступления даны ниже в табл. 6.2.

Таблица 6.2

Рекомендации по распределению времени выступления

Раздел доклада	Рекомендованное время
Введение	2 минуты
Основная часть	7 минут
Заключение	1 минута
Общий объём времени	10 минут

Во время выступления пользоваться текстом доклада не следует, поэтому все ключевые вопросы должны быть отражены в презентации, которая помогает в процессе выступления. Особенности подготовки презентации рассмотрены в следующем разделе.

6.2. Подготовка презентации

Презентация является эффективным способом изложения сути и результатов проведенного исследования. Её цель на защите результатов курсовой работы – проинформировать о содержании исследования и

убедить в достоверности и обоснованности полученных результатов, предлагаемых рекомендаций. Стиль проведения презентации – формальный.

Подготовка презентации включает следующие этапы:

1. Обдумывание структуры и содержания.
2. Разработка плана.
3. Написание текста презентации.
4. Подготовка слайдов презентации.
5. Репетиция выступления.

Презентация должна ясно и веско доводить до аудитории центральную идею исследования и полученные результаты. Основой подготовки презентации служит доклад. Структура презентации аналогична структуре и плану доклада. Рекомендации по структуре и содержанию презентации приведены в табл. 6.3.

Таблица 6.3

Примерная структура и содержание презентации результатов курсовой работы

Слайд презентации	Содержание
Слайд 1	Титульный лист: ВУЗ, филиал, выпускающая кафедра. Тема курсовой работы. ФИО докладчика. ФИО, степень, звание и должность научного руководителя.
Слайд 2	Актуальность темы, включая установленную проблему. Объект и предмет исследования.
Слайд 3	Цель и задачи исследования.
Слайд 4	Теоретическая база, методы и инструменты исследования.
Слайд 5	Основные положения, выносимые на защиту.
Слайд 6-8	Содержание исследования: предлагаемое решение задач исследования с обоснованием.
Слайд 9	Анализ достигнутых результатов. Новизна и практическая значимость полученных результатов.
Слайд 10	Общее заключение и выводы. Перспективы развития темы и полученных результатов.

Рекомендации по распределению времени выступления приведены в предыдущем разделе.

Презентация должна быть наглядной. Материал рекомендуется представлять в структурном, графическом и схематичном виде. В тексте следует избегать длинных предложений.

При подготовке слайдов рекомендуется придерживаться следующих правил:

1. Слайды должны быть простыми, не перегруженными текстом и излишними данными.
2. Желательно использовать шаблон со светлым фоном, который не отвлекает внимание от содержания слайда.
3. Текст должен легко читаться, рекомендуемый размер шрифта не ниже 20pt, цвет – синий или черный. Текст должен быть написан простыми, короткими предложениями, отражать основные положения доклада, существенную информацию. Рекомендуется употреблять общепринятую терминологию, пояснять узкоспециализированные понятия.
4. Не следует использовать в презентации звуковые эффекты и большое количество анимации.
5. Рисунки, графики, таблицы должны иметь название.
6. Содержание слайдов должно соответствовать выступлению.

Дополнительные материалы, подкрепляющие выступление и не вошедшие в презентацию, могут быть оформлены в виде раздаточного материала к докладу. Примером таких материалов могут служить основные тезисы презентации, блок-схемы, примеры разработанных документов и др. В случае наличия раздаточного материала в процессе выступления необходимо делать ссылку на соответствующий материал.

6.3. Защита курсовой работы

Студент обязан написать, сдать и защитить курсовую работу в соответствии с предъявляемыми к ней требованиями и согласно установленному графику выполнения курсовой работы.

Курсовая работа допускается к защите при условии законченного оформления и допуска научного руководителя. В случае недопуска курсовой работы к защите, руководитель курсовой работы проставляет в экзаменационной ведомости студенту неудовлетворительную оценку.

Защита курсовых работ должна быть проведена до начала экзаменационной сессии.

Защита курсовой работы может проводиться:

1. В форме оценивания научным руководителем.
2. В форме публичной защиты на открытом заседании комиссии (по решению кафедры).

В первом случае проверка работы завершается оценкой научного руководителя, которая доводится до сведения студента.

Во втором случае защита курсовой работы проходит на открытом заседании специальной комиссии. В состав комиссии включаются

преподаватели кафедры, на которой выполняется курсовая работа, а также могут входить преподаватели других кафедр. На защиту курсовой работы приглашаются научный руководитель курсовой работы, преподаватели, студенты.

К защите студенту необходимо подготовить демонстрационный материал, оформленный в виде презентации, и доклад, с помощью которого он сможет чётко и кратко изложить основные положения курсовой работы. Презентация курсовой работы и доклад обсуждаются и согласовываются с научным руководителем.

Процедура защиты курсовой работы в случае публичной защиты на открытом заседании комиссии представлена на рис. 6.1.

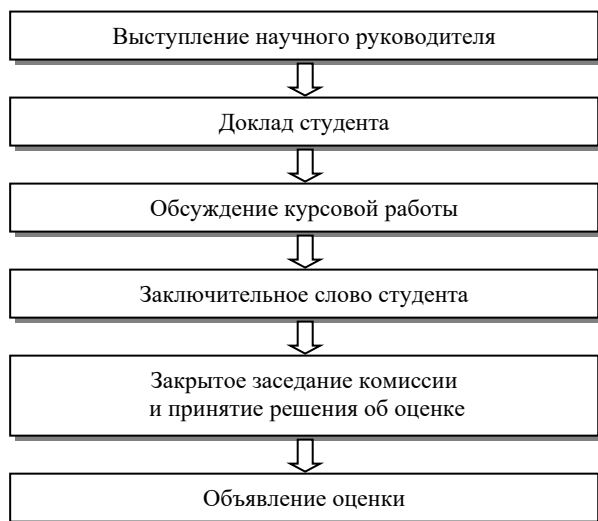


Рис. 6.1. Процедура защиты курсовой работы

Защита курсовой работы начинается с того, что научный руководитель приглашает студента к защите.

Студент делает доклад, в котором излагает актуальность темы курсовой работы, характеризует поставленные цель и задачи, методы исследований, обосновывает предлагаемое решение, аргументирует полученные результаты и выводы. В процессе доклада студент использует подготовленные презентационные материалы.

По окончании доклада студенту задают вопросы, связанные с темой курсовой работы. Вопросы могут быть заданы не только членами комиссии, но и всеми присутствующими на защите.

После ответа студента на заданные ему вопросы происходит обмен мнениями, в котором могут принять участие члены комиссии, научный руководитель и все желающие. В заключении автору курсовой работы предоставляется возможность в кратком выступлении защитить или разъяснить положения, которые встретили возражения, ответить на сделанные замечания и рекомендации, привести дополнительные материалы.

После заключительного слова студента процедура защиты курсовой работы считается оконченной.

На закрытом заседании комиссии подводятся итоги защиты курсовой работы и принимается решение об её оценке.

Итоговая оценка за курсовую работу формируется на основе оценок:

1. Научного руководителя за качество работы, степень её соответствия требованиям, предъявляемым к курсовым работам.
2. Членов комиссии кафедры за содержание работы, её защиту, включая доклад, ответы на вопросы и замечания научного руководителя.

Решение принимается большинством голосов.

Научный руководитель объявляет всем присутствующим оценку, заседание закрывается.

Студент, получивший неудовлетворительную оценку за курсовую работу, считается имеющим академическую задолженность, которую должен ликвидировать в установленном порядке.

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам [Текст]. – Москва: Госстандарт России, 1995. – 28 с.
2. ГОСТ 7.05-2008. Библиографическая ссылка. Общие требования и правила составления [Текст]. – М.: Стандартинформ, 2008. – 23 с.
3. ГОСТ 7.1-2003. Библиографическая запись. Общие требования и правила составления [Текст]. – М.: ИПК Изд-во стандартов, 2004. – 47 с.
4. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления [Текст]. – М.: Изд-во стандартов, – 2001. – 22 с.
5. ГОСТ 7.80-2000. Библиографическая запись. Заголовок. Общие требования и правила составления [Текст]. – Минск: ИПК Изд-во стандартов, 2000. – 11 с.
6. ГОСТ 7.82-2001. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления [Текст]. – Минск: ИПК Изд-во стандартов, 2001. – 23 с.
7. ГОСТ 19.102-77. Единая система программной документации. Стадии разработки [Текст]. – М.: Госстандарт России, 1978. – 4 с.
8. Ворона, В.А. Научные исследования. Концептуальные, теоретические и практические аспекты [Текст]. Учебное пособие для вузов / В.А. Ворона, В.А. Тихонов. 2-е изд., перераб. и доп. – М.: Горячая линия – Телеком, 2013. – 296 с.
9. Кузнецов, И.Н. Научное исследование. Методика проведения и оформление [Текст]/ И.Н. Кузнецов. – 2-е изд., перераб. и доп. – М.: Издательско-торговая корпорация «Дашков и К0», 2006. – 460 с.
10. Минто, Б. Принцип пирамиды Минто. Золотые правила мышления, делового письма и устных выступлений [Текст]. – М.: Манн, Иванов, Фербер, 2012. – 272 с.
11. Научные работы [Текст]: Методика подготовки и оформления / Сост. И.Н. Кузнецов. – 2-е изд., перераб. и доп. – Мн.: Амалфея, 2000. – 544 с.
12. Савина, И.А. Методика библиографического описания [Текст]/ практическое пособие / И.А. Савина. – М.: Либерея-Бибинформ, 2007. – 144 с.
13. Сукиасян, Э.Р. Список литературы к курсовой и дипломной работе. Рекомендации по составлению [Текст] / Э.Р. Сукиасян. – Москва, 2001.

ПРИЛОЖЕНИЕ А

Образец курсовой работы по дисциплине «Программирование»

ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМ. Т.Г. ШЕВЧЕНКО

Рыбницкий филиал

Кафедра физики, математики и информатики

КУРСОВАЯ РАБОТА

по дисциплине

«ПРОГРАММИРОВАНИЕ»

на тему:

«РЕАЛИЗАЦИЯ ИГРЫ “ТЕТРИС”»

Студента I курса
направления «Педагогическое
образование»
профиля «Информатика»
дополнительного профиля
«Иностранный язык»
Иванова Ивана Ивановича

Научный руководитель:
преподаватель
Чернега Наталья Владимировна

Рыбница, 2014

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ИССЛЕДОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	4
1.1. История создания игры	4
1.2. Правила игры	6
1.3. Обзор существующих программ	6
2. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА, ОПИСАНИЕ РЕЗУЛЬТАТОВ	9
2.1. Требования к программному продукту	9
2.2. Меню	9
2.3. Описание пользовательских подпрограмм	10
2.3. Руководство для пользователя	13
2.4. Анализ полученных результатов	15
ЗАКЛЮЧЕНИЕ	16
СПИСОК ЛИТЕРАТУРЫ	17
ПРИЛОЖЕНИЯ	

ВВЕДЕНИЕ

Игры – неперенная составляющая любой человеческой культуры. Слово «игра» прилагается почти ко всякому детскому занятию. Посредством игры наиболее интенсивно развивается целостное восприятие окружающего мира, наглядно-образное мышление, творческое воображение. Поэтому кроме программ, которые облегчают людям работу, программисты придумали увлекательное развлечение – компьютерные игры. Сначала в эти игры играли только они сами, но очень скоро их эстафету приняли дети и взрослые по всему миру. Сейчас уже трудно найти человека, имеющего компьютер и ни разу не игравшего в компьютерные игры.

Актуальность исследования определяется тем, что создание игр – это одно из самых популярных и интересных направлений в программировании. Оно позволяет изучить графические возможности компьютера, глубже понять способы его работы.

