

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

**КУРСОВОЕ ПРОЕКТИРОВАНИЕ
ПО ОСНОВАМ ПРОГРАММИРОВАНИЯ**

Методические рекомендации

Рыбница, 2015

УДК 681.3.06
ББК 32.973.2-018
К93

Составители:

Е.С. Гарбузняк, преп.

Н.В. Чернега, преп.

О.М. Нагаевский, преп.

Рецензенты:

Л.К. Скодорова, канд. социол. наук, доцент, каф. прикладной информатики, филиал ПГУ им. Т.Г. Шевченко в г. Рыбница

Л.Я. Козак, канд. техн. наук, доцент, каф. физики, математики и информатики, филиал ПГУ им. Т.Г. Шевченко в г. Рыбница

К93 Курсовое проектирование по основам программирования: Методические рекомендации / Сост.: Е.С. Гарбузняк, Н.В. Чернега, О.М. Нагаевский. – Рыбница, 2015. – 115 с. (в обл.)

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

Методические рекомендации предназначены для студентов I курса направления «Программная инженерия». Могут быть использованы студентами других направлений.

УДК 681.3.06

ББК 32.973.2-018

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

© Е.С. Гарбузняк, Н.В. Чернега, О.М. Нагаевский,
составление, 2015

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. ОБЩИЕ ПОЛОЖЕНИЯ О КУРСОВЫХ РАБОТАХ	7
1.1. Задачи курсовой работы.....	7
1.2. Порядок подготовки и защиты курсовой работы	7
1.3. Общие требования к выполнению курсовой работы	9
1.4. Общие требования к структуре, содержанию и оформлению курсовой работы.....	9
1.5. Оценка курсовой работы.....	10
2. ПОНЯТИЙНЫЙ АППАРАТ ИССЛЕДОВАНИЯ.....	12
2.1. Актуальность, степень изученности и научной разработанности темы исследования	12
2.2. Объект и предмет исследования	13
2.3. Цель и задачи исследования	13
2.4. Новизна и практическая значимость	14
3. ВЫБОР ТЕМЫ И ПЛАНИРОВАНИЕ КУРСОВОЙ РАБОТЫ... ..	15
3.1. Выбор темы исследования	15
3.2. Составление рабочего плана.....	16
3.3. Работа с литературой.....	18
4. НАПИСАНИЕ КУРСОВОЙ РАБОТЫ	20
4.1. Разработка структуры работы	20
4.2. Подготовка введения.....	22
4.3. Работа над основной частью.....	23
4.4. Подготовка заключения и формулирование выводов	26
5. ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ	28
5.1. Правила оформления текстового материала	28
5.2. Правила оформления иллюстраций	30
5.3. Правила оформления формул.....	31
5.4. Правила оформления таблиц	32
5.5. Правила оформления библиографического списка.....	35
5.6. Правила оформления библиографических ссылок.....	45

5.7. Правила оформления приложений	47
5.8. Правила оформления блок-схем	48
6. ЗАЩИТА КУРСОВОЙ РАБОТЫ.....	51
6.1. Подготовка доклада	51
6.2. Подготовка презентации	53
6.3. Защита курсовой работы	55
СПИСОК ЛИТЕРАТУРЫ	58

ПРИЛОЖЕНИЕ

ВВЕДЕНИЕ

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

Целью курсовой работы является углубление знаний и расширение навыков по разработке алгоритмов и их реализации на персональном компьютере. Курсовая работа выполняется на языке высокого уровня и предполагает использование графических возможностей языка, работу с текстами, файловыми структурами.

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.

В результате выполнения курсовой работы студент приобретает практические навыки постановки задачи, ее алгоритмизации, тестирования разработанного программного продукта, а также самостоятельного освоения нового материала в соответствии заданию курсовой работы.

Дисциплина «Основы программирования» относится к базовой части профессионального цикла (Б3.Б.2) подготовки студентов по направлению «Программная инженерия». Целью дисциплины является освоение теоретических и практических основ программирования.

Курсовая работа по основам программирования позволяет выполнить основные задачи дисциплины: получение представления об основах программирования; приобретение знаний о принципах организации, структурах технических и программных средств, используемых в программировании; 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 «Основы программирования».

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 «Основы программирования», приведенный в приложении методических рекомендаций, позволит студентам избежать типичных ошибок при выполнении курсовых работ и грамотно их оформить.

Методические рекомендации предназначены для студентов I курса направления «Программная инженерия». 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 ниже 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 дисциплине «Основы программирования» представлены ниже (табл. 1.1).

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

Группа требований	Описание основных требований
Структура	Типовая структура курсовой работы: – титульный лист; – оглавление; – введение; – основная часть; – заключение; – список литературы; – 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	Стил ь изложения – научный. Требования к оформлению курсовой работы рассмотрены в соответствующем разделе.

В случае невыполнения требований к структуре, содержанию и оформлению курсовая работа к защите не допускается.

1.5. Оценка курсовой работы

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

Оценка вносится в экзаменационную ведомость и зачетную книжку. Отрицательная оценка в зачетную книжку не вносится.

Классификация и описание критериев оценки курсовых работ приведены ниже (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.

Полные названия курсовых работ с оценками вносятся в приложение к диплому в соответствии с распоряжением УАП ПГУ им. Т.Г. Шевченко.

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 курсовой работы рекомендуется использовать типовые лексико-синтаксические конструкции, приведенные ниже (табл. 2.1).

Таблица 2.1

Рекомендуемые лексико-синтаксические конструкции

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 ...
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 ниже (рис. 3.1).

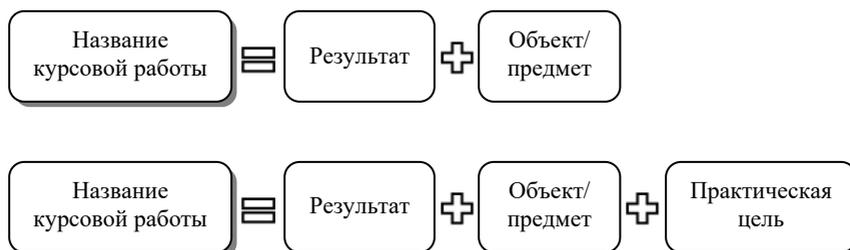


Рис. 3.1. Схема сочетания возможных информационных блоков названия темы курсовой работы

Примеры формулирования тем курсовых работ в зависимости от направлений и информационных блоков приведены ниже (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.

Ниже приведен перечень тем курсовых работ по дисциплине «Основы программирования» различного уровня сложности:

1. Анализ функции многочлена.
2. Реализация трехмерных фигур из составных частей.
3. Построение сечения четырехмерного куба.
4. Реализация модели «Облака».
5. Реализация символьного дифференцирования.
6. Реализация форматирования исходного кода.
7. Сравнение исходных текстов программы.
8. Построение фигур из треугольников.
9. Построение эквидистанты.
10. Реализация управления телепрограммами.
11. Реализация построения лабиринта.
12. Реализация стенографического алгоритма.

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 состоянием.

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

План работы имеет произвольную форму. Ниже (табл. 3.2) представлен пример плана-графика работ по основам программирования.

Таблица 3.2

Пример плана-графика работ

Вид документа, отчетной работы	Срок сдачи	Принимающее лицо / подразделение
План работы, проект введения	16 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)	21 апреля	Научный руководитель
Материалы 2 раздела (законченный вариант)	10 мая	Научный руководитель
Окончательный вариант курсовой работы	26 мая	Научный руководитель
Защита курсовой работы	2 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 курсовых работ составляет 10-15 источников.

Правила оформления списка литературы в зависимости от вида источника информации приведены в разделе «Правила оформления библиографического списка» методических рекомендаций.

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

ОГЛАВЛЕНИЕ

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

Рекомендуемая структура курсовой работы по дисциплине «Основы программирования» представлена ниже.

Титульный лист

ОГЛАВЛЕНИЕ

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.

1.4. Выбор среды программирования.

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, 2015. Образец титульного листа приведен в приложении методических рекомендаций.

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

Выводы к соответствующим разделам не выделяются в отдельные пункты, а располагаются в конце раздела для подведения общих итогов, особенностей выполнения определенного раздела.

Заголовки разделов и подразделов должны точно отражать содержание относящегося к ним текста. Рекомендуется в заголовках 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 курсовой работы;
- определяется область применения, значимость результатов;
- описывается структура курсовой работы.

Ниже (табл. 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 курсовой работы.

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.

Ниже (рис. 4.1) графически изображена основная часть курсовой работы, представленная в виде пирамиды.

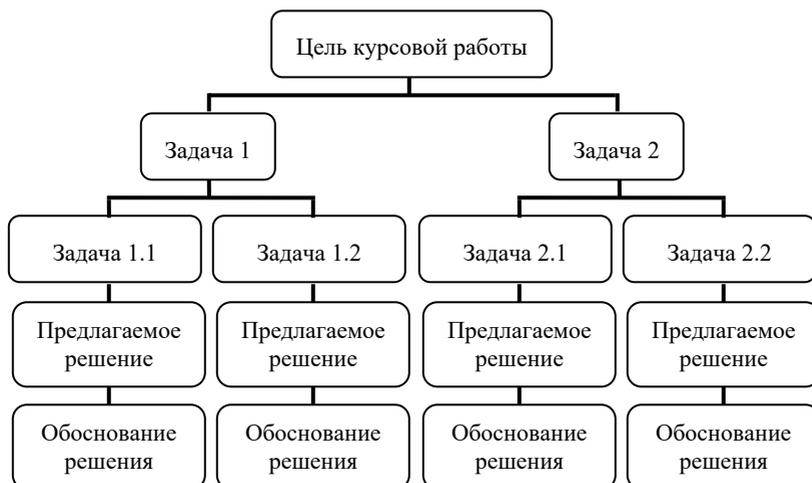


Рис. 4.1. Основная часть курсовой работы, представленная в виде пирамиды

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.

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

Таблица 4.2

Список лексико-синтаксических конструкций, рекомендуемых при изложении основной части

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

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.

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. 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.

Ниже (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

При разработке заключительной части 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.
3. Связывать 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 апробации.
7. Излагать предложения по внедрению решений.
8. Обосновывать основные направления для дальнейшего развития исследования.

Рекомендуемый объем заключения курсовой работы – 2-3 страницы.

После заключения в курсовой работе приводят список литературы 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.

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-четырёх 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).

Правила оформления текстового материала курсовой работы приведены ниже (табл. 5.1).

Таблица 5.1

Правила оформления текстового материала курсовой работы

Группа правил	Содержание
Оформление основного текста	<p>Формат страницы – А4 (210х297 мм), без рамки, книжная ориентация, левое поле – 30 мм, верхнее и нижнее поля – 20 мм, правое – 15 мм.</p> <p>Шрифт – <i>Times New Roman</i>, размер – 14 pt, цвет – чёрный.</p> <p>Строчные буквы.</p> <p>Выравнивание – по ширине.</p> <p>Межстрочный интервал – полуторный.</p> <p>Абзацный отступ 1,25.</p>
Нумерация страниц	<p>Арабские цифры, сквозная нумерация по всему тексту работы.</p> <p>Первая страница – титульный лист, на котором нумерация страниц не ставится, на следующей странице ставится цифра 2.</p> <p>Порядковый номер страницы – нижнее поле страницы, справа.</p>
Оформление заголовков разделов	<p>Шрифт – <i>Times New Roman</i>, размер – 14 pt, полужирный, цвет – чёрный.</p> <p>ВСЕ БУКВЫ заголовка раздела ПРОПИСНЫЕ.</p> <p>Выравнивание – по центру.</p> <p>Межстрочный интервал – одинарный. Интервал перед заголовком – 0 pt, после заголовка – 20 pt.</p> <p>Точка в конце заголовка не ставится.</p> <p>Не подчеркивается.</p> <p>Переносы слов не допускаются.</p> <p>Если заголовок состоит из нескольких строк, то интервал между ними – одинарный.</p>
Оформление заголовков подразделов	<p>Шрифт – <i>Times New Roman</i>, размер – 14 pt, полужирный, цвет – чёрный.</p> <p>Все буквы заголовка подраздела, кроме первой, строчные.</p> <p>Выравнивание – по центру.</p> <p>Межстрочный интервал – полуторный. Интервал перед подзаголовком – 12 pt, после подзаголовка – 8 pt.</p> <p>Точка в конце заголовка не ставится.</p> <p>Не подчеркивается.</p> <p>Переносы слов не допускаются.</p> <p>Если подзаголовок состоит из нескольких строк, то интервал между ними – одинарный.</p>
Нумерация разделов и подразделов	<p>Нумерация разделов – арабские цифры, в пределах всей работы.</p> <p>Номер подраздела состоит из номеров раздела и подраздела, разделенных точкой.</p>

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

4. 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».

5. Иллюстрации размещают под текстом, в котором впервые дана ссылка на них, или на следующей странице, а при необходимости – в приложении.

6. 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 двоеточия, а между её элементами – точка с запятой.

Пример оформления иллюстрации приведен ниже на рисунке (рис. 5.1).