Объект курсовой работы – создание программного продукта.

Предмет курсовой работы – программный продукт, реализующий игру «Тетрис».

Цель курсовой работы – создание программного продукта, реализующего игру «Тетрис».

Для достижения этой цели были поставлены следующие *задачи*:

- изучить теоретические сведения, необходимые для решения данной задачи;
- систематизировать и обобщить полученные знания;
- изучить аналогичные программные продукты;
- создать пользовательский интерфейс;
- создать программный продукт, реализующий игру «Тетрис»;
- провести тестирование и отладку программы.

Цель и задачи курсовой работы определили её структуру. Она состоит из введения, двух разделов, заключения, списка литературы и приложений.

В первом разделе рассмотрена история игры «Тетрис», проведен обзор существующих программных продуктов, изучены правила игры.

Второй раздел посвящен разработке программного продукта, в нём описывается структура программы, пользовательские подпрограммы и подведены тестирование программы и анализ полученных результатов.

Заключение содержит в себе результаты проделанной работы при реализации игры «Тетрис».

В приложениях представлены блок-схема, описывающая структуру программы, и листинг разработанного программного продукта.

1. ИССЛЕДОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. История создания игры

«Тетрис» представляет собой простую и известную игру. Задача состоит в том, чтобы из разных фигур сложить одну строку, которая удаляется. За каждое удаление строки игрок получает одно очко. Если он сможет набрать достаточно очков, то сможет перейти на следующий уровень. С увеличением уровня увеличивается скорость падения фигур.

Все гениальное – просто. Редко, очень редко такая простота становится мировым достоянием. Именно так случилось с «Тетрисом» – программой, изменившей жизнь одного конкретного человека, оказавшей влияние на историю игровой индустрии и подарившей счастье миллионам людей. Ее создателем является Алексей Леонидович Пажитнов.

На самом деле графическая и логическая идея «Тетриса» была придумана не Алексеем. В конце 70-х – начале 80-х по всему миру (и в СССР в том числе) успешно продавались головоломки американского математика Соломона Голомба, самая известная из которых называлась *Pentomino Puzzle*. Их идея была довольно проста и до боли знакома любому современнику – из нескольких фигур нужно было собирать заданные фигуры. Судьбоносное знакомство Алексея с этой забавой состоялось в начале 80-х, когда такие головоломки были ещё в диковинку.

После этого русский инженер задумался о создании компьютерного варианта игры. Пажитнов дополнил идею: собирать «фигурки в стакане» предстояло в реальном времени, причём фигурки состояли из пяти элементов (Пентамино – от греч. «Пента» – пять), и по задумке должны были во время падения проворачиваться вокруг собственного центра тяжести. Электронные возможности машин того времени оказались слишком слабы для воплощения человеческих идей – для компьютерного «Пентамино» попросту не хватало ресурсов. Тогда Пажитнов принимает решение сократить количество блоков, из которых состояли падающие фигурки до четырёх, воплотив, таким образом «в цифре» не такое ресурсоёмкое «Тетрамино» (от греч. «Тетра» – четыре). Сокращённое название «Тетрис» прижилось чуть позже.

Нехитрая, в общем, идея дала впоследствии удивительные результаты. Первый вариант игры Алексей написал быстро, взяв за основу семь фигурок, ставших впоследствии стандартным набором «Тетриса», за две недели он запрограммировал основу игры. В ней в «стакан» падали даже не графические изображения фигур, а их текстовые аналоги, состоящие из пробелов, скобок и т.п. Выглядело это все примитивно – игра создавалась на базе языка *Pascal* и на компьютере «Электроника-60», у которого даже монитора нормального не было. Но зато игра работала.

Первое время за пределы той же лаборатории «Тетрис» не выходил. Пажитнов, апробировав начальный вариант на себе и коллегам, привлёк к делу соратника, шестнадцатилетнего Вадима Герасимова, и в течении двух месяцев была создана первая графическая, цветная версия «Тетриса», обладавшая управлением и таблицей рекордов. Важным моментом стал тот факт, что Герасимов фактически помог переделать игру на платформу PC, набирающую популярность с каждым днём. Вместе с PC популярность набирали и первые 5.25 дисководы, поэтому вопрос первоначального «тиражирования» решился довольно быстро – скопировав получившуюся забаву на дискеты, Алексей сотоварищи распространили её по кулуарам НИИ. Друг переписывал у друга, затем давал копировать ещё троим знакомым, и потихоньку электронное «письмо счастья» расползлось по столице, и выскользнуло за её пределы. Игра становилась известной в СССР.

PC-версия «Тетриса» просочилась за границу, и уже в Венгрии её адаптировали под нужды альтернативных платформ – *Commodore C64* и *Apple 2*. В тот момент в Институте гостил Роберт Штайн, владелец собственной игровой студии *Andromeda*. Опытный владелец моментально почувствовал потенциал игрушки. Штайн предлагает выкупить права сразу на все версии игры: на *Apple 2* и *C64* у венгров, а PC-вариант, соответственно, у русских.

Штайн в это время возвращается в Лондон и приступает к поискам рынков сбыта игры. Результатом этих поисков стал контракт между *Mirrorsoft* и *Andromeda*, который исчислялся смехотворной суммой в 3000 фунтов стерлингов и плавающим процентом от доходов (7-15% в зависимости от количества проданных копий).

А тем временем американцы из *Spectrum Holobyte*, у которых была версия игры от англичан из *Mirrorsoft*, на которую они не имели прав, всерьёз взялись за программу. Американцы переделывают оформление игры, оставляя механику без изменений. Чёрный фон меняют на заставки, в качестве мелодии звучит «Калинка-малинка» и «Э-эх, ухнем!», появляется заставка, в которой перед собором Василия Блаженного приземляется Матиас Руст на своём знаменитом самолётике *Cessna*.

«Тетрис» сразу получил известность и начал продаваться очень хорошими тиражами, постепенно завоёвывая всё большую и большую популярность. Пройдёт менее года и «Тетрису» присудят несколько важнейших наград Американской Ассоциации Разработчиков Программного Обеспечения. В частности, как лучшему потребительскому софту, лучшей оригинальной разработке и лучшей развлекательной программе.

А в это время в Японии выходит PC-версия «Тетриса», а также его *Famicom*-вариант на консоли от *Nintendo*, которая в итоге расходуется

по стране Восходящего Солнца более чем двухмиллионным тиражом. Для того, чтобы объяснить, как такое стало возможным, необходимо изучить историю намного глубже.

Хэнк Роджерс – одна из ключевых фигур во всей этой истории. Владелец небольшой японской фирмы *Bullet Proof Software* стал тем человеком, которому *Spectrum Holobyte* продала права публикации «Тетриса» на рынке Японии. Роджерс привозит в Москву Аракаву и его юрисконсульта Говарда Линкольна. 21 марта 1989 г. крупнейшему игроку на рынке электронных развлечений *Nintendo* отошли все права на «Тетрис» (для видеорынка) за сумму, которую независимые аналитики оценивают в пределах 3-5 млн долларов.

«Тетрис» стал суперхитом всех времён и народов. Благодаря ему, за первые несколько лет существования *GameBoy* разошелся более чем 30-миллионным тиражом, а самих картриджей с программой было продано около 15 млн. «Тетрис» же принёс компании *Nintendo* по разным оценкам от 2 до 3 миллиардов долларов чистой прибыли, учитывая все порты, версии и лицензионные отчисления.

1.2. Правила игры

Случайные фигуры тетрамино падают сверху в прямоугольный стакан шириной 25 и высотой 30 клеток. В полёте игрок может поворачивать фигуру и двигать её по горизонтали. Также можно «сбрасывать» фигуру, то есть ускорять её падение, когда уже решено, куда фигура должна упасть. Фигура летит, пока не наткнётся на другую фигуру либо на дно стакана. Если при этом заполнился горизонтальный ряд из 25 клеток, он пропадает и всё, что выше его, опускается на 1 клетку. Темп игры постепенно увеличивается. Название игры происходит от количества клеток, из которых состоит каждая фигура. Игра заканчивается, когда новая фигура не может поместиться в стакан. Игрок получает очки за каждую фигуру, поэтому его задача – заполнять ряды, не заполняя сам стакан как можно дольше, чтобы таким образом получить как можно больше очков.

1.3. Обзор существующих программ

Игра реализована практически на всех современных компьютерах, включая мобильные телефоны, игровые видеоприставки, телевизоры (как дополнительная функция), множество карманных игровых устройств. Есть варианты игры для всех сколько-нибудь распространённых операционных систем, а также для *Java*. Есть порт

даже для осциллографа. Трудно, если вообще возможно, назвать такую вычислительную платформу, где бы не было этой игры.

Пожалуй, наибольшую популярность приобрела реализация «Тетриса» для игровой консоли *Game Boy* и видеоприставки *NES*.

Во многих реализациях стакан изначально не пуст. Есть реализации (например, бесплатная *Gravytris* для *Microsoft Windows*) с более реалистичными правилами гравитации: например, при пропадании горизонтального ряда блоки, которые выше его, соединяются в связные области и каждая область падает, пока не наткнётся на блок; это может привести к заполнению новых рядов и новым падениям, и так далее.

Были написаны трёх- (*Blockout* фирмы *California Games*, 1989 год), четырёх- (*HyperTetris*, 1996 год) и даже n-мерный (*Polytope Tetris*, 2003 год) варианты тетриса, а также модификации для двух и более игроков.

Существуют также версии игры, в которых игра ведётся не на очки, а на открытие спрятанного изображения. Для облегчения игры есть варианты без сложных S- и Z-образных фигур и без увеличения скорости игры.

Кроме того, очень популярной разновидностью «Тетриса» является *TetColor*, написанный в 1991 году Сергеем Сотниковым (Тула), где исчезают линии (горизонтальные, вертикальные и диагональные), собранные из одного цвета. Аналогична ей *Acid Drop*, разработанная программистом Дэнисом М. Киссом.

Существует и более сложная версия игры, основанная на пентамино, с фигурками из пяти элементов, – *Pentix* («Пентикс»), в которой количество возможных фигур увеличено до 12-18.

Также есть реализация игры, где элементами фигур являются треугольники – *Crazy Tetris om Astatix Software*, *Amazing Tetris*.

Один из новейших примеров – игра *Dwice*, разработанная в 2006 году самим изобретателем «Тетриса», Алексеем Пажитновым.

Также «Тетрис» вложен в качестве пасхального яйца в текстовый редактор *emacs* (открывается после нажатия Esc + X и ввода команды *tetris*).

Кроме того, «Тетрис» встроен в клиент *µTorrent*. Для этого необходимо открыть «Справка» – «О программе» и нажать клавишу *T*.

PhysTris – тетрис с реалистичным физическим поведением блоков фигур. В игре можно поменять режим игры, количество кубиков фигур (4,5 и более), музыку и фон. Игру разработал *Peter Petrov* в 2009 году.

После изучения теоретического материала было решено создать программный продукт, реализующий игру «Тетрис» в среде программирования *Turbo Pascal*, который будет соответствовать

общепринятым стандартам, установленным различными версиями игры «Тетрис» и обладать следующими функциями:

- меню с такими пунктами, как «играть», «опции» и «выход»;
- страницу рекордов;
- сохранение и открытие игры;
- наличие уровней сложности с возрастающей скоростью;
- стандартный набор фигур.

Основное назначение программы – развлечение играющих, совершенствование их координации и логического мышления.

2. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА, ОПИСАНИЕ РЕЗУЛЬТАТОВ

2.1. Требования к программному продукту

При разработке программного продукта требовалось решить следующие задачи:

1. Формирование игрового экрана.
2. Формирование, движение и остановка фигур.
3. Сохранение и загрузка игры, пауза.
4. Переназначение управления.
5. Вывод во время игры на экран различных параметров:
 - времени игры,
 - следующую фигуру,
 - количество очков и жизнь.
6. Создание списка рекордов и его просмотр, а также вывод из файла информации об игре и лицензии.

2.2. Меню

«Тетрис» – это игра. Поэтому при разработке интерфейса пользователя был учтен ряд особенностей:

1. Интерфейс должен быть интуитивно понятен простому пользователю компьютера.
2. Цветовая схема не должна раздражать человека.

Программа предназначена для приятного проведения свободного времени. Она не должна иметь сильно сложную систему управления. Поэтому была использована всем известная система управления программой – меню. Каждому пункту меню присвоено название, соответствующее функциональному назначению. Это облегчает работу пользователя.

Что же касается второго пункта, здесь применены различные цвета. В сочетании они не несут никакой психической нагрузки, поэтому время препровождение для человека будет приятным.

В программе был использован русифицированный шрифт TRIP.CHR.

Для вставки рисунка из bmp-файла была использована функция:

Function Draw(x0,y0:integer; fname:string; transparent:boolean):integer,
где *x0*, *y0* – левые верхние координаты рисунка; *Fname* – имя bmp-файла; *Transparent* – включать отсечения или нет.

Для управления меню мышью были использованы процедуры и функции стандартного модуля *Turbo Pascal 7.1* – «*Mouse_ru*»:

Mousein(x1,y1,x2,y2) – функция, возвращающая *True*, если мышь находится в заданном прямоугольнике, где *x1,y1,x2,y2* – координаты левого верхнего угла и правого нижнего углов.

Mousepressed – функции, возвращающая *True*, если нажата клавиша мыши.

Initmouse – функция, возвращающая *True*, если мышь доступна.

Showmouse – показывает указатель мыши.

Hidemouse – прячет указатель мыши.

Для работы с клавиатурой были использованы процедуры и функции стандартного модуля *Turbo Pascal 7.1* – «*Keyboard*»:

Inkey – функция, считывающая коды нажатых клавиш с буфера клавиатуры. Отличается от функции *Readkey* тем, что позволяет считывать коды функциональных и управляющих клавиш.

Суть работы меню заключается в том, что на экран последовательно в цикле выводятся текст меню и, если указатель мыши перемещается в прямоугольник, в котором находится меню, то текст меню прорисовывается, синим цветом, иначе оно рисуется зелёным цветом.

На этом принципе основана процедура *MenuShow*, описанная ниже.

Procedure MenuShow(x1,y1,x2,y2:integer; text:string; var mousepresed:boolean); где *x1,y1,x2,y2* – координаты верхнего левого и правого нижнего угла меню; *Text* – текст меню; *Mousepresed* – признак нажатия меню.

2.3. Описание пользовательских подпрограмм

При разработке продукта был составлен алгоритм, отображающий взаимосвязь основных блоков программы (Приложение 1).

Программный продукт состоит из пользовательских подпрограмм, листинг которых представлен в Приложении 2.

Для формирования игрового экрана был взят за основу следующий принцип: игровой экран состоит из квадратов, одни из которых светятся, а другие нет. Для этой цели удобнее всего использовать матрицу, в которой светящиеся квадраты обозначаются 1, чёрные – 0. Для обозначения фигур, которые остановились, используется 2.

Для вывода игровой матрицы на экран была написана процедура *Ikran*:

Procedure Ikran(a:maska), где *a* – игровая матрица.

Для подготовки экрана к игре была создана процедура *Podgotovka*.

Для обнуления игровой матрицы была использована процедура *obnulenie*:

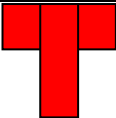
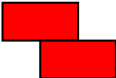
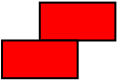
Procedure obnulenie (*k:integer; var a:maska*), где *k* показывает полное это обнуление или нет.

Для формирования фигур (табл. 2.1) были использованы константы матрицы, значение которых присваивалось переменной – фигуре. Для этих целей использована процедура *figureGreat*:

Procedure figureGreat (*n:byte; var b:figura*), где *N* – номер фигуры, *B* – созданная фигура.

Таблица 2.1

Элементы, используемые в игре

Название фигуры	Фигура	Массив, соответствующий фигуре
Q		((0,0,0,0), (0,1,1,0), (0,1,1,0), (0,0,0,0))
I		((0,0,0,0), (1,1,1,1), (0,0,0,0), (0,0,0,0))
R		((0,0,0,0), (0,1,1,0), (0,1,0,0), (0,0,0,0))
T		((0,0,0,0), (1,1,1,0), (0,1,0,0), (0,0,0,0))
Z		((0,0,0,0), (1,1,0,0), (0,1,1,0), (0,0,0,0))
S		((0,0,0,0), (0,1,1,0), (1,1,0,0), (0,0,0,0))
L		((0,1,0,0), (0,1,0,0), (0,1,1,0), (0,0,0,0))
J		((0,0,1,0), (0,0,1,0), (0,1,1,0), (0,0,0,0))

Для переворота фигур была использована процедура *perevorot*, которая транспонирует матрицу-фигуру:

Procedure perevorot (var b:figura).

Движения фигуры осуществляется при помощи функции *figure*:

Function figure (b:figura; var a:maska; var i0,j0:byte):Boolean, где *b* – фигура; *a* – игровой экран; *i0,j0* – новые координаты фигуры.

Данная функция также следит за тем, чтобы фигура не входила в другую фигуру, не выходила за рамки экрана. Если движущаяся фигура остановилась, то *figure* принимает значение *True*.

Остановка фигуры осуществляется при помощи процедуры *figurestop*:

Procedure figurestop(i0,j0:byte;b:figura; var a:maska), где *i0,j0* – координаты фигуры, *b* – фигура, *a* – игровой экран.

Работает эта процедура так: матрице игрового экрана *a* присваивается значение фигуры *B* в точке (*i0,j0*). После обнуления игрового экрана фигура прорисовывается на том же месте.

Удаление заполненной линии и подсчет очков осуществляется при помощи процедуры *DelLine*:

Procedure DelLine(var a:maska; var score:byte; var level:byte), где *a* – игровой экран, *score* – итоговый счет, *level* – итоговый уровень.

Для сохранения игры была написана процедура *save*:

Procedure save (a:maska; b:figura; i,j:integer; score,level:byte; h,min,sec:word), которая создает файл с именем *save.txt* и сохраняет в него параметры игры: *a* – игровой экран, *b* – фигура, *i, j* – координаты фигуры, *score* – количество очков, *level* – уровень, *h, min, sec* – время игры.

Для загрузки игры была разработана процедура *load*:

Procedure load(Nkey:key), где *Nkey* – управляющие клавиши, которая считывает игровые параметры и вызывает со считанными параметрами игру.

Прорисовывает игровой экран в текущий момент времени процедура *tetris*:

Procedure tetris (Nkey:key; a1:maska; b1:figura; i3,j3:integer; score1, level1:byte; h1,min1,sec1:word); где *Nkey* – управляющие клавиши; *a1* – массив игрового экрана; *B1* – фигура; *i3,j3* – координаты фигуры; *Scored1* – количество очков; *Level1* – текущий уровень; *h1, min1, sec1* – время игры;

Пауза осуществляется при помощи процедуры *pauza*:

Procedure pauza(var t1:time), где *t1* – время паузы.

Процедура показа и переназначения клавиш:

Procedure Knopki (var newkey:key), где *Newkey* – новые клавиши управления. Данная подпрограмма использует процедуру *Pokaz*, которая возвращает название клавиши по её коду.

Вывод на экран информации осуществляется при помощи процедуры *interface*, которая показывает количество очков, жизнь, следующую фигуру, время игры:

Procedure interface (score,level:byte; b:figura; T1:time; var s3,s2,s1:string; var h,min,sec:word), где *Score* – количество очков; *Level* – уровень; *B* – следующая фигура; *T1* – время начала игры; *s3,s2,s1* – время, жизнь, очки; *H,min,sec* – время игры.

Если во время игры игрок дойдет до 9-го уровня, то имя игрока и время игры добавятся к таблице рекордов и сохраняются в файле *recordi.txt* при помощи локальной процедуры *recordi*:

Procedure recordi(h,m,sec:word), где *h,m,sec* – время игры.

Для показа таблицы рекордов использована глобальная процедура *Recordi*, которая считывает рекорды из файла *recordi.txt*.

Вывод информации об игре и о лицензии осуществляется при помощи процедуры *procedure Info(a:word)*, которая выводит на экран, в зависимости от *a*, содержимое файлов *INFO1.txt* (меню «Об игре») или *INFO2.txt* (меню «Лицензия»). При *a=1* – *INFO1.txt*, при *a=2* – *INFO2.txt*.

2.3. Руководство для пользователя

После запуска программного продукта появляется меню, состоящее из трёх пунктов «Играть», «Опции», «Выход». Вид меню показан на рис. 2.1.

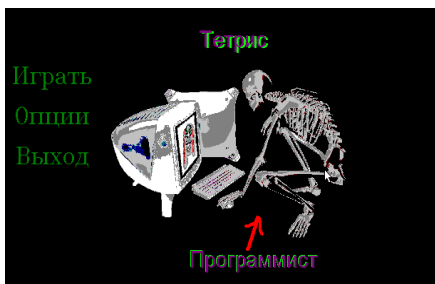


Рис. 2.1. Меню игры «Тетрис»

При выборе пункта меню «Играть», начнется новая игра. Окно игры состоит из трех областей: слева показана информация об уровне и количестве заработанных очков, по центру располагается игровое поле, справа – время игры и следующая фигура. Вид игрового окна показан на рис. 2.2.

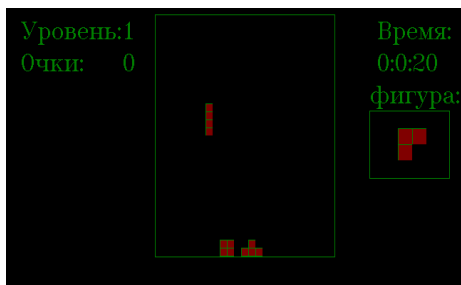


Рис. 2.2. Начало новой игры

Для управления падающими на поле фигурами используются выделенные для этого клавиши (табл. 2.2).

Таблица 2.2

Стандартное управление	
Клавиша	Действие
←	Движение влево
→	Движение вправо
↓	Движение вниз
Пробел	Переворот фигуры
P	Пауза
F5	Быстрое сохранение

При нажатии кнопки *ESC* или выборе пункта меню «Выход» будет осуществлен выход из программы.

При выборе пункта меню «Опции» на экран выводится меню (рис. 2.3), включающее такие пункты, как «Загрузить», «Рекорды», «Кнопки», «0 игр», «Лицензия».

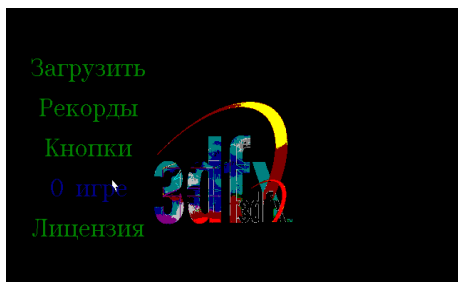


Рис. 2.3. Меню, вызываемое командой «Опции»

Пункт меню «Загрузить» позволяет загрузить из файла сохраненную ранее игру.