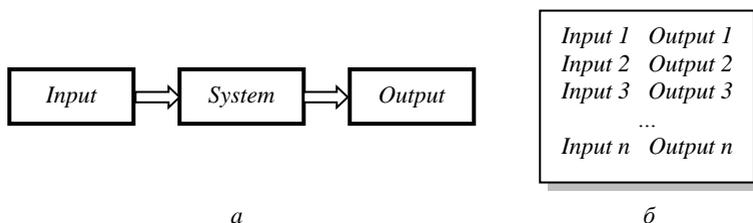


Рис. 5.1. Система ввода-вывода: *a* – взаимодействие системы с внешним миром; *б* – общий вид экспериментальных данных

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

5.3. Правила оформления формул

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

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)».

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.

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.

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 работе несколько таблиц.

**Характеристики
целочисленных типов**

Таблица 5.1

Тип	Диапазон значений	Требуемая память (байт)
<i>Integer</i>	- 32768 .. 32767	2
<i>ShortInt</i>	- 128 .. 127	1
<i>LongInt</i>	- 2147483648 .. 2147483647	4
<i>Byte</i>	0 .. 255	1
<i>Word</i>	0 .. 65535	2

Боковик
Прографка

Нумерационный заголовок
 Тематический заголовок
 Головка
 Хвост

Рис. 5.2. Элементы таблицы

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.

Таблицы, как и рисунки, имеют названия и порядковую нумерацию. Название должно отражать содержимое таблицы, быть точным и кратким. Правила оформления таблиц приведены ниже (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 вверху 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 – <i>Times New Roman</i> , кегль 12. В конце заголовков и подзаголовков таблиц точки не ставят.
Оформление основной части таблицы	Одинарный межстрочный интервал, без отступа, шрифт – <i>Times New Roman</i> , 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

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

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».

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

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 литературы должен содержать перечень источников, использованных при выполнении работы, как правило, список содержит не менее 10 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 литературы размещается после текста работы и предшествует приложениям. Сведения о наличии списка литературы отражаются в оглавлении [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; опублик. 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; опублик. 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. Тутьгин. – 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. – М.: 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. – М.: Юристъ, 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. – М.: 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 сведениях 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: Либерия, 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]. – М.: 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. – М.: ИНФРА-М, 2000. – 1 эл. опт. диск (CD-ROM).

Энциклопедия классической музыки [Электронный ресурс]. – Электрон. дан. – М.: 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.rissi.ru>

Шпринц, Лев. Книга художника: от миллионных тиражей – к единичным экземплярам [Электронный ресурс] / Л. Шпринц. – Электрон. текстовые дан. – М.: [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. – М., 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. – М.: 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 тексте 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 ниже (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	<p>Оформляется как примечание, вынесенное из текста документа вниз страницы.</p> <p>При нумерации применяют единообразный порядок: сквозную нумерацию по всему тексту, в пределах каждого раздела, части и т.д.</p> <p>Для связи с текстом документа используют знак сноски.</p> <p>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).</p>
Затекстовая	<p>Оформляется как перечень библиографических записей, помещенный после текста документа или его составной части.</p> <p>Используется сплошная нумерация для всего текста работы или для отдельных разделов, частей и т.п.</p> <p>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.</p> <p>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.</p>

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.

В курсовой работе, как правило, принято использовать затекстовые ссылки так, как показано в примере 2.

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 оглавлении 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 ниже (табл. 5.5).

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 $2a$.

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.	

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.

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.

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.

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 (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.

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 20 pt, 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 обязательно должны иметь название.

6. Содержание слайдов должно соответствовать выступлению.

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, курс, специальность. 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 курсовой работы.
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.

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 по дисциплине «Основы программирования» 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 в соответствии с утвержденным на кафедре графиком.

Защита курсовой работы по дисциплине «Основы программирования» может проводиться:

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

В первом случае проверка курсовой 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, в том числе куратор группы, 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 рисунке ниже (рис. 6.1).

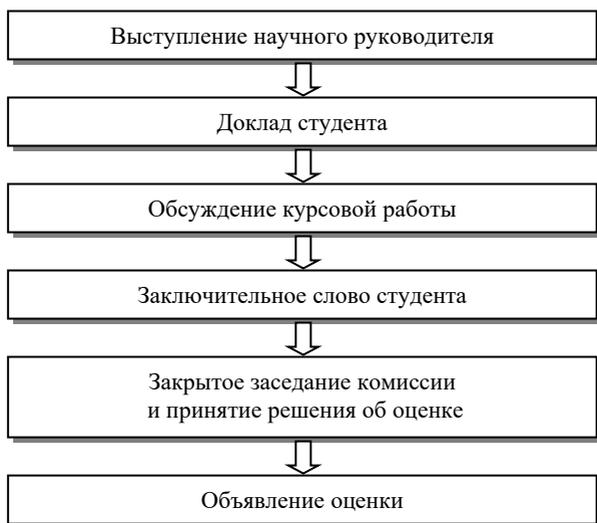


Рис. 6.1. Процедура защиты курсовой работы

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]. – М.: 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.: 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.: Либерей-Библинформ, 2007. – 144 с.

ПРИЛОЖЕНИЕ

Образец курсовой работы по дисциплине «Основы программирования»

**ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Т.Г. ШЕВЧЕНКО
Рыбницкий филиал**

Кафедра физики, математики и информатики

КУРСОВАЯ РАБОТА по дисциплине «ОСНОВЫ ПРОГРАММИРОВАНИЯ» на тему: «РЕАЛИЗАЦИЯ ИГРЫ «БОМБЕРЫ»

Студента I курса
направления «Программная инженерия»
профиля «Разработка программно-
информационных систем»
Думинюка Богдана Юрьевича

Научный руководитель:
преподаватель
Чернега Наталья Владимировна

Рыбница, 2014

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. ИССЛЕДОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	5
1.1. Графика в <i>Turbo C++</i>	5
1.2. Инициализация и закрытие графического режима.....	8
1.3. Правила игры	9
1.4. Обзор существующих программных продуктов.....	9
2. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА, ОПИСАНИЕ РЕЗУЛЬТАТОВ.....	11
2.1. Меню и игровое поле	11
2.2. Переназначение управления	12
2.3. Движение персонажей.....	14
2.4. Разрушение объектов	14
2.5. Бонусная система	17
2.6. Помощь и динамическая статистика.....	17
2.7. Анализ полученных результатов.....	18
ЗАКЛЮЧЕНИЕ.....	19
СПИСОК ЛИТЕРАТУРЫ	20
ПРИЛОЖЕНИЯ	

ВВЕДЕНИЕ

С развитием технологий компьютеры все больше и больше вливаются в повседневную жизнь человека. Персональные компьютеры – это уже не просто машины для математических вычислений, их общедоступность значительно расширила область применения. Не обошлось без игр. Индустрия игр быстро набирала обороты, стремительно развивалась, завоевывая все большую и большую популярность у подрастающего поколения. И если первые игры отличались простотой дизайна и логики, то современная компьютерная игра – очень сложная программа, дело рук большого коллектива разработчиков. Компьютерные игры занимают не последнее место в жизни современных людей.

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 «Бомберы».

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

Для достижения данной цели были поставлены следующие задачи:

- изучить теоретические сведения, необходимые для решения данной задачи;
- систематизировать и обобщить полученные знания;
- изучить аналогичные программные продукты;
- создать программный продукт, реализующий игру «Бомберы»;
- 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 проведены обзор графического режима в *Turbo C++*, обзор существующих программных продуктов, изучены правила игры «Бомберы».

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

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

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

1. ИССЛЕДОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Графика в *Turbo C++*

Turbo C++ поставляется с полной библиотекой графических функций, позволяющих создание экранных графиков и диаграмм. Графические функции доступны только для 16-разрядных приложений *DOS* [2].

Любое изображение на экране монитора формируется из отдельных элементов – пикселей (от англ. *Pixel – Picture Element* – элемент изображения). Экран монитора можно рассматривать как матрицу пикселей. Для получения того или иного изображения на экране монитора как в графическом, так и в текстовом режимах, необходимо заставить светиться строго определенную группу пикселей. В текстовом режиме на экран могут выводиться только определенные символы, образы которых хранятся в постоянной или оперативной памяти компьютера, а управление отдельными пикселями невозможно. В графическом режиме появляется возможность управления отдельными пикселями, что позволяет формировать любые изображения.

С технической точки зрения управлением монитора занимается специальное устройство компьютера – видеоадаптер. Именно согласно его сигналам зажигаются и гаснут отдельные точки на экране монитора. Конструктивно видеоадаптер – это весьма сложное электронное устройство, управляемое собственным микропроцессором. Качество изображения, получаемое на экране монитора, во многом зависит не только от самого монитора, но и от типа видеоадаптера. Это может быть монохромный дисплейный адаптер (*MDA*) для базового (только текстового) дисплея, либо это может быть графический адаптер, например, цветной графический адаптер (*CGA*), улучшенный графический адаптер (*EGA*), монохромный графический адаптер *Hercules* или видеографическая матрица (*VGA/SVGA*). Каждый из этих адаптеров может работать в нескольких режимах. Режим определяет величину экрана – 80 или 40 символов в строке (только в текстовом режиме), разрешающую способность экрана (только в графическом режиме) и тип дисплея (цветной или чёрно-белый). Какой бы адаптер ни был установлен на компьютере, можно использовать один и тот же набор графических процедур и

функций *Turbo C* благодаря тому, что их конечная настройка на конкретный адаптер осуществляется автоматически. Эту настройку выполняют графические драйверы.

Для получения изображения на экране монитора необходимо поместить в видеопамять определенную информацию. Графическое программирование на таком уровне представляется весьма непростой и трудоемкой задачей, требующей знания особенностей того или иного видеоадаптера.

Чтобы сделать процесс создания графических изображений более простым и эффективным, фирма *Borland International* разработала специальную библиотеку *GRAPH*, а также набор графических драйверов, позволяющих работать с различными типами видеоадаптеров [5]. Графический драйвер – это вспомогательная программа-посредник, обеспечивающая взаимодействие пользовательских программ с конкретным графическим устройством. Графические драйверы хранятся в файлах с расширением **.BGI* (*Borland Graphic Interface*). Обычно эти файлы расположены в каталоге *\BC\BGI* (или *\TC\BGI*).

Основными функциями библиотеки *GRAPH*, позволяющие работать с графикой, являются:

- *initgraph(&grdriver, &grmode, path)* – функция инициализации графического режима, где *&grdriver, &grmode, path* – параметры загружаемого режима, где *grdriver* и *grmode* – переменные типа *integer*, а *path* – путь к загружаемому драйверу графического режима;

- *closegraph()* – функция закрытия графического режима данная функция без каких-либо параметров и закрывает текущий графический режим;

- *graphresult()* – возвращает код ошибки для последней неудачно выполненной операции;

- *arc(int x, int y, int stangle, int endangle, int radius)* – функция рисует дугу по заданным координатам *x, y*, начальным и конечным углом *stangle, endangle* и заданным радиусом *radius*;

- *bar(int left, int top, int right, int bottom)* – функция рисует прямоугольник, начиная от левого верхнего угла с координатами *left, top* до правого нижнего угла с координатами *right, bottom*;

- *circle(int x, int y, int radius)* – функция рисует окружность с центром в точке с координатами *(x, y)* и радиусом *radius*;

– *cleardevice()* – функция очищает экран в графическом режиме и возвращает текущую позицию указателя в точку с координатами $(0, 0)$;

– *ellipse(int x, int y, int stangle, int endangle, int xradius, int yradius)* – функция рисует эллипс с координатами в точке (x, y) , начальным и конечным углами *stangle*, *endangle*, и радиусами *xradius*, *yradius* по осям x и y ;

– *fillellipse (int x, int y, int xradius, int yradius)* – функция рисует закрашенный эллипс;

– *floodfill(int x, int y, int border)* – область, ограниченная цветной границей закрашивается установленным образцом и цветом;

– *getmaxx()* – возвращает значение типа *integer*, которое соответствует максимальному количеству точек по оси x для текущего графического режима;

– *getmaxy()* – возвращает значение типа *integer*, которое соответствует максимальному количеству точек по оси y для текущего графического режима;

– *getpixel(int x, int y)* – функция возвращает номер цвета точки, находящейся по координатам (x, y) ;

– *setcolor(int color)* – устанавливает цвет по его номеру заданному переменной *color*;

– *setfillstyle(int pattern, int color)* – функция устанавливает образец и цвет заливки;

– *settextstyle(int font, int direction, int charsize)* – устанавливает характеристики текста: стиль, положение (горизонтальное или вертикальное), размер шрифта.

В растровой компьютерной графике экран представляет собой прямоугольный массив пикселей (рис. 1.1). Любое изображение строится как композиция из светящихся или погашенных пикселей. Каждый пиксел адресуется двумя целыми числами – номером по горизонтали x и номером по вертикали y .

Значения X_{max} и Y_{max} зависят от текущего графического видеорежима. В случае адаптера *VGA* и режима *VGAHI* $X_{max} = 639$, а $Y_{max} = 479$.

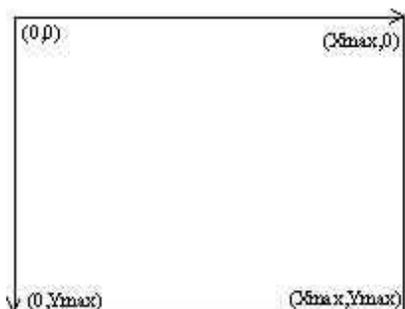


Рис. 1.1. Система координат на экране монитора

В библиотеке *graphics.h* определены две функции *getmaxx()* и *getmaxy()*, возвращающие текущие значения параметров *Xmax* и *Ymax* соответственно.

1.2. Инициализация и закрытие графического режима

Прежде чем программировать в графическом режиме, необходимо его инициализировать. Для этого необходимо проделать следующие действия:

1. Включить поддержку графической библиотеки в компиляторе. В *Turbo C++* это делается следующим образом: *options/linker/libraries*. Необходимо поставить крестик около *Graphics library* и применить изменения.

2. Подключить библиотеку *Graphics.h* (библиотеку графических функций): *#include <Graphics.h>*.

3. Инициализировать графический режим. Для этого необходимо описывать переменные, которые определяют графический драйвер и монитор: *int gdriver, gmode; gd=Detect;* (*detect* – самоопределение лучшего режима). Инициализация графического режима: *initgraph(gdriver, gmode, «путь к файлу»)*.

С этого момента все графические средства доступны пользователю.

Также можно указать для функции *initgraph* использование конкретного графического драйвера и конкретный режим, либо задать автообнаружение установленного видеоадаптера и выбор соответствующего драйвера уже во время выполнения.

После того, как графический драйвер загружен, можно определить его имя при помощи функции *getdrivename*, а число поддерживаемых драйвером режимов – при помощи функции

getmaxmode. Функция *getgraphmode* сообщает, какой графический режим запущен в данный момент. Имея номер режима, можно определить его имя при помощи функции *getmodename*, изменить графический режим при помощи функции *setgraphmode*. Вернуть исходный видеорежим (тот, который был установлен до инициализации графики) можно с помощью *restorecrtmode*. Функция *restorecrtmode* вернет экран в текстовый режим, но не закроет при этом графическую систему (загруженные шрифты и драйверы останутся в памяти).

И наконец, закончив работу в графике, необходимо вызвать функцию *closegraph* для того, чтобы закрыть графическую систему. Функция *closegraph* выгружает драйвер из памяти и восстанавливает первоначальный видеорежим (через обращение к *restorecrtmode*).

1.3. Правила игры

Бомберы – игра для двух, трёх или четырёх человек. Игровое поле представляет собой прямоугольник. Есть взрываемые и не взрываемые блоки (не взрываемые блоки показаны на игровом поле серым цветом, взрываемые – коричневым). Под взрываемыми блоками могут встречаться бонусы: увеличение дальности поражающего действия, поправка здоровья на 50%. Чтобы записать их себе в актив, необходимо взять их, коснувшись бонуса своим бомбером.

Игроки начинают сражение в противоположных углах игрового поля. Чтобы встретиться и вступить в сражение, игроки должны взорвать блоки (только взрываемые), которые их разделяют. Имеет смысл стараться взорвать как можно больше блоков самому, опередив соперников, поскольку скрытые под ними бонусы позволяют значительно увеличить возможности бомбера. Вступив в сражение, необходимо ставить бомбы так, чтобы при взрыве бомба нанесла урон противнику. Однако очень важно самому не попадать под действие своей или чужой бомбы.

Примечание: если здоровье бомбера стопроцентно, и ему удастся взять из взрываемого блока бонус «поправка здоровья на 50%», то здоровье бомбера увеличивается до 150%.

1.4. Обзор существующих программных продуктов

В первых играх, появившихся на компьютерах, часто использовались лабиринты, по которым бегали странные пиксельные существа, сражавшиеся или убегающие друг от друга, «поедавшие» сердечки и другие полезные элементы, пробивавшие стены с помощью хитроумных приспособлений. «Бомберы» сделаны по такому же принципу. Создания разных расцветок и форм, со смешными выражениями бегают по запутанным коридорам, не забывая подкладывать бомбы, чтобы освободить проход.

Игра позволяет не только взрывать и крушить все кругом, но и просчитывать свои действия, быстро реагировать на угрозы, убегать от бомб и прятаться от соперников. Все это развивает ловкость и сообразительность, помогает в мгновение ока принять нужное решение и выполнить главную цель игры – разрушить лабиринт и уничтожить противников, а самому остаться целым и невредимым.

Аналоги этой игры реализованы практически на всех современных компьютерах, включая мобильные телефоны и игровые видеоприставки.

Первые варианты реализации данной игры были осуществлены для *NES (Nintendo Entertainment System)* – восьмибитной игровой консоли, популярной во второй половине 1980-х и начале 1990-х годов, известной в России как *Dendy*.

Существует несколько вариантов этой игры:

- *bomberman* – 2D игра;
- *bomberman* – приложение для мобильных телефонов;
- бомберы – игры в социальных сетях;
- *bomberman* – приложение для *NES (Dendy)*.

Наибольшую популярность приобрела реализация игры для *NES*, в которой игра проходит в режиме «прохождения». Игрок управляет персонажем в виде человека на прямоугольном уровне, содержащем неразрушаемые и разрушаемые кирпичные блоки, а в зависимости от уровня по полю перемещаются разные враги: амёбы, луковицы, монетки, шары и прочее. В некоторых реализациях игры попадают трудно уничтожаемые боссы.

Первая часть данной игры для игровой приставки *NES* выпускалась отдельно, а также неофициально включалась в

многоигровки(пример: *9999 in 1*). Главный персонаж игры – бомбермен (представленный, как робот, которому нужно сбежать с завода по производству бомб). Задача героя – расставлять по уровню бомбы, взрывать стены, мешающие проходу и врагов, которые убивают прикосновением.

В реализации игры в социальных сетях режимом игры является «соревнование». Играют несколько человек, пока не останется один победитель.

После изучения теоретического материала было решено создать программный продукт, реализующий игру «Бомберь», аналогичную игре в социальных сетях, в среде программирования *Turbo C++*.

2. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА, ОПИСАНИЕ РЕЗУЛЬТАТОВ

2.1. Меню и игровое поле

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

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

«Бомберы» – 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 система управления программой – меню. Каждому пункту меню присвоено название, соответствующее функциональному назначению. Это облегчает работу пользователя.

Что касается второго пункта, здесь применены различные цвета. В сочетании они не несут никакой психологической нагрузки, поэтому времяпрепровождение для человека будет приятным.

Для использования меню была составлена пользовательская подпрограмма-функция *menu (void)*, при запуске которой на экране появляется окно, состоящее из трех пунктов (рис. 2.1).



Рис. 2.1. Меню

Для осуществления возможности переключения пунктов меню был использован оператор *switch()*. В конце подпрограммы находятся три условных оператора, которые реализуют действия из пунктов меню соответственно. В процессе выполнения функции *menu(void)* может быть выполнен только один из условных операторов.

В случае, если условие первого условного оператора возвращает значение *true*, то происходит очищение экрана и запуск пользовательской подпрограммы-функции *draw_gamefield(void)*, которая рисует на экране игровое поле.

В случае, если условие второго условного оператора возвращает значение *true*, то происходит очищение экрана и запуск пользовательской подпрограммы-функции *options(void)*.

В случае если условие третьего условного оператора возвращает значение *true*, то происходит выход из программы при помощи функции *exit()*.

Игровое поле представляет собой прямоугольник, на котором расположены разрушаемые и не разрушаемые блоки. Границы поля обозначены зеленым прямоугольником. Разрушаемые и не разрушаемые блоки расставляются при помощи цикла *while* и функции *bar(x1, y1, x2, y2)*. Целочисленное значение цвета разрушаемого блока равно 6, а не разрушаемого равно 8.

2.2. Переназначение управления

Программой предусмотрено стандартное управление для каждого игрока (табл. 2.1.)

Таблица 2.1.

Стандартное управление

Действие	Игрок №1	Игрок №2
Движение вверх	«W»	«I»
Движение вниз	«S»	«K»
Движение вправо	«D»	«L»
Движение влево	«A»	«J»
Установить/взорвать бомбу	«E»	«O»

В случае, если условие второго условного оператора из функции *menu* (*void*), описанной выше, возвращает значение *true*, то происходит очищение экрана и запуск пользовательской подпрограммы-функции *options* (*void*). При этом на экране появляется соответствующее сообщение (рис. 2.2).

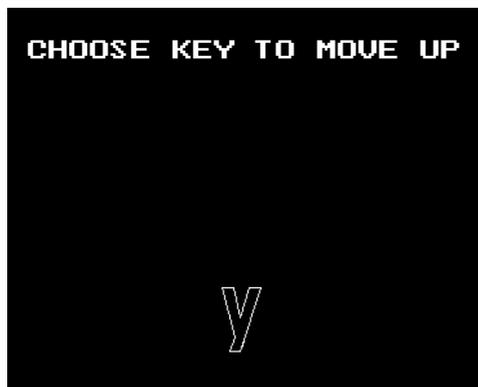


Рис. 2.2. Переназначение управления

Функция *options* (*void*) позволяет переназначить управление для второго игрока. Для первого игрока управление стандартное, и осуществляется клавишами «w», «s», «a», «d», «e».

Чтобы переназначить управление была использована функция *getch()*, которая ожидает нажатия клавиши и возвращает *ASCII*-код нажатой клавиши. Далее *ASCII*-код конвертируется в тип *char* и выводится на экран. Если переназначение управления произошло успешно, то на экране появляется соответствующая надпись (рис. 2.3).

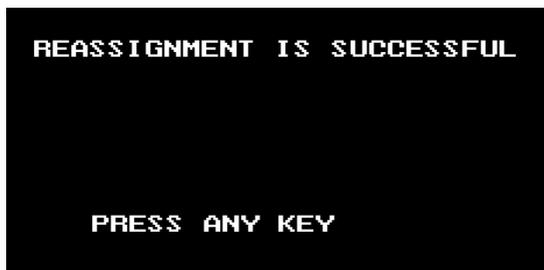


Рис. 2.3. Успешное переназначение управления

Все надписи выводятся в графическом режиме при помощи функции *outtextxy(x, y, textstring)*. При нажатии клавиши после успешного переназначения происходит запуск функции *menu(void)*, описанной выше.

2.3. Движение персонажей

Одним из главных аспектов игры является движение персонажей. В разработанном программном продукте движение персонажей на игровом экране осуществляется посредством четырех пользовательских подпрограмм-функций, осуществляющих движения в конкретном направлении: *move_up(x, y, color)*, *move_down(x, y, color)*, *move_left(x, y, color)*, *move_right(x, y, color)*, где *x* и *y* – координаты персонажа, а переменная *color* – цвет персонажа. Движение в конкретном направлении происходит по принципу: одно нажатие клавиши – один шаг. Длина одного шага составляет 35 пикселей.

Руководство движением в конкретном направлении осуществляется одной главной подпрограммой (*move(void)*), являющейся главной над предыдущими четырьмя. Она содержит в себе бесконечный цикл, при помощи которого и осуществляется игровой процесс. Выход из этого цикла происходит только тогда, когда уровень жизни какого-то персонажа (либо обоих одновременно) достигнет 0%.

Принцип осуществления движения в подпрограмме *move(void)* следующий: при помощи функции *kbhit()* захватывается клавиша с клавиатуры, затем если *ASCII*-код этой клавиши совпадает с кодом клавиши, отвечающей за движение, то осуществить движение в соответствующем направлении. Например, клавиша «*W*» отвечает за движение вверх, и если была нажата клавиша «*W*», то осуществить движение вверх (т.е. запустить функцию *move_up(x, y, color)*). Затем это действие повторяется в бесконечном цикле.

2.4. Разрушение объектов

Разрушение объектов, как и движение, является одним из ключевых моментов в игре. Для его реализации были написанные следующие подпрограммы-функции: *job_bomb(x, y, color)*,

$destroy_krest(x, y, color)$, $destroy(x, y, color)$, где x и y – координаты для установки/взрыва бомбы, а $color$ – цвет взрыва. Цвет первой функции означает цвет персонажа, а для второй и третьей принимает всего 2 значения: светло-красный и черный, причем первый для отображения взрыва, а второй для закраски.

Функция $job_bomb(x, y, color)$ вызывается из бесконечного цикла функции $move(void)$ в случае нажатия клавиши, отвечающей за установку/взрыв бомбы. В случае установки бомба ставится на координаты персонажа, а затем при смещении персонажа с места она прорисовывается. В случае если бомба установлена, при нажатии клавиши будет осуществляться взрыв посредством функций $destroy(x, y, color)$ и $destroy_krest(x, y, color)$. Таким образом, в игре осуществлена возможность ставить только одну бомбу и взрывать ее по нажатии клавиши.