Пункт меню «Рекорды» выводит на экран список рекордов. Для выхода необходимо нажать клавишу *ESC*. Если игрок дойдет до 9-го уровня, то его имя и время запишутся в таблицу рекордов.

Меню «Кнопки» позволяет менять управление. Для смены управляющей клавиши необходимо щелкнуть мышью на нужную кнопку и нажать на новую клавишу управления. Новое управление сразу сохраняется. Для выхода необходимо нажать *ESC*.

Меню «Об игре» выводит на экран краткую информацию о программном продукте.

Меню «Лицензия» показывает сведения о лицензии данного программного продукта.

2.4. Анализ полученных результатов

В ходе тестирования созданного программного продукта было выявлено, что поставленная цель – создание программного продукта, реализующего игру «Тетрис» – и все задачи курсовой работы были достигнуты.

Также при выполнении курсовой работы были реализованы дополнительные функции:

- графическое меню с использованием растрового изображения и манипулятора – мышь;
- возможность сохранения и загрузки сохраненной игры;
- запись рекордов.

В результате тестирования программного продукта студентами не было получено жалоб об отклонениях в правилах игры по сравнению с общепринятыми стандартами, установленными многочисленными версиями игры «Тетрис». Это позволяет сделать вывод о соответствии реализованного алгоритма.

Особенностью игры является наглядность информации: на экране постоянно отображена игровая ситуация. Текст программы удобочитаем, т.к. содержит комментарии, поясняющие суть той или иной ситуации.

ЗАКЛЮЧЕНИЕ

В процессе написания курсовой работы на тему «Создание программного продукта, реализующего игру «Тетрис» поставленная цель была достигнута.

При достижении поставленной цели были решены следующие задачи:

- изучены аналогичные программные продукты;
- изучены теоретические сведения, необходимые для решения данной задачи;
- систематизированы и обобщены полученные знания;
- создано пользовательское меню;
- создан программный продукт, реализующий игру «Тетрис»;
- проведены тестирование и отладка программы;

В ходе выполнения курсовой работы были изучены новые области, такие как: работа с графическим драйвером в среде DOS; работа с манипулятором – мышью в среде DOS; работа с клавиатурой в неблокируемом режиме; создание модулей.

В ходе тестирования созданного программного продукта были выявлены следующие недостатки:

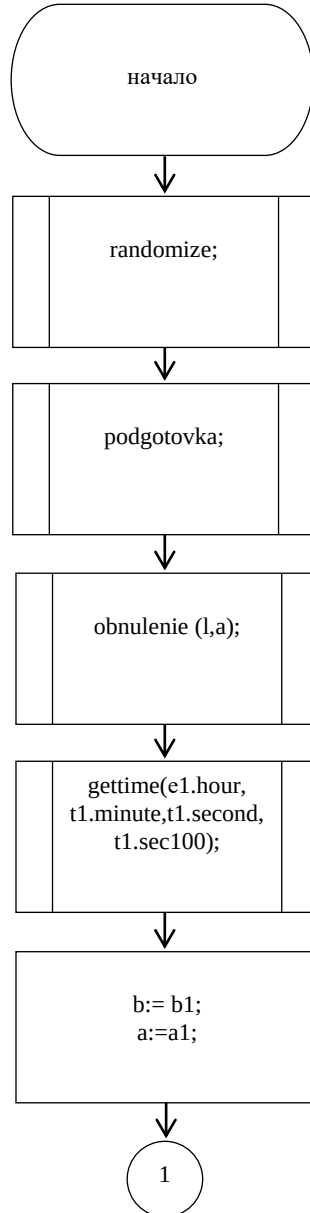
1. При длительном нажатии кнопки, перемещающей фигуру вправо или влево, движение может продолжаться еще некоторое время, даже если отпустить клавишу.
2. В игре нет звукового сопровождения.
3. В программе нет многопользовательского интерфейса и возможности сохранять и загружать несколько игр.
4. В списке рекордов имена победителей не сортируются по времени, а записываются последовательно.
5. Измененное управление при новом запуске игры меняется на стандартное.
6. Скорость движения фигур меняется на разных компьютерах, так как она установлена процедурой *Delay*.

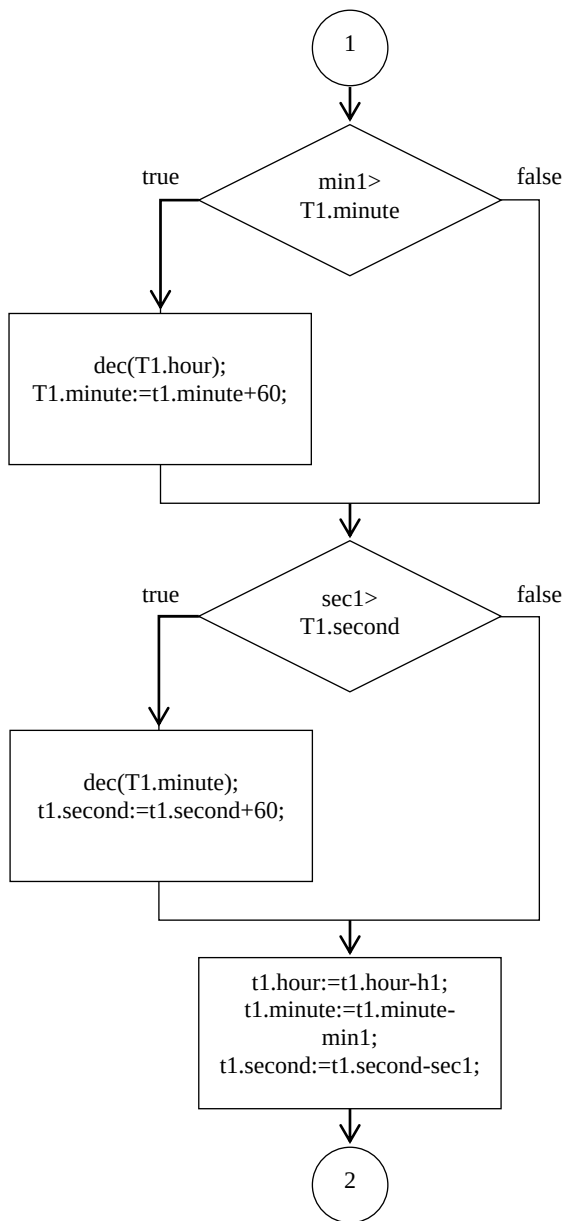
Подводя итоги, можно сказать, что при помощи разработанного программного продукта можно приятно провести время, совершенствовать координацию и развивать логическое мышление. Программа не занимает много места, не требовательна к установленному программному обеспечению.

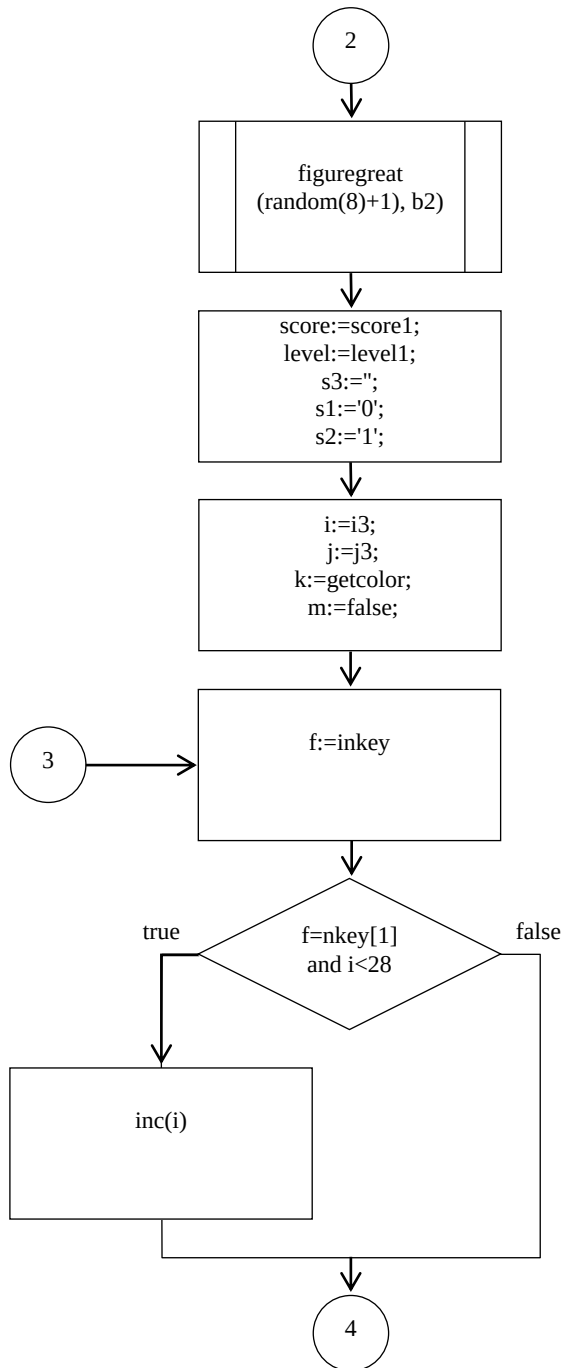
СПИСОК ЛИТЕРАТУРЫ

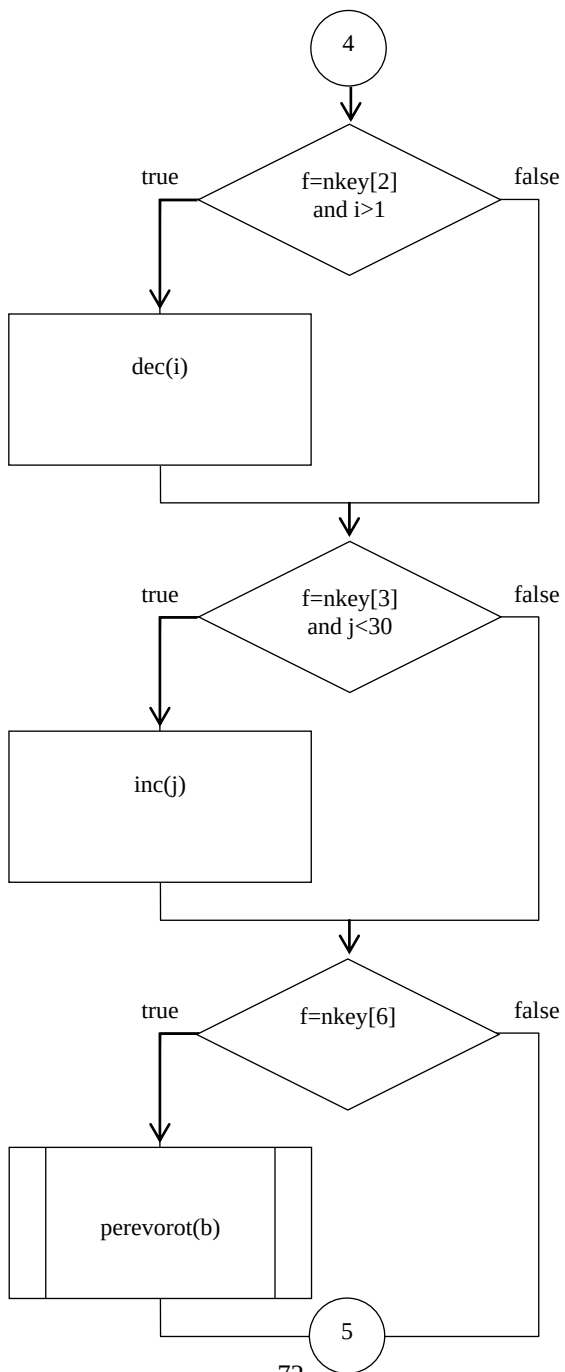
1. Глинский, Я.Н., Анохин, В.Е., Ряжская, В.А. Паскаль. *Turbo Pascal и Delphi* [Текст]. – 3-е изд. – Львов: Деол, 2003. – 208 с.
2. Иванова, Г.С. Основы программирования [Текст]: Учебник для вузов. – 2-е изд., перераб. и доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 416 с.
3. Культин, Н.Б. *Turbo Pascal* в задачах и примерах [Текст]. – СПб.: БХВ-Петербург, 2006. – 256 с.
4. Культин, Н.Б. Программирование в *Turbo Pascal 7.0* и *Delphi* [Текст]. – 3-е издание. – СПб.: БХВ-Петербург, 2012.
5. Меженный, О.А. *Turbo Pascal*: учитесь программировать [Текст]. – М.: Издательский дом «Вильямс», 2001. – 448 с.
6. Меженный О.А. Самоучитель *Turbo Pascal* [Текст]. – М.: Диалектика, 2008. – 330 с.
7. Немнюгин, С.А. *Turbo Pascal* [Текст]: практикум. – СПб: Питер, 2001. – 256 с.
8. Павловская, Т.А. Паскаль. Программирование на языке высокого уровня [Текст]: Учебник для вузов. – СПб.: Питер, 2007. – 293 с.
9. Попов, В.Б. Паскаль и Дельфи [Текст]: самоучитель / В.Б Попов и др. – СПб.: Питер, 2004. – 543 с.
10. Программирование на языке Паскаль [Текст]: задачник / под ред. О.Ф. Усковой. – СПб.: Питер, 2002. – 336 с.
11. Фаронов В.В. *Turbo Pascal 7.0*. Практика программирования [Текст]. Учебное пособие. – М.: КноРус, 2012. – 414 с.
12. Шпак Ю.А. Программирование в *Turbo Pascal*. Переход к *Delphi* [Текст]. – Киев: МК-Пресс, 2006. – 416 с.