Функция $destroy_krest(x, y, color)$ рисует крестик на одной клетке игрового поля. Функция $destroy(x, y, color)$ использует функцию $destroy_krest(x, y, color)$ для прорисовки взрыва либо закраски следов взрыва. Взрывы осуществляются в четырех направлениях (вверх, вниз, вправо, влево) до границ поля, либо до не разрушаемых блоков. Если на пути взрыва не встречается препятствий, то взрыв осуществляется до конца радиуса взрыва (рис. 2.4).

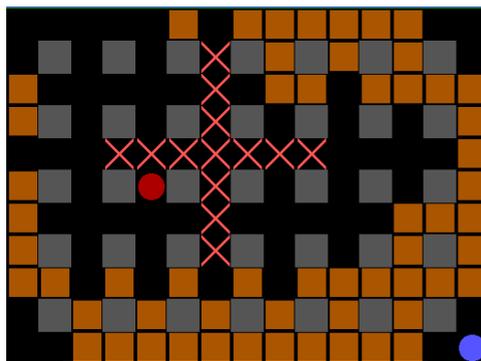


Рис. 2.4. Полный радиус взрыва

Если в радиусе взрыва оказывается разрушаемый блок, то он взрывается, причем после этого взрыв в данном направлении прекращается (рис. 2.5).

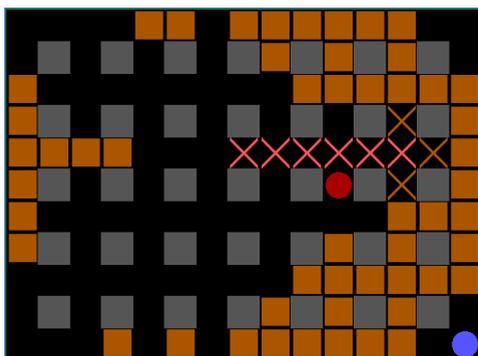


Рис. 2.5. Взрыв блока и прекращение взрыва

Если в радиусе взрыва оказался персонаж, то он закрашивается крестиком другого цвета (рис. 2.6), и его уровень жизни уменьшается на 50%, а затем прорисовывается заново, причем после этого взрыв в данном направлении прекращается.

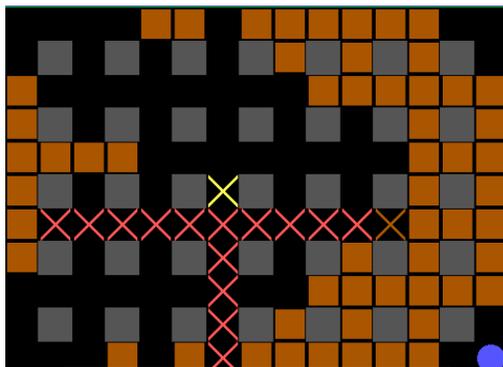


Рис. 2.6. Бомбер в радиусе взрыва

Если в радиусе взрыва оказался бонус, то он не взрывается, причем после этого взрыв в данном направлении прекращается.

В программе предусмотрены три варианта окончания игры:

1. Победа первого игрока.
2. Победа второго игрока.
3. Ничья.

Если уровень жизни какого-либо персонажа достиг 0%, то появляется одна из надписей, представленных на рисунке ниже (рис. 2.7).



Рис. 2.7. Вывод программы при победе: a – первого игрока; $б$ – второго игрока

Если уровень жизни обоих персонажей достиг 0%, то появляется соответствующее сообщение о проигрыше (рис. 2.8).



Рис. 2.8. Проигрыш обоих игроков

После окончания вывода результата игры программа ожидает ввода пользователя и затем завершает свою работу.

Сохранение и загрузка игры предусмотрена не была, поэтому для начала игры необходимо перезапустить программу.

2.5. Бонусная система

Во всех реализациях игры существовали бонусы. В данном программном продукте было решено создать два бонуса: бонус на увеличение здоровья XP (+50%) и бонус на увеличение радиуса взрыва (+1), обозначаемые соответствующими картинками (рис. 2.9).

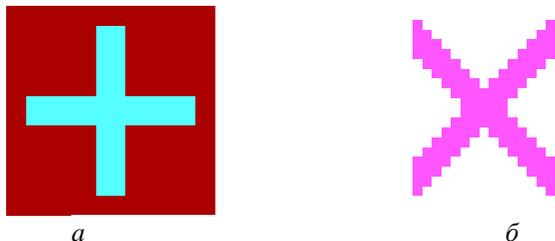


Рис. 2.9. Бонусы игры: a – увеличение здоровья; $б$ – радиуса взрыва

Данный функционал был реализован в пользовательской функции *draw_bonus(x, y)* (где x, y – координаты взорванного блока) при помощи оператора *switch(bonus)*. Значение переменной *bonus* задается рандомно (*rand%25*), затем при помощи вышеуказанного оператора выбирается действие для определенного значения. Если *rand%25* возвращает 0, то на игровом поле появляется бонус на увеличение радиуса взрыва, а если *rand%25* возвращает 1, то на игровом поле появляется бонус на увеличение *XP*. В остальных случаях ничего не происходит.

2.6. Помощь и динамическая статистика

Для упрощения освоения управления игрой была написана функция *help(void)*. Она выводит на экран под игровым полем клавиши для управления и подписи к ним, а справа от игрового поля прорисовывает таблицу, в которой будут размещены данные об игроках. Клавиши управления хранятся в структуре. Для наглядного отображения количества *XP* и радиуса взрыва для каждого игрока было решено создать динамическую статистику, которая изменяется сразу же после изменения каких-либо характеристик игрока. Для этого была написана пользовательская подпрограмма-функция *Dynamic_statistic(void)*, которая вызывается в функции *move(void)* на каждом шагу цикла. Принцип работы этой функции следующий: закрашиваются поля таблицы, прорисованной функцией *help(void)*, а затем в них записываются данные. Если данные не изменились, закрашка и прорисовка все равно сработают, но так как процесс выполнения программы достаточно быстрый, пользователь не заметит никаких изменений.

2.7. Анализ полученных результатов

В ходе тестирования созданного программного продукта было выявлено, что поставленная цель – создание программного продукта, реализующего игру «Бомберы» – и все задачи курсовой работы были достигнуты. Также при выполнении курсовой работы были реализованы дополнительные функции:

1. Инициализация графического режима посредством пользовательской подпрограммы-функции *InitGraph(void)*.

2. Функция *help(void)*, которая выводит под игровым полем информацию об управлении.

В результате тестирования программного продукта студентами не было получено жалоб об отклонениях в правилах игры по сравнению с общепринятыми стандартами, установленными многочисленными версиями игры «Бомберы». 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 «Бомберы» поставленная цель была достигнута.

При достижении поставленной цели были решены следующие задачи:

- изучены теоретические сведения, необходимые для решения данной задачи;
- систематизированы и обобщены полученные знания;
- изучены аналогичные программные продукты;
- создан программный продукт, реализующий игру «Бомберы»;
- проведено тестирование и отладка программы.

В ходе выполнения курсовой работы были изучена новая область, а именно работа с графическим драйвером в среде *DOS*.

В ходе тестирования созданного программного продукта были выявлены следующие недостатки:

1. Не всегда отображается бомба при одновременном нажатии клавиш у обоих игроков.
2. В игре нет звукового сопровождения.
3. Измененное управление при новом запуске игры вновь меняется на стандартное.
4. Скорость отображения взрывов, скорость переключения окон в переназначении управления, а также скорость отображения финального окна может меняться в зависимости от производительности компьютера, т.к. она установлена функцией *Delay*.

Подводя итоги, можно сказать, что при помощи разработанного программного продукта можно приятно провести время и совершенствовать координацию. Программа не занимает много места на жестком диске, несложна в освоении, а также не требовательна к установленному программному обеспечению.

СПИСОК ЛИТЕРАТУРЫ

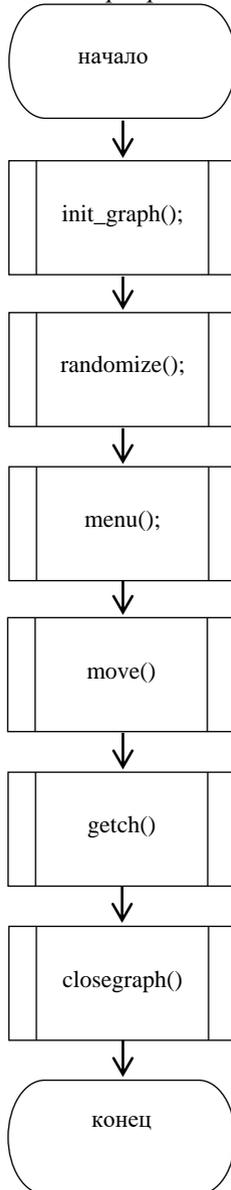
1. Подбельский, В.В. Язык C#. Базовый курс [Текст]. – М.: Финансы и статистика, 2011. – 384 с.
2. Страуструп, Б. Язык программирования C++: Пер. с англ. – 3-е спец. изд. [Текст]. – М.: Бином, 2003. – 1104 с.
3. Фролов, А.В. Визуальное проектирование приложений C# [Текст]. / А.В. Фролов, Г.В. Фролов. – М.: КУДИЦ-ОБРАЗ, 2003. – 512 с.
4. Эккель, Б. Философия C++. Введение в стандартный C++: Пер. с англ. – 2-е изд. [Текст]. – СПб.: Питер, 2004. – 572 с.
5. *Borland Graphics Interface (BGI) for Windows* [Электронный ресурс]. – Справочник по BGI. – Режим доступа: <http://www.cs.colorado.edu>.
6. *Cyberforum* [Электронный ресурс]. – Форум программистов и сисадминов. – Режим доступа: <http://www.cyberforum.ru>.
7. *CITforum* [Электронный ресурс]. – Справочник по работе с DOS. – Режим доступа: <http://citforum.ru>.
8. *DosBox* – эмуляция игр [Электронный ресурс]. – Описание эмуляции игр в *DosBox*. – Режим доступа: <http://games-history.ru>.
9. Алгоритмы работы с графикой [Электронный ресурс]. – Режим доступа: http://www.dmtsoft.ru/un/aticle_graphic_s.
10. Либерти, Джеймс. Освой C++ за 21 день [Электронный ресурс]. – Режим доступа: <http://royallib.com>.

ПРИЛОЖЕНИЯ

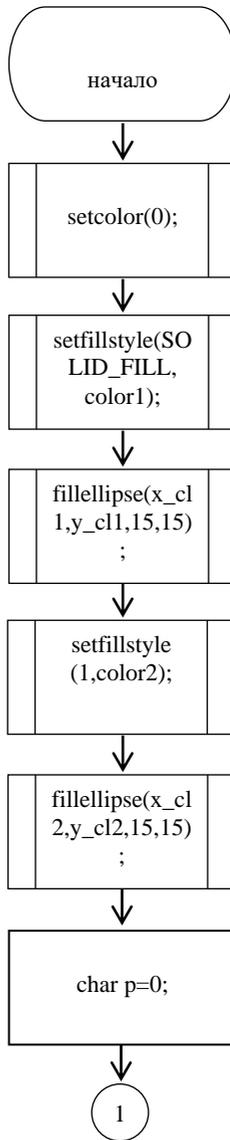
ПРИЛОЖЕНИЕ 1

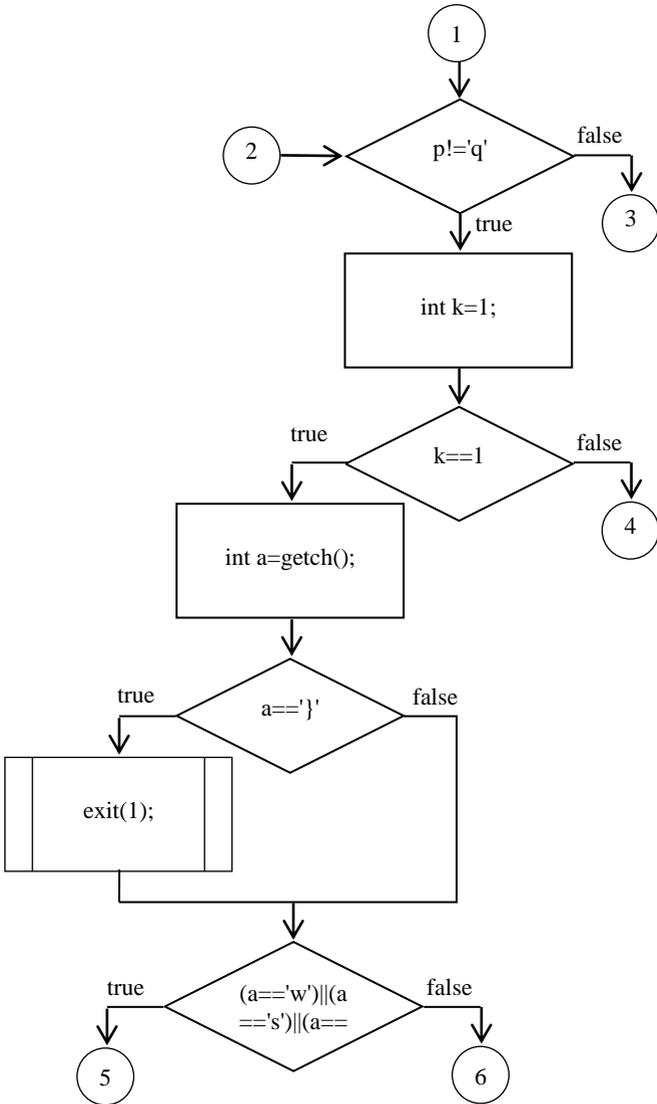
Алгоритм программного продукта

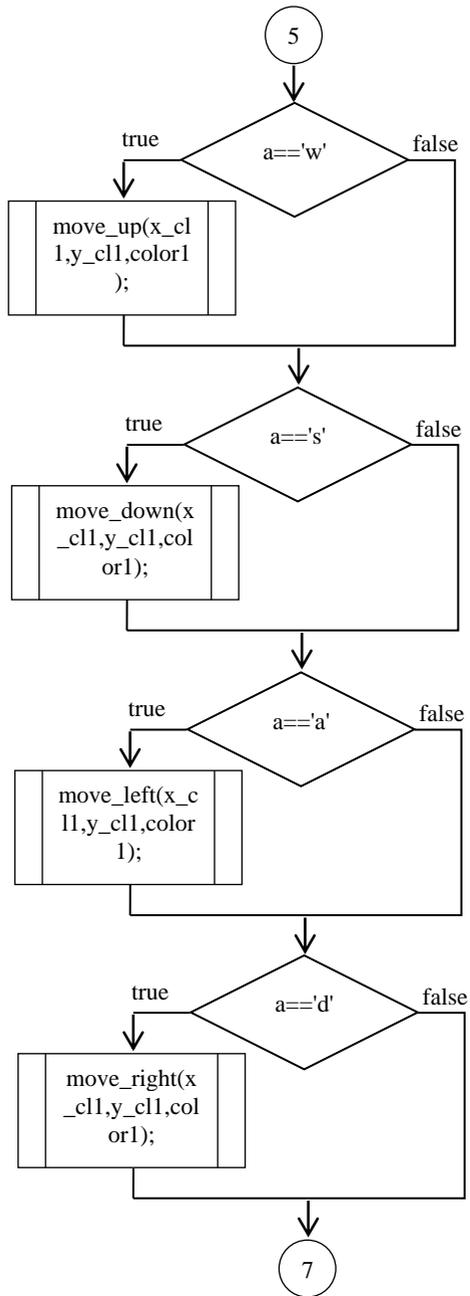
Основная программа

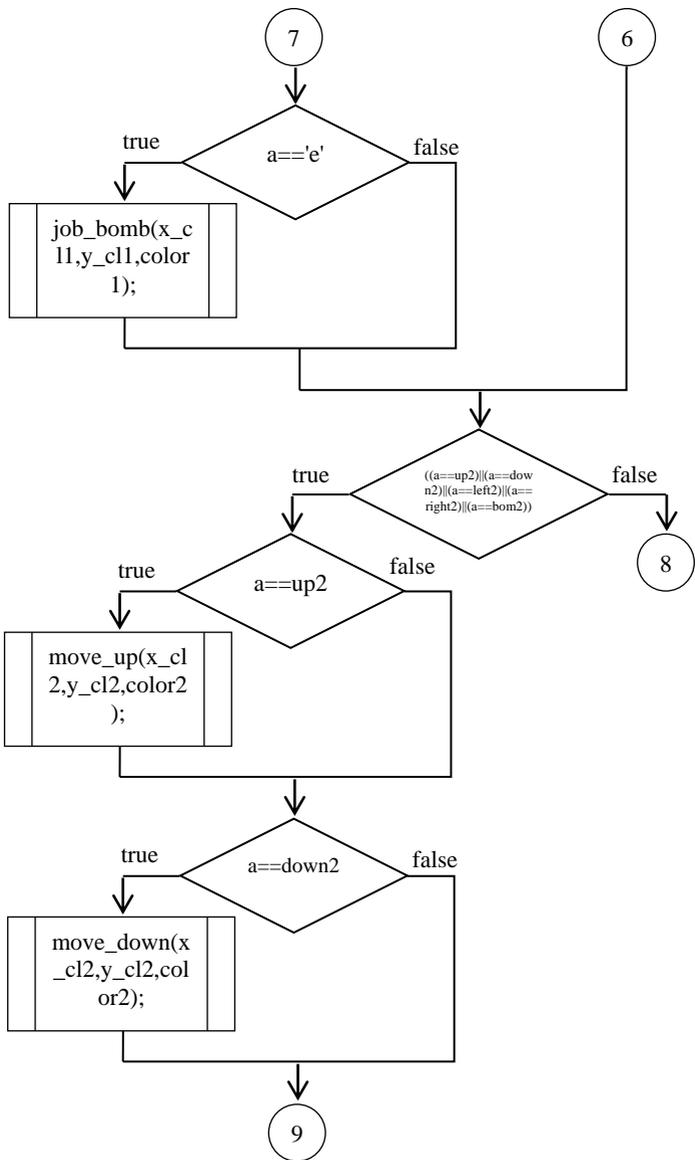


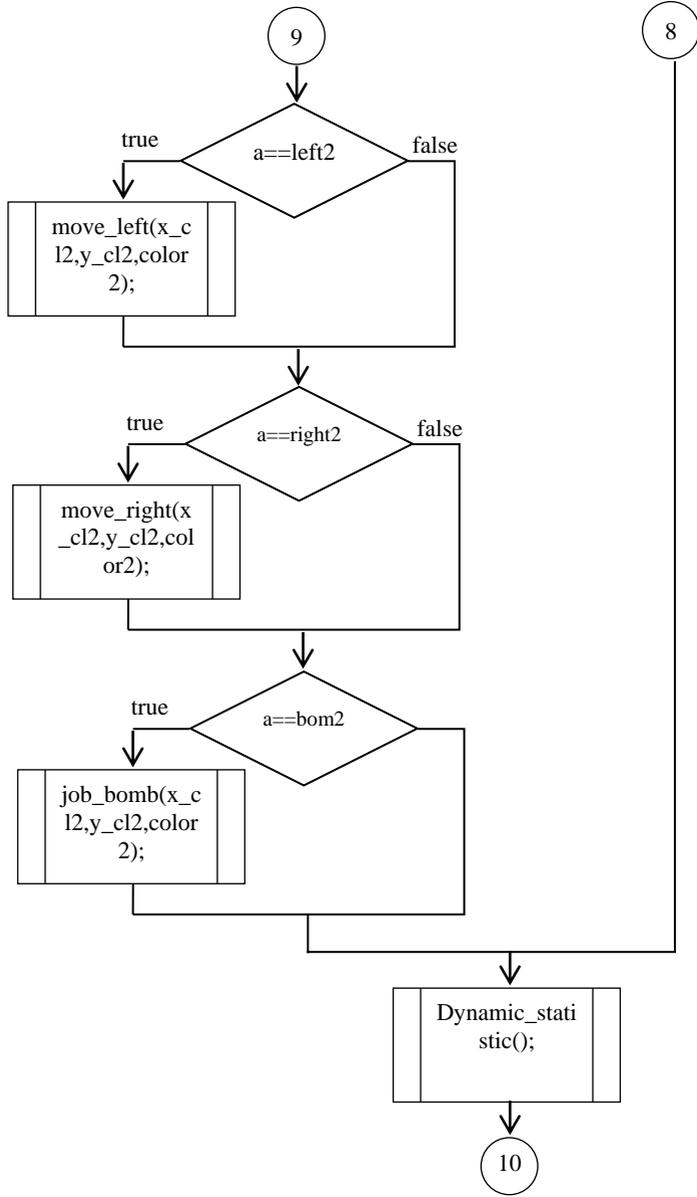
Функция Move()