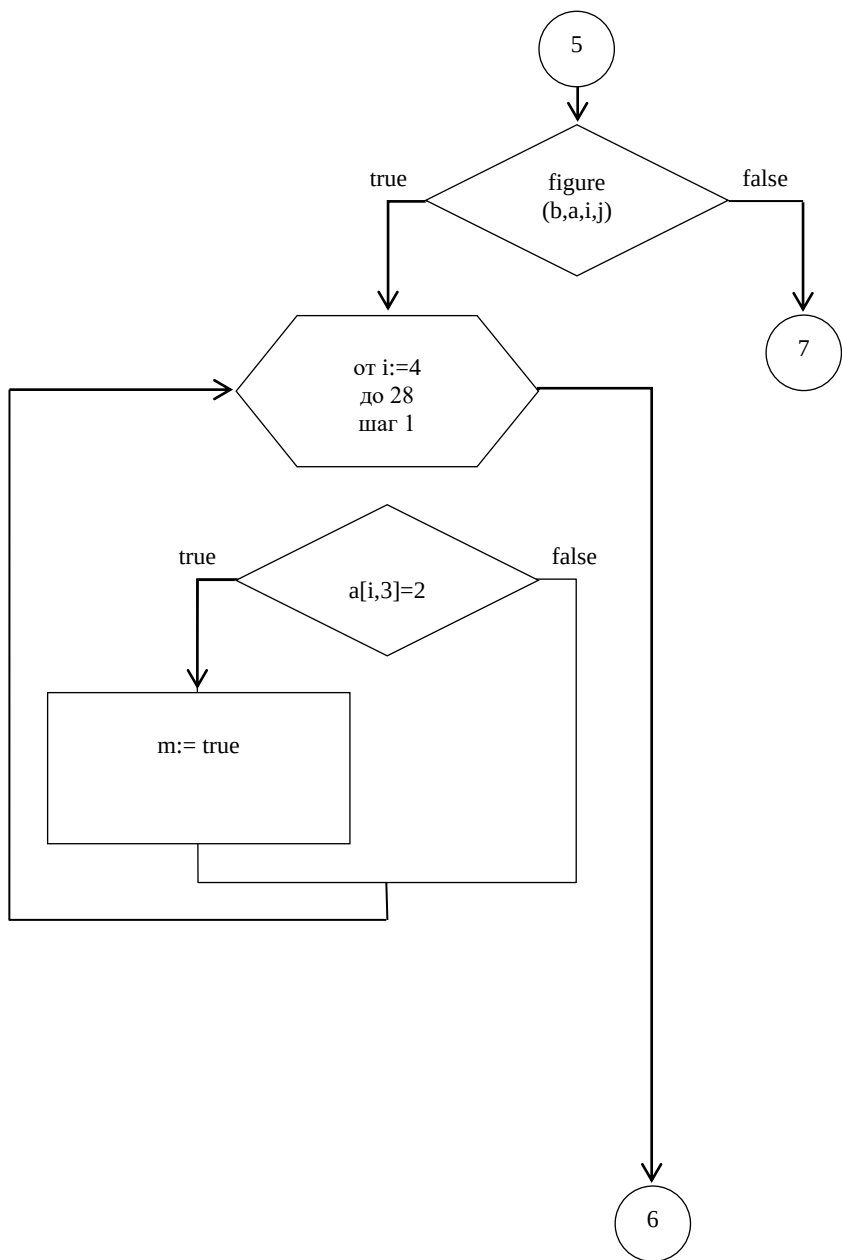
Алгоритм программного продукта

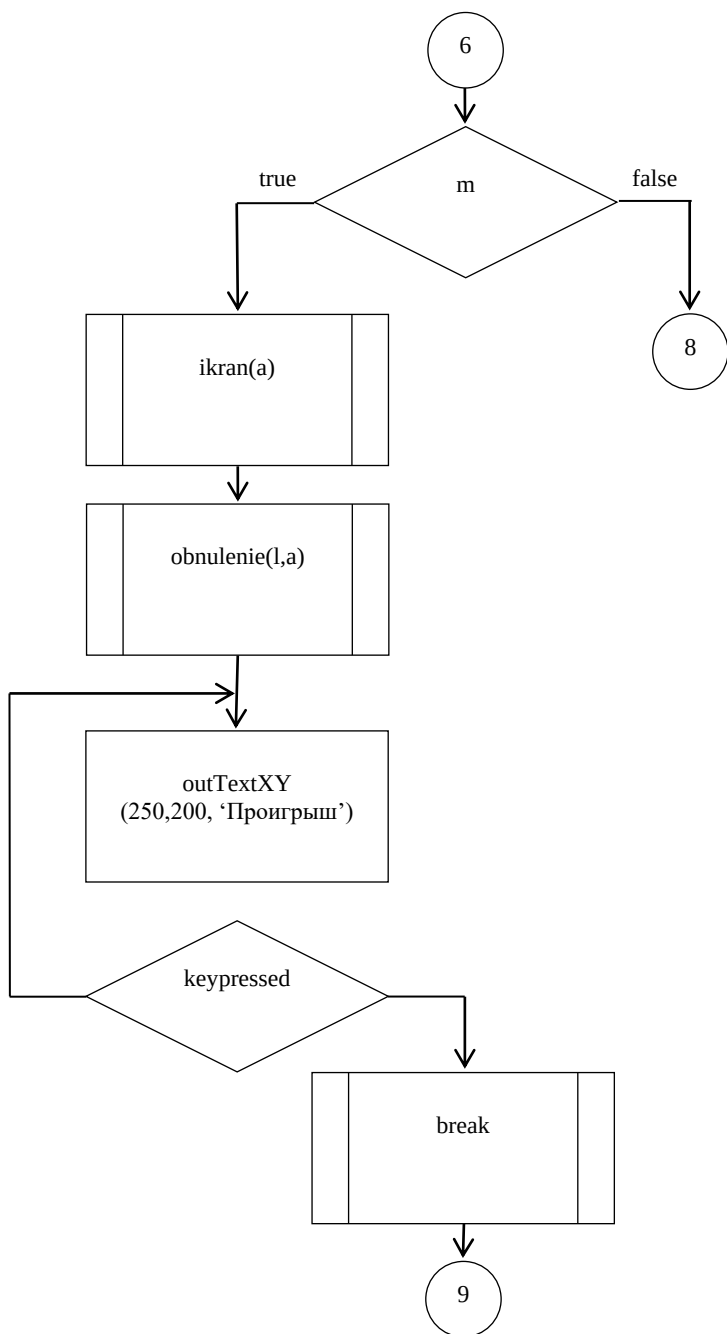


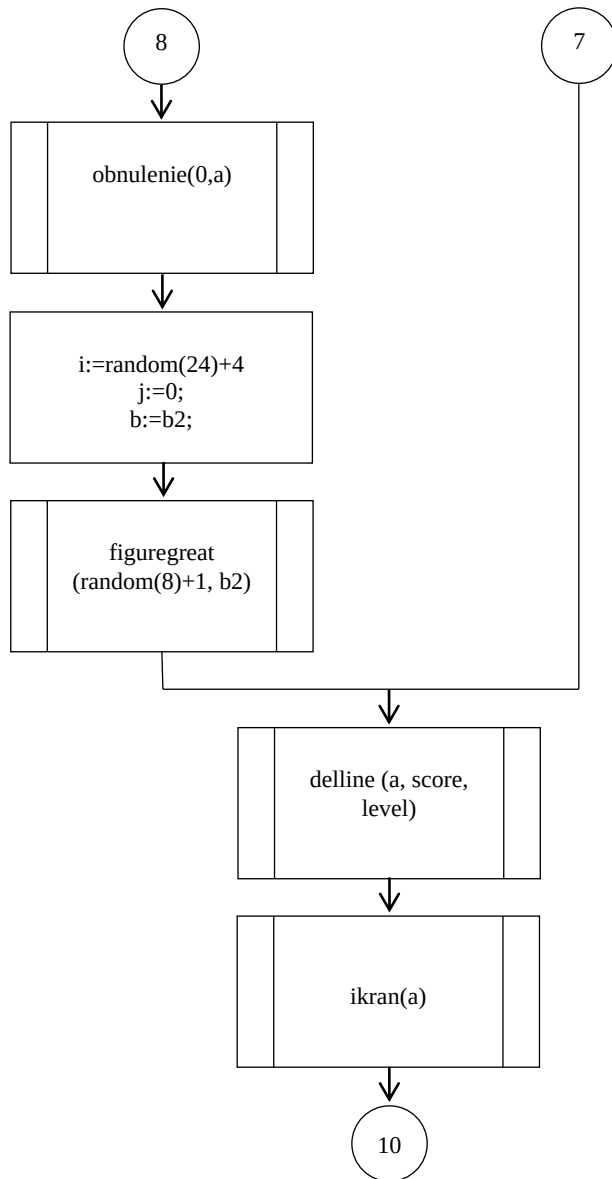


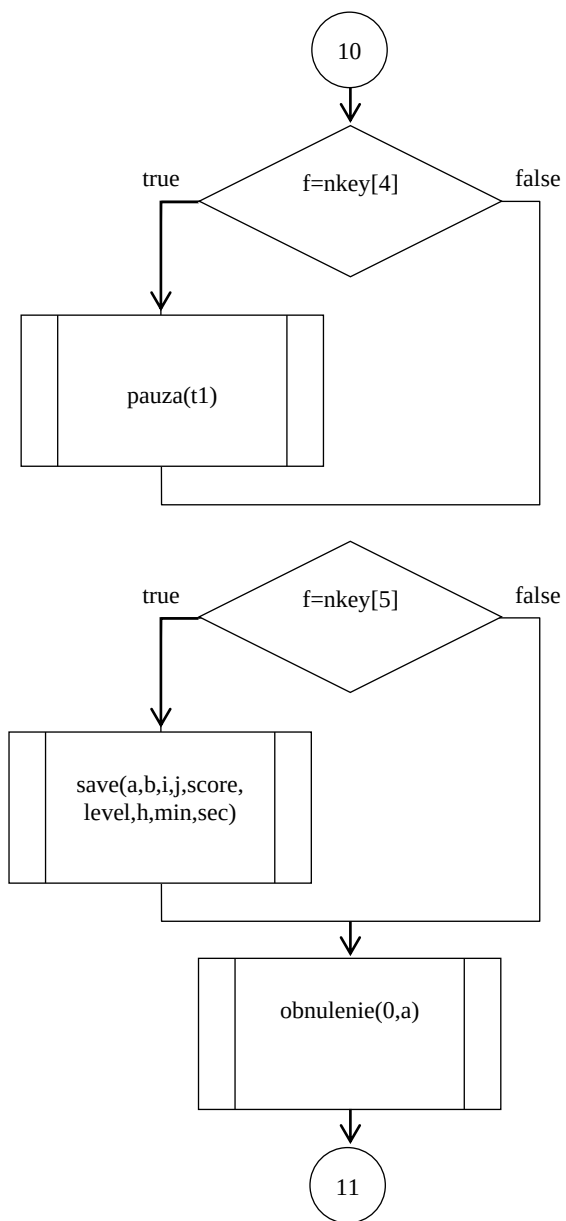


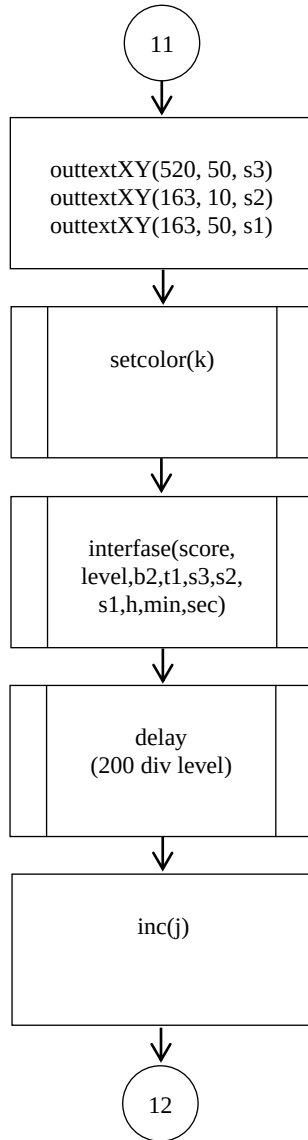


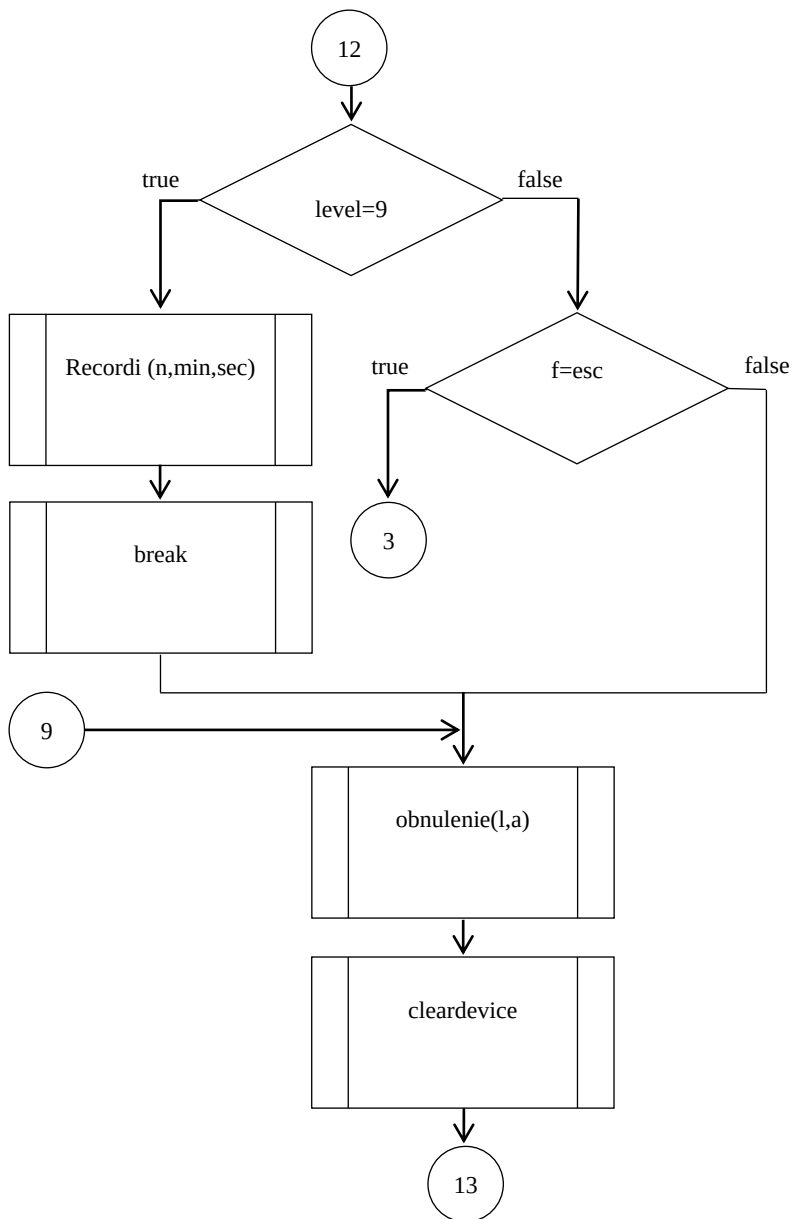


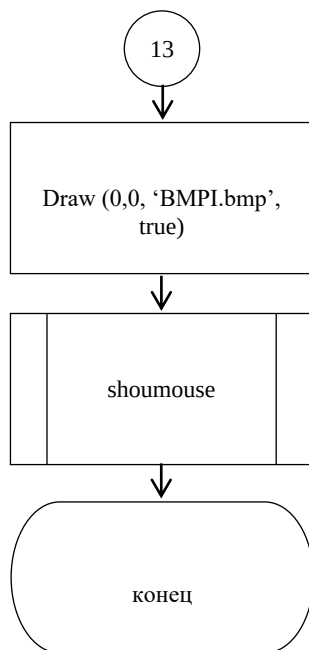












ПРИЛОЖЕНИЕ 2

Листинг программного продукта

Основная программа

```
PROGRAM Tetris;  
{Вызов модуля GAME}  
uses game;  
BEGIN  
    menu;  
END.
```

Модуль, содержащий меню

```
unit game;  
interface  
uses  
{Модуль для работы с мышью из Turbo Pascal 7.1}  
Mouse_ru,  
{Модуль для работы с клавиатурой из Turbo Pascal 7.1 : позволяет  
считывать коды функциональных клавиш и клавиш управления}  
keyboard,  
graph,  
{Модуль содержащий все подпрограммы игры}  
igra;  
var newkey:key;  
{Процедура, показывающая меню Опции}  
procedure menu;  
{Процедура, показывающая главное меню}  
procedure menu1;  
implementation  
  
procedure menu1;  
const  
text2:array[1..5]of string=('Загрузить','Рекорды','Кнопки', 'Об игре',' Лицензия');  
var  
    i,kнопка:integer;  
    a:array[1..5]of boolean;  
begin  
    hidemouse;{Процедура, прячущая указатель мыши}  
    cleardevice;  
    draw(50,0,'BMP1.bmp',true);{Выводит на экран рисунок BMP1.bmp}  
    showmouse;{Процедура, показывающая курсор}  
    repeat  
        кнопка:=inkey;{Читает код нажатой клавиши}  
        for i:=1 to 5 do  
            menushow(40,0+i*50,150,50,text2[i],a[i]);  
            {Процедура, показывающая строку меню}  
        if (a[1])and((Кнопка=Enter)or(mousepressed)) then
```



```

begin
    load(newkey); {Загружает игру}
    exit;
end;
if (a[2])and((Knopka=Enter)or(mousepressed)) then
    recordi;
if (a[3])and((Knopka=Enter)or(mousepressed)) then
    knopki(newkey);
if (a[4])and((Knopka=Enter)or(mousepressed)) then
    info(1);
if (a[5])and((Knopka=Enter)or(mousepressed)) then
    info(2);
until knopka=ESC;
hidemouse;
cleardevice;
draw(0,0,'BMPI.bmp',true);
showmouse;
end;

procedure menu;
const
text1:array[1..3]of string=('Играть','Опции','Выход');
var
    i,knopka:integer;
    a:array[1..3]of boolean;
    a1:maska;
    b:figura;
begin
    grafinit;
    setgraphmode(1);
    settextstyle(1,0,4);
    cleardevice;
    draw(0,0,'BMPI.bmp',false);
    initmouse;
    showmouse;
    obnulenie(1,a1); {Процедура, обнуляющая игровой экран}
    figuregreat(random(8)+1,b); {Создание фигуры}

    {Назначение клавиш управления}
    newkey[1]:=RIGHT;
    newkey[2]:=LEFT;
    newkey[3]:=DOWN;
    newkey[4]:=ord('p');
    newkey[5]:=F5;
    newkey[6]:=Space;
    repeat
        Knopka:=inkey;
        for i:=1 to 3 do

```

```

        menushow(20,10+i*50,150,50,text1[i],a[i]);
        if (a[1])and((Кнопка=Enter)or(mousepressed)) then
            tetris(newkey,a1,b,3,0,0,1,0,0,0);{Начало игры}
        If (a[2])and((Кнопка=Enter)or(mousepressed)) then
            menu1;{Меню Опции}
        until (Кнопка=ESC)or((a[3])and(mousepressed));
        closegraph;
    end;
end.

```

Модуль, содержащий все основные процедуры и функции

```

unit igra;
interface
uses graph,keyboard,mouse_ru,crt,dos;
type
    key=array[1..6]of integer;{Массив управляющих клавиш}
    figura=array[1..4,1..4]of byte;{Фигура}
    Maska=array[1..31,1..33]of byte;{Массив игрового экрана}
    time=record
        hour,minute,second,sec100:word;
    end;

const
    Pыt1='d:\tp1\bgi';{Путь до графического драйвера}
    Pыt2='D:\TP1\BIN\PRO\TETPИС2\';{ Путь до папки с игрой}
    fig=array[1..8]of figura=
        {Массивы фигур}
        (((0,0,0,0),
        (0,1,1,0),
        (0,1,1,0),
        (0,0,0,0)),

        ((0,0,0,0),
        (1,1,1,1),
        (0,0,0,0),
        (0,0,0,0)),

        ((0,0,0,0),
        (0,1,1,0),
        (0,1,0,0),
        (0,0,0,0)),

        ((0,0,0,0),
        (1,1,1,0),
        (0,1,0,0),
        (0,0,0,0)),

```

((0,0,0,0),
(1,1,0,0),
(0,1,1,0),
(0,0,0,0)),

((0,0,0,0),
(0,1,1,0),
(1,1,0,0),
(0,0,0,0)),

((0,1,0,0),
(0,1,0,0),
(0,1,1,0),
(0,0,0,0)),

((0,0,1,0),
(0,0,1,0),
(0,1,1,0),
(0,0,0,0)));

{Игра Тетрис
Nkey – управляющие клавиши;
a1 – массив игрового экрана;
B1 – фигура;
I3, J3 – координаты фигуры;
Scored1 – количество очков;
Level1 – текущий уровень;
h1, min1, sec1 – время игры;}