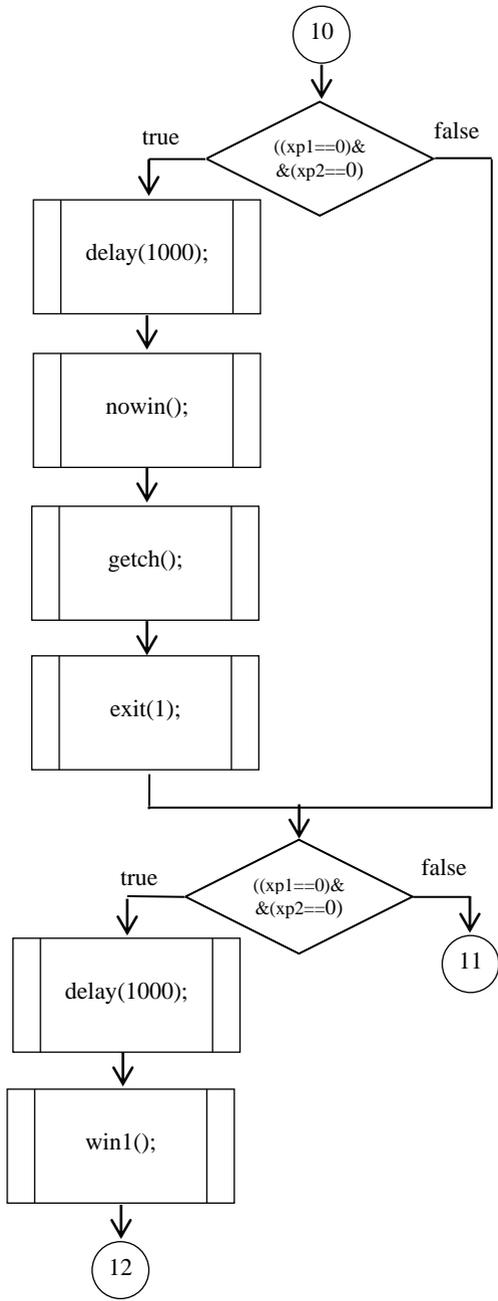


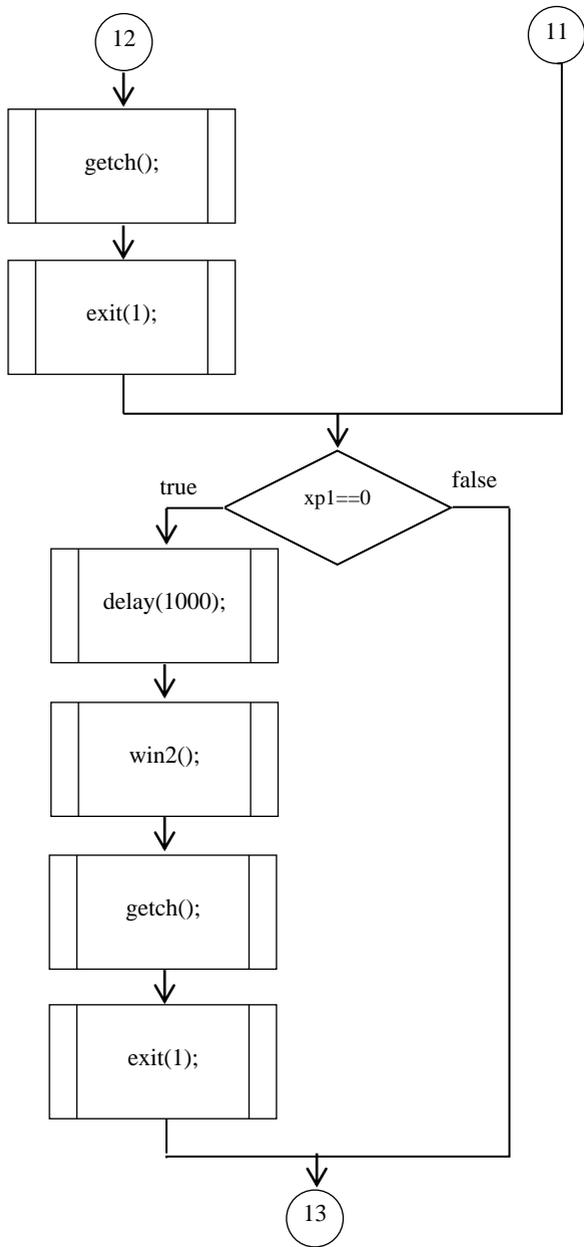


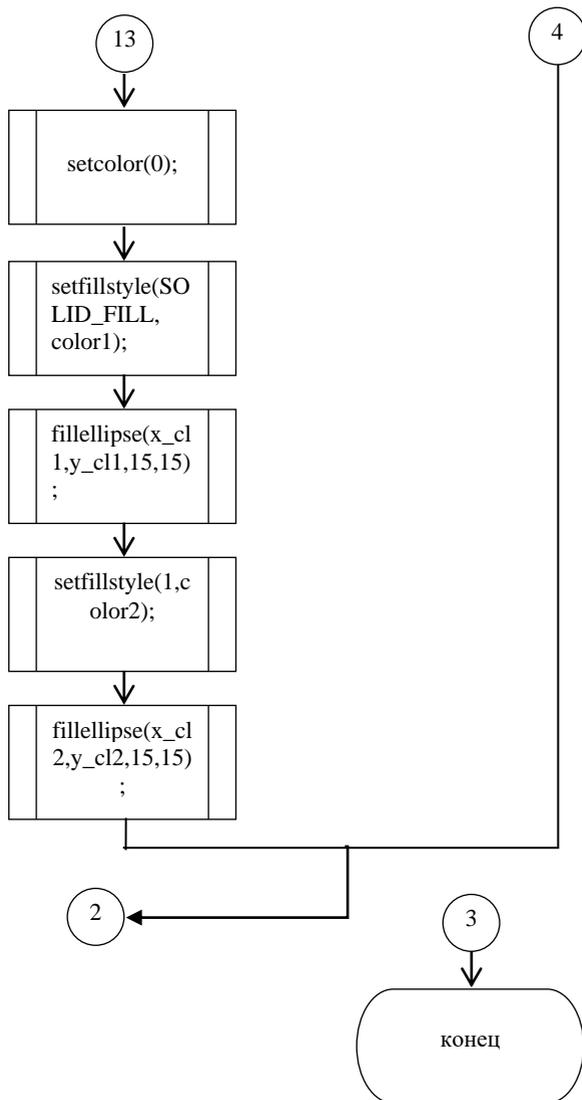




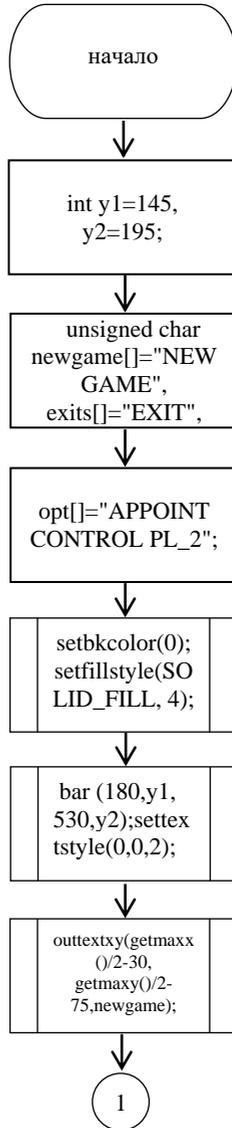


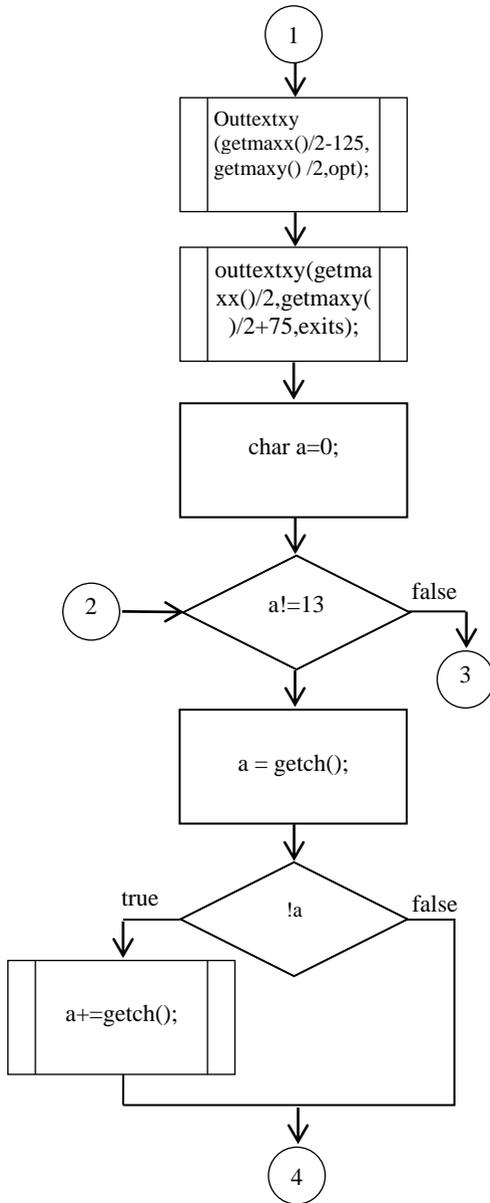


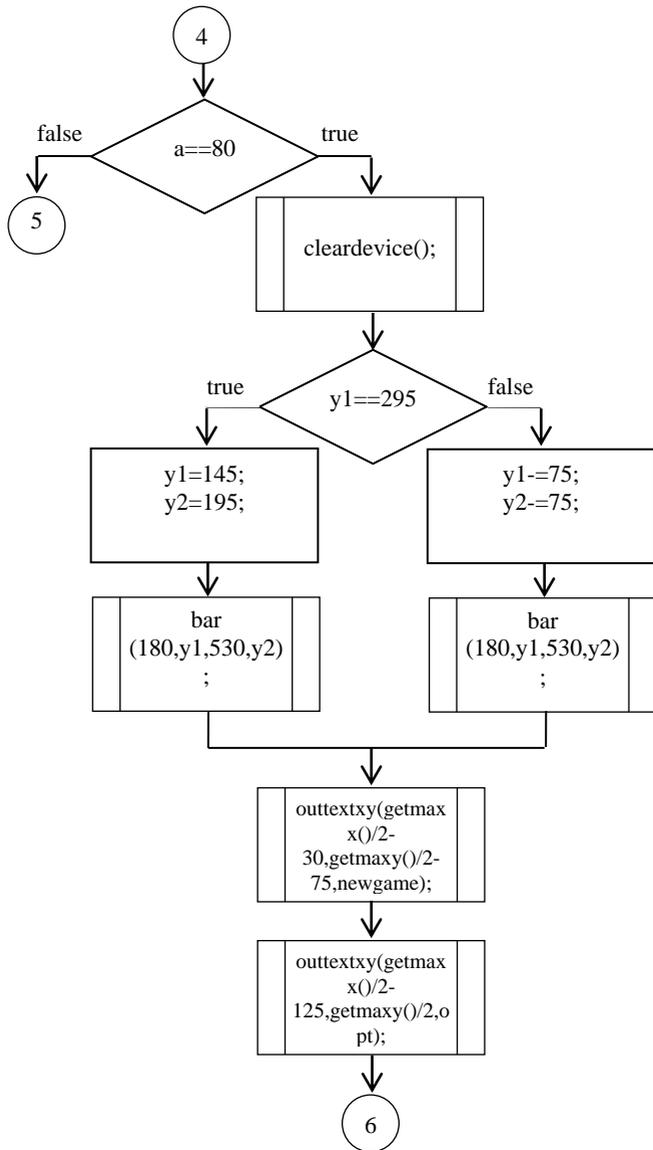


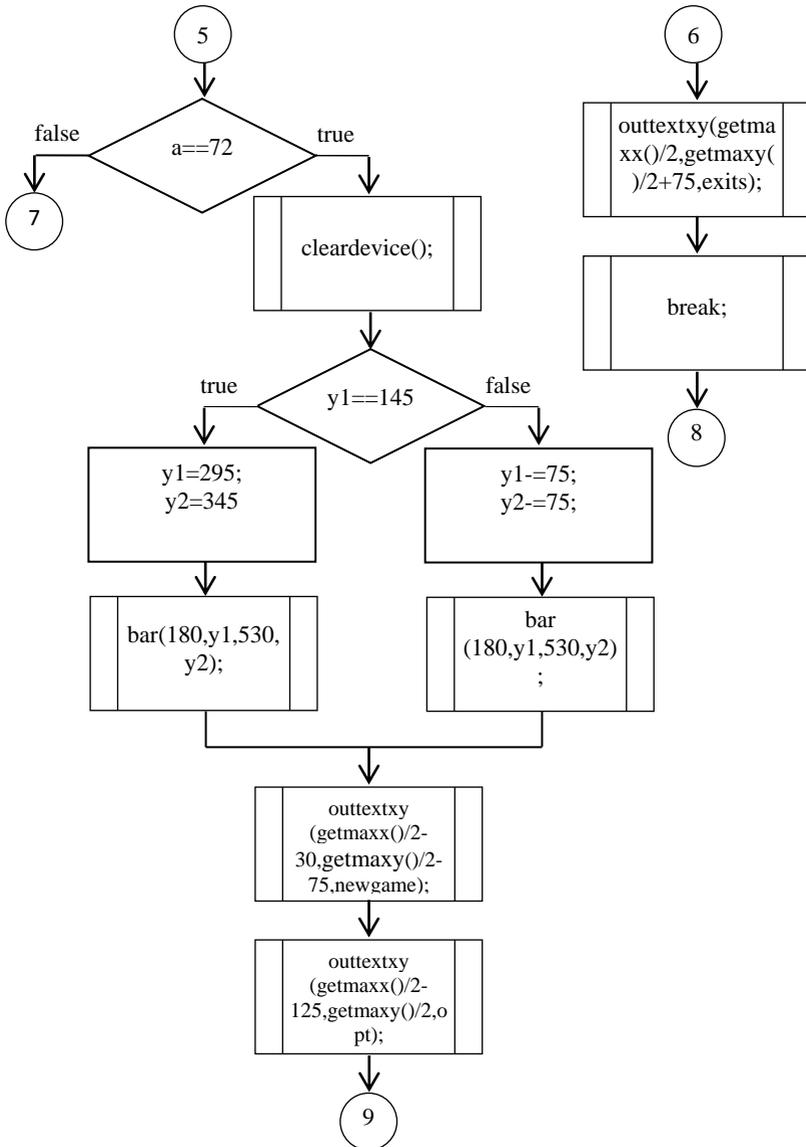


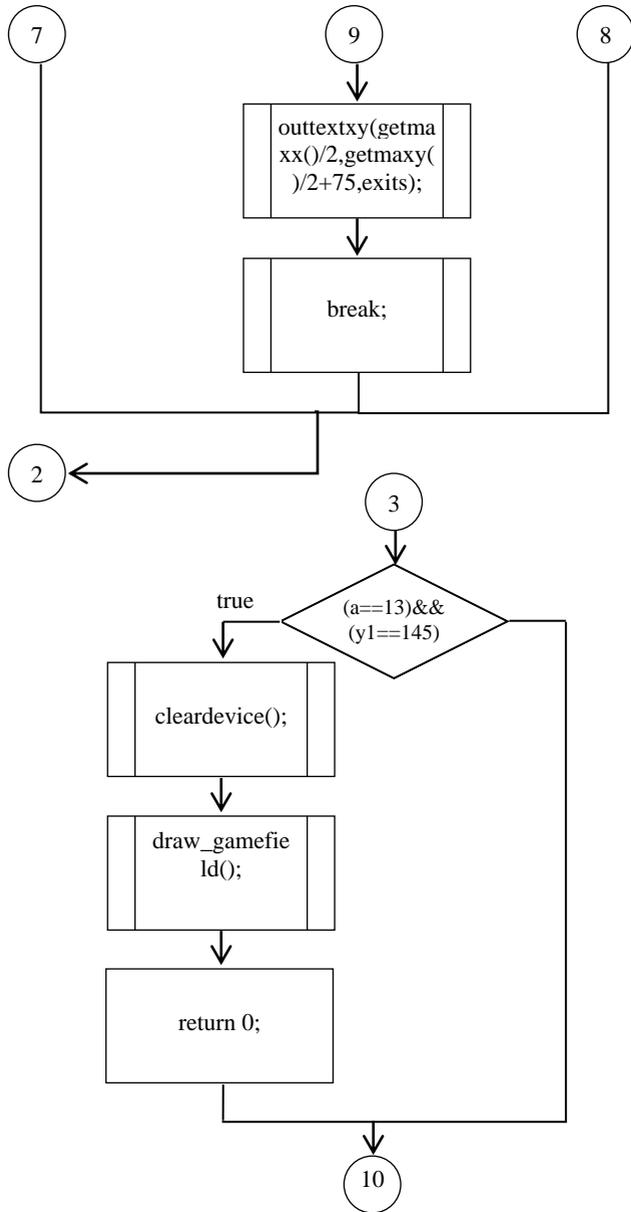
Функция Мени(void)











Листинг программного продукта

```

#include <iostream.h>
#include <dos.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <graphics.h>
#include <time.h>
#include <string.h>
//=====GLOBAL CONST=====|
int left2=106,right2=108,up2=105,down2=107,bom2=111;
struct help { //for help
    unsigned char *movie;
    unsigned char *kl;
};
help helps0[5],helps1[5];
int bx,by,bx1,by1,bx2,by2; //points of boom
int color1=4,color2=9; //player's colors
int x_cl1=55, y_cl1=50, x_cl2=545, y_cl2=400; //player's points
int up_long1=2, up_long2=2; //bonus
int boom1=2, boom2=2; //bonus
int make1=0, make2=0; //make boom
int xp1=2,xp2=2; //player's HP
//-----GRAPHICS-----|
void init_graph ();
int menu ();
int options ();
void draw_gamefield ();
void help();
void win1 ();
void win2 ();
void nowin();
//-----DVIJENIE-----|
int move_up (int x, int &y, int color);
int move_down (int x, int &y, int color);
int move_left (int &x, int y, int color);
int move_right (int &x, int y, int color);
void move ();
//-----BOMBS-----|