Procedure Tetris (Nkey:key; a1:maska; b1:figura; i3, j3:integer; score1,
level1:byte; h1, min1, sec1:word);

{Процедура загрузки игры
Nkey – управляющие клавиши;}
procedure Load(Nkey:key);

{Процедура выводющая меню на экран и возвращающая признак
нажатия меню
X1,Y1 и X2,Y2 – координаты верхнего левого верхнего и правого
нижнего угла меню;
Text – текст меню;
Mousepresed – признак нажатия меню;}
procedure MenuShow(x1,y1,x2,y2:integer;text:string;VAR
mousepresed:boolean);

{Процедура показа и переназначения клавиш
Newkey – новые клавиши управления}
procedure Knopki(var newkey:key);

```

    {Процедура, инициализирующая графический режим}
procedure grafinit;

    {Процедура, выводящая на экран, в зависимости от 'a', содержимое
    файлов INFO1.txt(меню 'Об игре') или INFO2.txt(меню 'Лицензия')
    При A=1 – INFO1.txt;
    При A=2 – INFO2.txt;}
procedure Info(a:word);

    {Процедура, обнуляющая матрицу игрового экрана
    A – Матрица игрового экрана;
    K(0 или 1)- обнуление полное или неполное; }
PROCEDURE obnulenie(k:integer;var a:maska);

    {Выводит из файла на экран список рекордов}
procedure recordi;

    {Процедура, создающая новую фигуру
    N – номер фигуры;
    B – созданная фигура}
procedure figureGreat(n:byte;var b:figura);

    {Процедура выводящая на экран рисунок из BMP файла
    X0, Y0 – Левые верхние координаты рисунка;
    Fname – имя bmp-Файла;
    Transparent – включать отсечения или нет}
function Draw(x0,y0:integer;fname:string;transparent:boolean):integer;

implementation
    {Процедура загрузки игры}
procedure load(Nkey:key);
var
    a1:maska;b1:figura;
    i3,j3,i,j,score,level:byte; k,h,min,sec:word;f:text;s:string;
begin
    assign(f,pyt2+'save.txt');
    {$I-}
    reset(f);
    {$I+}
    k:=IORResult;
    if k=0 then
        begin
            while not eof(f) do
                begin
                    for i:=1 to 31 do begin
                        for j:=1 to 33 do begin
                            readln(f,s);

```

```

                                val(s,a1[i,j],k);
                                end;end;
                                for i:=1 to 4 do begin
                                for j:=1 to 4 do begin
                                    readln(f,s);
                                    val(s,b1[i,j],k);
                                end;end;
                                readln(f,s);val(s,i3,k);
                                readln(f,s);val(s,j3,k);
                                readln(f,s);val(s,score,k);
                                readln(f,s);val(s,level,k);
                                readln(f,s);val(s,h,k);
                                readln(f,s);val(s,min,k);
                                readln(f,s);val(s,sec,k);
                                end;
                                close(f);
                                {Вызов игры со считанными параметрами}
                                tetris(nkey,a1,b1,i3,j3,score,level,h,min,sec)
                                end
                                else begin
                                    hidemouse;
                                    cleardevice;
                                    outtextxy(100,getmaxy div 2,'Нет сохраненных игр ');
                                    readln;
                                    cleardevice;
                                    showmouse;
                                    draw(0,0,'BMPI.bmp',true);
                                end;
                                showmouse;
                                end;

                                {Процедура, инициализирующая графический режим}
                                procedure grafinit;
                                var
                                    d,k:integer;
                                begin
                                    d:=detect;
                                    initgraph(d,k,pyt1);
                                end;

                                {Процедура выводющая на экран рисунок из BMP файла }
                                function Draw(x0,y0:integer; fname:string; transparent: boolean):integer;
                                label bye;
                                const
                                    color: array[0..15] of byte=(0,4,2,6,1,5,3,7,8,12,10,14,9,13,11,15);
                                type
                                    bmpinfo=record
                                        hl,h2: char;

```

```

        Size: longint;
        Reserved: longint;
        offset: longint;
        b: longint;
        Width: longint;
        Height: longint;
        Plans: word;
        bpp: word;
    end;
var
    f: file of bmpinfo;
    bmpf: file of byte;
    res: integer;
    info: bmpinfo;
var
    x, y: integer;
    b: byte;
    bh: byte;
    bl: byte;
    nb: integer;
    np: integer;
    tpcolor: byte;
    i, j: integer;
begin
    assign(f, fname);
    {$I-}
    reset(f);
    {$I+}
    res := IOResult;
    {Проверяет, существует ли файл с данным именем}
    If res <> 0 then
        begin
            Draw := -1;
            goto bye;
        end;
    read(f, info);
    close(f);
    {Проверяет, является ли файл BMP формата}
    if info.bpp <> 4 then
        begin
            Draw := -2;
            goto bye;
        end;
    x := x0;
    y := y0 + info.height;
    nb := (info.width div 8) * 4;
    if (info.width mod 8) <> 0 then
        nb := nb + 4;

```

```

assign (bmpf,fname);
reset (bmpf);
seek (bmpf,info.offset);
if transparent then
    begin
        read(bmpf,b);
        tpcolor:=b shr 4;
        seek (bmpf,info.offset);
    end
else tpcolor:= 17;
for i:=1 to info.height do
    begin
        np:=0;
        for j:=1 to nb do
            begin
                if not EOF (bmpf) then
                    read(bmpf,b);
                if np < info.width then
                    begin
                        bh:= b shr 4;
                        if bh <> tpcolor
                            then putpixel (x, y,
                                color [bh]);
                        inc(x);
                        inc(np);
                    end;
                if np<info.width then
                    begin
                        bl:=b and 15;
                        if bl <> tpcolor
                            then putpixel(x,y,color[bl]);
                        inc(x);
                        inc(np);
                    end;
            end;
        end;
        x:=x0;
        dec(y);
    end;
close(bmpf);
Draw:=info.height;
bye:
end;

{Процедура, выводящая меню на экран}
procedure MenuShow(x1,y1,x2,y2:integer;text:string; var
    mousepresed:boolean);
var k:byte;
begin

```

```

k:=green;
  {Проверяет, находится ли мышь в прямоугольнике с данными
  координатами}
  if mousein(x1,y1,x1+x2,y1+y2) then
    begin
      setcolor(blue);
      mousepresed:=True;
    end
  else
    begin
      setcolor(green);
      mousepresed:=false;
    end;
  outtextxy(x1+x2 div 2-textwidth(text) div 2,
  y1+(y2-textheight(text))div 2-5,text);
  setcolor(k);
end;

  {Процедура, выводящая на экран меню 'Об игре' или меню
  'Лицензия'}
  procedure info(a:word);
  var b:text;i:byte;
      k,d:integer;s:string;
  begin
    if a=1 then s:=pyt2+'info1.txt'
    else s:=pyt2+'info2.txt';
    i:=getcolor;
    setcolor(green);
    assign(b,s);
    {$I-}
    reset(b);
    {$I+}
    k:=IOResult;
    if k=0 then
      begin
        k:=0;
        hidemouse;
        cleardevice;
        settextstyle(1,0,1);
        draw(50,0,'11.bmp',true);
        while not eof(b) do
          begin
            readln(b,s);
            outtextxy(0,20*k,s);
            inc(k);
          end;
        repeat
          d:=inkey;

```



```

        until d=ESC;
        close(b);
        setcolor(k);
        cleardevice;
        draw(50,0,'BMP1.bmp',true);
        showmouse;
        settextstyle(1,0,4);
    end;
end;

{Выводит из файла на экран список рекордов}
procedure recordi;
var
    b:text;i:byte;
    k,d:integer;s,s1:string;
begin
    i:=getcolor;
    setcolor(red);
    assign(b,pyt2+'recordi.txt');
    {$I-}
    reset(b);
    {$I+}
    k:=IOResult;
    if k=0 then
        begin
            k:=1;
            hidemouse;
            cleardevice;
            settextstyle(1,0,4);
            readln(b,s);
            outtextxy(0,0,s);
            while not eof(b) do
                begin
                    str(k,s1);
                    readln(b,s);
                    if s<>" then
                        outtextxy(0,27*k,s1+' '+s);
                    inc(k);
                end;
            repeat
                d:=inkey;
            until d=ESC;
            close(b);
            setcolor(k);
            cleardevice;
            draw(50,0,'BMP1.bmp',true);
            showmouse;
            settextstyle(1,0,4);

```

```

        end;
end;

{Процедура показа и переназначения клавиш}
procedure Knopki(var newkey:key);
const
    text3:array[1..6]of string=('Вправо','Влево','Вниз','Пауза',
                                'Сохранить', 'Переворот');

type knop=array[1..6]of string;
var i,k:integer;a:array[1..6]of boolean;
    d:knop;

{Процедура, возвращающая название клавиши по её коду
Newkey – массив кодов клавиш;
D – массив названий клавиш}
procedure pokaz(newkew:key;var d:knop);
var i:byte;
begin
    for i:=1 to 6 do
        begin
            case newkey[i] of
                269:d[i]:='ВВОД';
                264:d[i]:='BACKSPACE';
                265:d[i]:='TAB';
                338:d[i]:='INSERT';
                339:d[i]:='DELETE';
                32:d[i]:='ПРОБЕЛ';
                329:d[i]:='PgUP';
                337:d[i]:='PgDOWN';
                327:d[i]:='HOME';
                328:d[i]:='ВВЕРХ';
                336:d[i]:='ВНИЗ';
                331:d[i]:='ВЛЕВО';
                333:d[i]:='ВПРАВО';
                315:d[i]:='F1';
                316:d[i]:='F2';
                317:d[i]:='F3';
                318:d[i]:='F4';
                319:d[i]:='F5';
                320:d[i]:='F6';
                321:d[i]:='F7';
                322:d[i]:='F8';
                323:d[i]:='F9';
                324:d[i]:='F10';
                389:d[i]:='F11';
                390:d[i]:='F12';
            else
                d[i]:=chr(newkew[i]);
            end

```

```

        end;
    end;
end;

{Процедура переназначения клавиш
 K – переназначаемая клавиша}
procedure perenaznaz(var k:integer);
var d:integer;
begin
    hidemouse;
    cleardevice;
    outtextxy(40,0,'Нажмите любую клавишу:');
    d:=0;
    repeat
        k:=inkey; {Функция считывающая код нажатой клавиши}
        if (k<>511)then
            d:=k
    until d<>0;
    cleardevice;
    outtextxy(40,0,'Кнопки:');
    showmouse;
end;

begin
    hidemouse;
    cleardevice;
    showmouse;
    repeat
        k:=inkey;
        outtextxy(40,0,'Кнопки:');
        pokaz(newkey,d);
        for i:=1 to 6 do
            begin
                outtextxy(10,0+i*50,text3[i]);
                menushow(300,0+i*50,150,50,d[i],a[i]);
                if (a[i])and((K=Enter)or(mousepressed)) then
                    begin
                        perenaznaz(newkey[i]);
                        pokaz(newkey,d);
                    end;
            end;
        until k=ESC;
    hidemouse;
    cleardevice;
    draw(50,0,'BMP1.bmp',true);
    showmouse;
end;

```

```

    {Процедура, обнуляющая матрицу игрового экрана }
PROCEDURE obnulenie(k:integer;var a:maska);
var i,j:integer;
begin
    for i:=1 to 31 do
        begin
            for j:=1 to 33 do
                begin
                    if (a[i,j]<>2)or(k=1) then
                        a[i,j]:=0;
                end;
            end;
        end;
    end;

    {Процедура, обнуляющая игровой экран}
procedure figureGreat(n:byte;var b:figura);
var i,j:byte;
begin
    for i:=1 to 4 do
        for j:=1 to 4 do
            b[i,j]:=fig[n,i,j];
        end;
    end;

    {Собственно сама игра}
procedure tetris(Nkey:key;a1:maska;b1:figura;
                i3,j3:integer;score1,level1:byte;h1,min1,sec1:word);
var
    a:maska;
    f,k,f1:integer;
    b,b2:figura;
    s,s3,s1,s2:string;
    score,level,i,j:byte;
    h,min,sec:word;
    T1,T2:time;m:boolean;

    {Подготовка экрана к игре}
procedure Podgotovka;
begin
    hidemouse;
    cleardevice;
    rectangle(209,8,461,310);
end;

{Процедура, прорисовывающая игровой экран}
procedure Ikran(a:maska);
var i,j:integer;
begin

```

```

        for i:=1 to 25 do
            begin
                for j:=1 to 30 do
                    begin
                        if (a[i+3,j]=1)or (a[i+3,j]=2) then
                            begin
                                SetFillStyle(1,red);
                                bar(200+i*10,1+j*10,210+i*10,9+j*10);
                                rectangle(200+i*10,-1+j*10,210+i*10,9+j*10);
                            end
                        else
                            begin
                                SetFillStyle(1,black);
                                bar(200+i*10,-1+j*10,210+i*10,9+j*10);
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

```

{Процедура переворота фигуры:
В – фигура}

```

procedure perevorot(var b:figura);
var i,j:byte;d:figura;
begin
    d:=b;
    for i:=1 to 4 do
        for j:=1 to 4 do
            b[i,j]:=d[5-j,i];
        end;
    end;
end;

```

{Процедура остановки фигуры
I0,J0-координаты фигуры;
В – фигура;
а – игровой экран;}

```

procedure figurestop(i0,j0:byte;b:figura;var a:maska);
var i,j:byte;
begin
    obnulenie(0,a);
    for i:=1 to 4 do
        begin
            for j:=1 to 4 do
                begin
                    if b[i,j]=1 then
                        a[i+i0,j+j0]:=2;
                    end;
                end;
            end;
        end;
    end;
end;
end;

```

```

    {Процедура удаления заполненной линии
    A – игровой экран;
    score – итоговый счет;
    level – итоговый уровень;}
procedure DelLine(var a:maska;var score:byte;var level:byte);
var i,j,j1,i1:byte;d:boolean;k:maska;
begin
    k:=a;
    for j:=1 to 30 do
        begin
            d:=true;
            for i:=4 to 28 do
                begin
                    if a[i,j]<>2 then d:=false;
                end;
            if d then
                begin
                    inc(score);
                    if (score div 10)<>0 then
                        level:=(score div 10)+1;
                    for i1:=4 to 28 do
                        for j1:=2 to j do
                            a[i1,j1]:=k[i1,j1-1];
                        end;
                end;
            end;
        end;
end;

    {Функция, перемещающая фигуру по экрану и следящая за тем
    чтобы фигура не выходила за рамки экрана и не попадала в другую
    фигуру
    b – фигура;
    a – игровой экран;
    i0,j0 – новые координаты фигуры }
function figure( b:figura; var a:maska;var i0,j0:byte):boolean;
var i,j:byte;d:boolean;k:maska;
    label metka,metka1;

    { Процедура, которая присваивает массиву игрового экрана
    массив фигуры
    i0, j0 – координаты фигуры;
    B – фигура;
    A– игровой экран;}
procedure prisiv(i0,j0:byte;b:figura;var a:maska);
var i,j:byte;
begin
    for i:=1 to 4 do
        for j:=1 to 4 do

```

```

        if b[i,j]=1 then
            a[i+i0,j+j0]:=1;
end;

begin
    figure:=false;
    k:=a;
    metka:
    prisl(i0,j0,b,a);
    d:=false;
    for i:=1 to 2 do
        for j:=1 to 4 do
            if (b[i,j]=1)and(k[i+i0,j+j0]=2) then
                d:=true;
        for i:=1 to 30 do
            begin
                if a[3,i]=1 then
                    d:=true;
            end;
        if d then
            begin
                obnulenie(0,a);
                a:=k;
                inc(i0);
                goto metka;
            end;
        metka1:
        prisl(i0,j0,b,a);
        d:=false;
        for i:=3 to 4 do
            for j:=1 to 4 do
                if (b[i,j]=1)and(k[i+i0,j+j0]=2) then
                    d:=true;
            for i:=1 to 30 do
                if a[29,i]=1 then
                    d:=true;
            if d then
                begin
                    obnulenie(0,a);
                    dec(i0);
                    a:=k;
                    goto metka1;
                end;
            d:=false;
            for i:=1 to 4 do
                for j:=1 to 4 do
                    if (b[i,j]=1)and(k[i+i0,j+j0+1]=2) then
                        d:=true;

```

```

for i:=3 to 33 do
  if a[i,30]=1 then
    d:=true;
  if d then
    begin
      figurestop(i0,j0,b,a);
      figure:=true;
    end;
end;

{Процедура, показывающая на экране количество очков,
уровень, следующую фигуру, время игры:
Score – количество очков;
Level –уровень;
В – следующая фигура;
T1 – время начала игры;
S3,s2,s1 – время, жизнь, очки;
H,min,sec – время игры}
procedure interfase(score,level:byte;b:figura;T1:time;
var s3,s2,s1:string;var h,min,sec:word);
var s:string;i,j:byte;
begin
  gettime(T2.hour,t2.minute,t2.second,t2.sec100);
  if T1.minute>T2.minute then
    begin
      dec(T2.hour);
      t2.minute:=t2.minute+60;
    end;
  if T1.second>T2.second then
    begin
      dec(T2.minute);
      t2.second:=t2.second+60;
    end;
  h:=t2.hour-T1.hour;
  min:=t2.minute-T1.minute;
  sec:=t2.second-T1.second;
  str(h,s);
  str(min,s1);
  str(sec,s2);
  s3:=s+'.'+s1+'.'+s2;
  outtextxy(520,10,'Время:');
  outtextxy(520,50,s3);
  str(level,s2);
  outtextxy(20,10,'Уровень:');
  outtextxy(163,10,s2);
  str(score,s1);
  outtextxy(20,50,'Очки:');
  outtextxy(163,50,s1);

```



```

outtextxy(510,90,'фигура:');
for i:=1 to 4 do
    begin
        for j:=1 to 4 do
            begin
                If b[i,j]=1 then
                    begin
                        rectangle(510,128,622,212);
                        SetFillStyle(1,6);
                        bar(510+i*20,110+j*20,530+i*20,130+j*20);
                        rectangle(510+i*20,110+j*20,530+i*20,130+j*20);
                    end
                else
                    begin
                        SetFillStyle(1,black);
                        bar(510+i*20,110+j*20,530+i*20,130+j*20);
                    end;
                end;
            end;
        end;
    end;

    {Быстрое сохранение игры }
procedure save(a:maska;b:figura;i,j:integer;score,
    level:byte;h,min,sec:word);
var f:text;i1,j1:byte;
begin
    assign(f,pyt2+'save.txt');
    rewrite(f);
    for i1:=1 to 31 do
        for j1:=1 to 33 do
            writeln(f,a[i1,j1]);
        for i1:=1 to 4 do
            for j1:=1 to 4 do
                writeln(f,b[i1,j1]);
            writeln(f,i);
            writeln(f,j);
            writeln(f,score);
            writeln(f,level);
            writeln(f,h);
            writeln(f,min);
            writeln(f,sec);
            close(f);
        end;
    end;

    {Процедура добавления нового рекорда
    H, m, sec – время игры}
procedure recordi(h,m,sec:word);
var

```

```

b:text;
k:byte;
s,h1,m1,sec1:string;
d,f:word;
begin
  assign(b,pyt2+'recordi.txt');
  {$I-}
  append(b);
  {$I+}
  k:=IOResult;
  str(h,h1);
  str(m,m1);
  str(sec,sec1);
  If k=0 then
    begin
      cleardevice;
      outtextxy(200,180,'Введите ваше имя:');
      f:=0;
      repeat
        d:=inkey;
        if d<>511 then
          begin
            s:=s+chr(d);
            Outtextxy(250+f*20,250,chr(d));
            inc(f);
          end;
        until d=space;
        writeln(b,s,h1+'.'+m1+'.'+sec1:30);
      end;
      close(b);
    end;
end;

{Пауза
Т1 – время паузы}
procedure pauza(var t1:time);
var t2,t3:time;f1:word;
begin
  gettime(T2.hour,t2.minute,t2.second,t2.sec100);
  OutTextXY(280,200,'Пауза');
  repeat
    f1:=inkey;
    until f1=nkey[4];
    gettime(T3.hour,t3.minute,t3.second,t3.sec100);
    T1.hour:=T1.hour+T3.hour-T2.hour;
    t1.minute:=t1.minute+t3.minute-t2.minute;
    t1.second:=t1.second+t3.second-t2.second;
  end;
end;

```

```

    {Начало основной процедуры TETRIS }
BEGIN
    randomize;
    podgotovka;
    obnulenie(1,a);
    gettime(T1.hour,t1.minute,t1.second,t1.sec100);
    b:=b1;
    a:=a1;
    if min1>T1.minute then
        begin
            dec(T1.hour);
            t1.minute:=t1.minute+60;
        end;
    if sec1>T1.second then
        begin
            dec(T1.minute);
            t1.second:=t1.second+60;
        end;
    T1.hour:=T1.hour-h1;
    t1.minute:=t1.minute-min1;
    t1.second:=t1.second-sec1;
    figuregreat(random(8)+1,b2);
    score:=score1;
    level:=level1;
    s3:="";
    s1:='0';
    s2:='1';
    i:=i3;
    j:=j3;
    k:=getcolor;
    m:=false;
    repeat
        f:=inkey;
        if (f=nkey[1])and(i<28) then
            inc(i);
        if (f=nkey[2])and(i>1)then
            dec(i);
        if (f=nkey[3])and(j<30) then
            inc(j);
        if f=nkey[6] then
            perevorot(b);
        if figure(b,a,i,j) then
            begin
                for i:=4 to 28 do
                    if a[i,3]=2 then
                        m:=true;

```

```

        if m then
            begin
                ikran(a);
                obnulenie(1,a);
                repeat
                    OutTextXY(250,200,'Проигрыш');
                until keypressed;
                break;
            end;
            obnulenie(0,a);
            i:=random(24)+4;
            j:=0;
            b:=b2;
            figuregreat(random(8)+1,b2);
        end;
        delline(a,score,level);
        ikran(a);
        if f=nkey[4] then
            pauza(T1);
            if f=nkey[5] then
                save(a,b,i,j,score,level,h,min,sec);
                obnulenie(0,a);
                setcolor(black);
                outtextxy(520,50,s3);
                outtextxy(163,10,s2);
                outtextxy(163,50,s1);
                setcolor(k);
                interfase(score,level,b2,t1,s3,s2,s1,h,min,sec);
                delay(200 div level);
                inc(j);
                if level=9 then
                    begin
                        recordi(h,min,sec);
                        break;
                    end;
            until f=ESC;
            obnulenie(1,a);
            cleardevice;
            draw(0,0,'BMPI.bmp',true);
            showmouse;
        end;
    END.

```

Учебное издание

ПРОГРАММИРОВАНИЕ

Методические рекомендации

Редактор *Е.С. Гарбузняк*
Компьютерная верстка *Н.В. Чернега*

Формат 60×90/16
Уч. изд. 6,3 п.л. Тираж 10 экз.