```

```

int bomb_up (int x, int &y, int color);
int bomb_down (int x, int &y, int color);
int bomb_left (int &x, int y, int color);
int bomb_right (int &x, int y, int color);
int job_bomb (int &x, int &y, int color);
//-----DESTROY-----|
void destroy_krest (int x, int y, int color);
void destroy (int x, int y, int up_long, int color);
//-----BONUS SYSTEM-----|-----|
void draw_bonus(int x, int y);
void Dynamic_statistic();
////////////////////////////////////
/*-----MAIN-----*/
void main ()
{
    init_graph();
    randomize();
    menu();
    move();
    getch();
    closegraph();
}
//-----MOVE-----
void move ()
{
    setcolor(0);
    setfillstyle(SOLID_FILL, color1);
    fillellipse(x_cl1,y_cl1,15,15);
    setfillstyle(1,color2);
    fillellipse(x_cl2,y_cl2,15,15);
    char p=0;
    while (p!='q')
    {
        int k=1;
        if (k==1) {
            int a=getch();
            if (a=='j') exit(1);

if ((a=='w')||(a=='s')||(a=='a')||(a=='d')||(a=='e'))
        {
            if (a=='w') move_up(x_cl1,y_cl1,color1);
            if (a=='s') move_down(x_cl1,y_cl1,color1);
            if (a=='a') move_left(x_cl1,y_cl1,color1);

```

```

        if (a=='d') move_right(x_cl1,y_cl1,color1);//player 1
        if (a=='e') job_bomb(x_cl1,y_cl1,color1);
    }
if ((a==up2)||((a==down2)||((a==left2)||((a==right2)||((a==bom2)))
{
    if (a==up2) move_up(x_cl2,y_cl2,color2);
    if (a==down2) move_down(x_cl2,y_cl2,color2);
    if (a==left2) move_left(x_cl2,y_cl2,color2);
    if (a==right2) move_right(x_cl2,y_cl2,color2);//player 2
    if (a==bom2) job_bomb(x_cl2,y_cl2,color2);
}
Dynamic_statistic();
if ((xp1==0)&&(xp2==0)) {delay(1000); nowin(); getch(); exit(1);}
if (xp2==0) {delay(1000); win1(); getch(); exit(1);}
if (xp1==0) {delay(1000); win2(); getch(); exit(1);}
setcolor(0);
setfillstyle(SOLID_FILL, color1);
fillellipse(x_cl1,y_cl1,15,15);
setfillstyle(1,color2);
fillellipse(x_cl2,y_cl2,15,15);
    }
}
}
/*-----JOB BOMB-----*/
int job_bomb(int &x, int &y, int color)
{
    bx=x; by=y;
    if ((make1>boom1)||((make2>boom2)) return 0;
    if ((make1!=0)&&(color==4))
        { destroy_krest(bx1,by1,12);
        destroy(bx1,by1,up_long1,12);
        delay(150);
        destroy_krest(bx1,by1,0);
        destroy(bx1,by1,up_long1,0);
        make1=0;
        return 0;
        }
    if ((make2!=0)&&(color==9))
        { destroy_krest(bx2,by2,12);
        destroy(bx2,by2,up_long2,12);
        delay(150);
        destroy_krest(bx2,by2,0);
        destroy(bx2,by2,up_long2,0);

```

```

        make2=0;
        return 0;
    }
while (1)
{
int a=getch();
if (color==4) {
    bx1=x; by1=y;
    if (a=='w') bomb_up(x,y,color);
    if (a=='s') bomb_down(x,y,color);
    if (a=='a') bomb_left(x,y,color);
    if (a=='d') bomb_right(x,y,color);
    make1++;
}
if (color==9) {
    bx2=x; by2=y;
    if (a==up2) bomb_up(x,y,color);
    if (a==down2) bomb_down(x,y,color);
    if (a==left2) bomb_left(x,y,color);
    if (a==right2) bomb_right(x,y,color);
    make2++;
}

break;
}
return 0;
}
/*-----NARISOVAT BONUS-----*/
void draw_bonus(int x, int y)
{
int newx=x, newy=y, bonus=rand()%25;
int a=x-12, b=y-12, c=x+12, d=y+12;
setcolor(2);
switch (bonus) {
    case 0: { //-----LONG-----
        setfillstyle(1,-1);
        bar(a+2,b+2,c-2,d-2);
        setcolor(13);
        for (int i=0; i<4; i++)
            line(a+5,b+3+i,c-5,d-7+i);

        for (i=0; i<4; i++)
            line(c-5,b+3+i,a+5,d-7+i);
        break;
    }
}
}

```

```

        }
        case 1: {          //-----HP-----
            setfillstyle(1,4);
            bar(a+2,b+2,c-2,d-2);
            setfillstyle(1,11);
            bar(newx-1,newy-8,newx+1,newy+8);
            bar(newx-8,newy-1,newx+8,newy+1);
            break;
        }
        default: break;
    }
}
////////////////////////////////////
//-----INTSIALIZATSIA GRAPH-----
-----
void init_graph()
{
    int gdriver, gmode, errorcode;
    gdriver=DETECT;
    initgraph(&gdriver, &gmode, "C://TC//BGI");
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf ("Graphics error:&s\n", grapherrormsg(errorcode));
        printf ("Press any key to halt:");
        getch();
        exit(1);
    }
}
//-----MENU-----
int menu(void)
{
    int y1=145, y2=195;
    unsigned char newgame[]="NEW GAME", exits[]="EXIT",
opt[]="APPOINT CONTROL PL_2";
    setbkcolor(0);
    setfillstyle(SOLID_FILL, 4);
    bar (180,y1, 530,y2);
    settextstyle(0,0,2);
    outtextxy(getmaxx()/2-30,getmaxy()/2-75,newgame);
    outtextxy(getmaxx()/2-125,getmaxy()/2,opt);
    outtextxy(getmaxx()/2,getmaxy()/2+75,exits);
}

```

```

char a=0;
while(a!=13){
a = getch();
if(!a) a+=getch();
    switch(a){
        case 80:{
                cleardevice();
                if (y1==295) {y1=145; y2=195;
bar(180,y1,530,y2);}
                else {y1+=75; y2+=75; bar (180,y1, 530,y2);}
                outtextxy(getmaxx()/2-30,getmaxy()/2-75,newgame);
                outtextxy(getmaxx()/2-125,getmaxy()/2,opt);
                outtextxy(getmaxx()/2,getmaxy()/2+75,exits);
                break;
                }//down
        case 72:{
                cleardevice();
                if (y1==145) {y1=295; y2=345;
bar(180,y1,530,y2);}
                else {y1-=75; y2-=75; bar (180,y1, 530,y2);}
                outtextxy(getmaxx()/2-30,getmaxy()/2-75,newgame);
                outtextxy(getmaxx()/2-125,getmaxy()/2,opt);
                outtextxy(getmaxx()/2,getmaxy()/2+75,exits);
                break;} //up
                }
    }
if ((a==13)&&(y1==145)) {cleardevice(); draw_gamefield(); return
0;}
if ((a==13)&&(y1==220)) {cleardevice(); options(); return 0;}
if ((a==13)&&(y1==295)) exit(1);
return 0;
}
/*-----OPTIONS-----*/
int options ()
{
cleardevice();
settextstyle(0,0,2);
outtextxy(150,100,"CHOOSE KEY TO MOVE UP");
up2=getch();
settextstyle(10,0,5);
outtextxy(300,250,(const char far*)&up2);
delay(1000);
settextstyle(0,0,2);

```

```

cleardevice();
outtextxy(150,100, "CHOOSE KEY TO MOVE DOWN");
down2=getch();
setttextstyle(10,0,5);
outtextxy(300,250,(const char far*)&down2);
delay(1000);
setttextstyle(0,0,2);
cleardevice();
outtextxy(150,100, "CHOOSE KEY TO MOVE LEFT");
left2=getch();
setttextstyle(10,0,5);
outtextxy(300,250,(const char far*)&left2);
delay(1000);
setttextstyle(0,0,2);
cleardevice();
outtextxy(150,100, "CHOOSE KEY TO MOVE RIGHT");
right2=getch();
setttextstyle(10,0,5);
outtextxy(300,250,(const char far*)&right2);
delay(1000);

setttextstyle(0,0,2);
cleardevice();
outtextxy(150,100, "CHOOSE KEY TO DESTROY BOMB");
bom2=getch();
setttextstyle(10,0,5);
outtextxy(300,250,(const char far*)&bom2);
delay(1000);

```

```

cleardevice();
setttextstyle(0,0,2);
outtextxy(150,150, "REASSIGNMENT IS SUCCESSFUL");
outtextxy(200,300, "PRESS ANY KEY");
getch();
cleardevice();
menu();
return 0;
}

```

```

//-----PRORISOVKA IGROVOGO POLIA-----

```

```

void draw_gamefield()
{
double x=37, y=32, rise=35, j=1, i=2;

```

```

int m=1, n=2;
/*-----PRORISOVKA GRANITS POLIA-----*/
setcolor(GREEN);
rectangle(36,31,getmaxx()-77,getmaxy()-61);
setfillstyle(SOLID_FILL, 9);
bar(0,30,getmaxx(),0);
bar(0,getmaxy(),35,0);
bar(getmaxx()-76,0,getmaxx(),getmaxy());
bar(getmaxx(),getmaxy(),0,getmaxy()-60);
/*-----RASSTANOVKA NERAZRUSHAEMIH BLOCOV-----*/
setfillstyle(1,8);
while (x+n*rise<getmaxx()-100)
{
    bar(x+m*rise,y+rise, x+n*rise,y+2*rise);
    x=37; y=32; j=1; i=2;
    while (y+j*rise<getmaxy()-100)
    {
        bar(x+m*rise,y+i*rise, x+n*rise,y+j*rise);
        i+=2; j+=2;
    }
    m+=2; n+=2;
}
/*-----RASSTANOVKA RAZRUSHAEMIH BLOCKOV-----*/
setfillstyle(1,6);
rise=30; int step=35;
x+=3; y+=2; i=2; j=0;
while (y+rise+j*step<getmaxy()-50) //---Prorisovka vertikalno---//
{
    while (x+rise+i*step<getmaxx()-100)
    {
        bar(x+i*step,y+j*step, x+rise+i*step,y+rise+j*step);
        i+=2;
    }
    j++; i=2;
}
i=3; j=0;
while (i<8)
{
    if(kbhit()) {int a=getch(); if(a=='q') exit(1);}
    bar(x,y+i*step,x+rise,y+rise+i*step);
    i+=2;
}
i=3; j=0;

```

```

while (i<8)
{
    bar(x+14*step,y+i*step,x+rise+14*step,y+rise+i*step);
    i+=2;
}
i=2;j=0;
while (x+rise+j*step<getmaxx()-50) //---Prorisovka gorizontalno---//
{
    if(kbhit()) {int a=getch(); if(a=='q') exit(1);}
    while (y+rise+i*step<getmaxy()-100)
    {
        bar(x+j*step,y+i*step, x+rise+j*step,y+rise+i*step);
        i+=2;
    }
    j++; i=2;
}
i=3; j=0;
while (i<=11)
{
    bar(x+i*step,y,x+rise+i*step,y+rise);
    i+=2;
}
i=3;
while (i<=11)
{
    bar(x+i*step,y+10*step,x+rise+i*step,y+rise+10*step);
    i+=2;
}
help();
}
/*-----HELP-----*/
void help()
{
    helps0[0].movie="    up    - 'w'";
    helps1[0].movie="    up    - ' '";
    helps0[1].movie="    down  - 's'";
    helps1[1].movie="    down  - ' '";
    helps0[2].movie="    left  - 'a'";
    helps1[2].movie="    left  - ' '";
    helps0[3].movie="    right - 'd'";
    helps1[3].movie="    right - ' '";
    helps0[4].movie="plant(destroy) bomb - 'e'";
    helps1[4].movie="plant(destroy) bomb - ' '";
}

```

```

setcolor(4);
settextstyle(0,0,1);
outtextxy(10,420,"Player 1:");      outtextxy(350,420,"Player 2:");
outtextxy(55,420,helps0[0].movie);
outtextxy(425,420,helps1[0].movie);
outtextxy(55,430,helps0[1].movie);
outtextxy(425,430,helps1[1].movie);
outtextxy(55,440,helps0[2].movie);
outtextxy(425,440,helps1[2].movie);
outtextxy(55,450,helps0[3].movie);
outtextxy(425,450,helps1[3].movie);
outtextxy(55,460,helps0[4].movie);
outtextxy(425,460,helps1[4].movie);

outtextxy(615,420, (const char far*)&up2);
outtextxy(615,430, (const char far*)&down2);
outtextxy(615,440, (const char far*)&left2);
outtextxy(615,450, (const char far*)&right2);
outtextxy(615,460, (const char far*)&bom2);

setcolor(15);
rectangle(570,65,637,400);
line(570,240,637,240);
setcolor(4);
outtextxy(573,120, "HP:");  outtextxy(573,295, "HP:");
outtextxy(573,162, "LONG:");  outtextxy(573,337, "LONG:");

setcolor(0);
rectangle(573,130,633,155);  rectangle(573,305,633,330);
rectangle(573,172,633,197);  rectangle(573,347,633,372);
setcolor(4);
outtextxy(573,68, "PLAYER_1"); outtextxy(573,243, "PLAYER_2");
if (xp1==2) outtextxy(585,140, "100%");  if (xp2==2)
outtextxy(585,315, "100%");
if (up_long1==2) outtextxy(600,182, "1");  if (up_long2==2)
outtextxy(600,357, "1");
}
/*-----DYNAMIC STATISTIC-----*/
void Dynamic_statistic ()
{
setcolor(4);
setfillstyle(1,9);

```

```

        bar(574,131,632,154);   bar(574,306,632,329);
        bar(574,173,632,196);   bar(574,348,632,371);
        if (up_long1==2) outtextxy(600,182, "1"); if (up_long2==2)
outtextxy(600,357, "1");
        if (up_long1==3) outtextxy(600,182, "2"); if (up_long2==3)
outtextxy(600,357, "2");
        if (up_long1==4) outtextxy(600,182, "3"); if (up_long2==4)
outtextxy(600,357, "3");
        if (up_long1==5) outtextxy(590,182, "4"); if (up_long2==5)
outtextxy(590,357, "4");
        if (up_long1==6) outtextxy(590,182, "5"); if (up_long2==6)
outtextxy(590,357, "5");
        if (up_long1==7) outtextxy(590,182, "MAX"); if (up_long2==7)
outtextxy(590,357, "MAX");

```

```

        if (xp1==4) xp1=3;           if (xp2==4) xp2=3;
        if (xp1==3) outtextxy(585,140, "150%"); if (xp2==3)
outtextxy(585,315, "150%");
        if (xp1==2) outtextxy(585,140, "100%"); if (xp2==2)
outtextxy(585,315, "100%");
        if (xp1==1) outtextxy(585,140, "50%"); if (xp2==1)
outtextxy(585,315, "50%");
        if (xp1==0) outtextxy(585,140, "0%"); if (xp2==0)
outtextxy(585,315, "0%");
    }

```

/*-----WIN-----*/

```
void win1 ()
```

```

{
    unsigned char gm[]="Game over";
    unsigned char pl1[]="Player_1 IS WINNER";
    cleardevice();
    setcolor(-1);
    settextstyle(1,0,4);
    outtextxy(getmaxx()/2-60,getmaxy()/2-50,gm);
    settextstyle(7,0,4);
    outtextxy(getmaxx()/2-150,getmaxy()/2,pl1);
}

```

```
void win2 ()
```

```

{
    unsigned char gm[]="Game over";
    unsigned char pl2[]="Player_2 IS WINNER";
    cleardevice();
    setcolor(-1);

```

```

    settextstyle(1,0,4);
    outtextxy(getmaxx()/2-60,getmaxy()/2-50,gm);
    settextstyle(7,0,4);
    outtextxy(getmaxx()/2-150,getmaxy()/2,pl2);
}
void nowin ()
{
    unsigned char gm[]="Game over";
    unsigned char pl2[]="TWO PLAYERS ARE LOSERS";
    cleardevice();
    setcolor(-1);
    settextstyle(1,0,4);
    outtextxy(getmaxx()/2-60,getmaxy()/2-50,gm);
    settextstyle(7,0,4);
    outtextxy(getmaxx()/2-200,getmaxy()/2,pl2);
}
/*-----DESTROY-----*/
void destroy (int x, int y, int up_long, int color)
{
    int rise=35, newx=x, newy=y, PointPixelColor;
    for (int i=1; i<up_long; i++) //krest up
    {
        if ((newy-35<31)||((getpixel(newx,newy-35)==8)) break;
        if ((getpixel(newx,newy-35)==13)||((getpixel(newx,newy-
35)==15)||((getpixel(newx,newy-35)==11)) break;
        if ((getpixel(newx,newy-35)==4)||((getpixel(newx,newy-
35)==9)||((getpixel(newx,newy-35)==14))
            { if (getpixel(newx,newy-35)==4) xp1--;
              if (getpixel(newx,newy-35)==9) xp2--;
              if (color!=0) destroy_krest(x,y-i*rise,14);
              else destroy_krest(x,y-i*rise,0);
              break;
            }
        if ((getpixel(newx,newy-35)==6))
            { destroy_krest(x,y-i*rise,6);
              if (color==0)
                  {
                      destroy_krest(x,y-i*rise,0);
                      draw_bonus(newx, newy-35);
                  }
            }
        break;
    }
    if (getpixel(newx,newy-35)==0) {

```

```

destroy_krest(x,y-i*rise,color);
newy-=35;
continue;
}
PointPixelColor=getpixel(newx+5,newy-70);
destroy_krest(x,y-i*rise,color);
if (PointPixelColor!=6) destroy_krest(x,y-i*rise,color);
else break;
newy-=35;
}
newy=y;
for (i=1; i<up_long; i++) //krest down
{
if ((newy+35>getmaxy()-61)||getpixel(newx,newy+35)==8) break;
if
((getpixel(newx,newy+35)==13)||getpixel(newx,newy+35)==15)||getpixel(newx,newy+35)==11) break;
if
((getpixel(newx,newy+35)==4)||getpixel(newx,newy+35)==9)||getpixel(newx,newy+35)==14)
{ if (getpixel(newx,newy+35)==4) xp1--;
if (getpixel(newx,newy+35)==9) xp2--;
if (color!=0)
destroy_krest(x,y+i*rise,14);
else destroy_krest(x,y+i*rise,0);
break;
}
if ((getpixel(newx,newy+35)==6))
{ destroy_krest(x,y+i*rise,6);
if (color==0)
{
destroy_krest(x,y+i*rise,0);
draw_bonus(newx, newy+35);
}
break;
}
if (getpixel(newx,newy+35)==0) {
destroy_krest(x,y+i*rise,color);
newy+=35;
continue;
}
PointPixelColor=getpixel(newx+5,newy+70);
destroy_krest(x,y+i*rise,color);

```

```

if (PointPixelColor!=6) destroy_krest(x,y+i*rise,color);
else break;
newy+=35;
}
newy=y;
for (i=1; i<up_long; i++) //krest left
{
if ((newx-35<36)||getpixel(newx-35,newy)==8) break;
if ((getpixel(newx-35,newy)==13)||getpixel(newx-
35,newy)==15)||getpixel(newx-35,newy)==11) break;
if ((getpixel(newx-35,newy)==4)||getpixel(newx-
35,newy)==9)||getpixel(newx-35,newy)==14)
    { if (getpixel(newx-35,newy)==4) xp1--;
      if (getpixel(newx-35,newy)==9) xp2--;
      if (color!=0) destroy_krest(x-i*rise,y,14);
      else destroy_krest(x-i*rise,y,0);
      break;
    }
if ((getpixel(newx-35,newy)==6))
    { destroy_krest(x-i*rise,y,6);
      if (color==0)
        {
          destroy_krest(x-i*rise,y,0);
          draw_bonus(newx-35, newy);
        }
      break;
    }
if (getpixel(newx-35,newy)==0) {
    destroy_krest(x-i*rise,y,color);
    newx-=35;
    continue;
}
PointPixelColor=getpixel(newx-70,newy+5);
destroy_krest(x-i*rise,y,color);
if (PointPixelColor!=6) destroy_krest(x-i*rise,y,color);
else break;
newx-=35;
}
newx=x;
for (i=1; i<up_long; i++) //krest right
{
if ((newx+35>getmaxx()-77)||getpixel(newx+35,newy)==8) break;

```

```

        if
((getpixel(newx+35,newy)==13)||getpixel(newx+35,newy)==15)||getpixel(newx+35,newy)==11)) break;
        if
((getpixel(newx+35,newy)==4)||getpixel(newx+35,newy)==9)||getpixel(newx+35,newy)==14))
                { if (getpixel(newx+35,newy)==4) xp1--;
                  if (getpixel(newx+35,newy)==9) xp2--;
                  if (color!=0)
destroy_krest(x+i*rise,y,14);
                else destroy_krest(x+i*rise,y,0);
                break;
                }

        if ((getpixel(newx+35,newy)==6))
                { destroy_krest(x+i*rise,y,6);
                  if (color==0)
                        {
                                destroy_krest(x+i*rise,y,0);
                                draw_bonus(newx+35, newy);
                        }
                  break;
                }
        if (getpixel(newx+35,newy)==0) {
                destroy_krest(x+i*rise,y,color);
                newx+=35;
                continue;
        }
        PointPixelColor=getpixel(newx+70,newy+5);
        destroy_krest(x+i*rise,y,color);
        if (PointPixelColor!=6) destroy_krest(x+i*rise,y,color);
        else break;
        newx+=35;
    }
    newx=x;
}
/*-----VZORVAT BOMBU-----*/
void destroy_krest (int x, int y, int color)
{
    if ((getpixel(x,y)==4)&&(color!=14)) { xp1--; color=14;}
    if ((getpixel(x,y)==9)&&(color!=14)) { xp2--; color=14;}
    int a=x-15, b=y-15, c=x+15, d=y+15;
    setfillstyle(1,0);
}

```

```

    bar(a,b-1,c,d);
    setcolor(color);
    for (int i=0; i<4; i++)
        line(a,b-2+i,c,d-2+i);
    for (i=0; i<4; i++)
        line(c,b-2+i,a,d-2+i);
}
//-----BOMBS UP DOWN LEFT RIGHT-----
-----
/*-----UP-----*/
int bomb_up (int x, int &y, int color)
{
    if (y-35<31) return 0;
    if ((getpixel(x,y-35)==6)||((getpixel(x,y-
35)==8)||((getpixel(x,y+35)==4)||((getpixel(x,y-35)==7)||((getpixel(x,y-35)==9))
return 0;
    setcolor(0);
    setfillstyle(1,0);
    fillellipse(x,y,15,15);
    setfillstyle(1,7);
    fillellipse(x,y,12,12);
    y-=35;
    setfillstyle(SOLID_FILL, color);
    fillellipse(x,y,15,15);
return 0;
}
/*-----DOWN-----*/
int bomb_down(int x, int &y, int color)
{
    if (y+35>getmaxy()-61) return 0;
    if
((getpixel(x,y+35)==6)||((getpixel(x,y+35)==9)||((getpixel(x,y+35)==4)||((getpixe
l(x,y+35)==8)||((getpixel(x,y+35)==7)) return 0;
    setcolor(0);
    setfillstyle(1,0);
    fillellipse(x,y,15,15);
    setfillstyle(1,7);
    fillellipse(x,y,12,12);
    y+=35;
    setfillstyle(SOLID_FILL, color);
    fillellipse(x,y,15,15);
return 0;
}

```

```

/*-----LEFT-----*/
int bomb_left(int &x, int y, int color)
{
    if (x-35<36) return 0;
    if ((getpixel(x-35,y)==6)||((getpixel(x-35,y)==8)||((getpixel(x-
35,y)==4)||((getpixel(x-35,y)==7)||((getpixel(x-35,y)==9)) return 0;
    setcolor(0);
    setfillstyle(1,0);
    fillellipse(x,y,15,15);
    setfillstyle(1,7);
    fillellipse(x,y,12,12);
    x-=35;
    setfillstyle(SOLID_FILL, color);
    fillellipse(x,y,15,15);
return 0;
}
/*-----RIGHT-----*/
int bomb_right(int &x, int y, int color)
{
    if (x+35>getmaxx()-77) return 0;
    if
((getpixel(x+35,y)==6)||((getpixel(x+35,y)==8)||((getpixel(x+35,y)==4)||((getpixe
l(x+35,y)==9)||((getpixel(x+35,y)==7)) return 0;
    setcolor(0);
    setfillstyle(1,0);
    fillellipse(x,y,15,15);
    setfillstyle(1,7);
    fillellipse(x,y,12,12);
    x+=35;
    setfillstyle(SOLID_FILL, color);
    fillellipse(x,y,15,15);
return 0;
}
//-----MOVE UP DOWN LEFT RIGHT-----
---
/*-----UP-----*/
int move_up (int x, int &y, int color)
{
    if (y-35<31) return 0;
    if ((getpixel(x,y-35)==6)||((getpixel(x,y-35)==8)||((getpixel(x,y-
35)==4)||((getpixel(x,y-35)==9)||((getpixel(x,y-35)==7)) return 0;
    if ((getpixel(x,y-35)==13)&&(color==4)&&(up_long1<7))
up_long1++;

```

```

        if ((getpixel(x,y-35)==13)&&(color==9)&&(up_long2<7))
up_long2++;
        if ((color==4)&&(getpixel(x,y-35)==11)&&(xp1<4)) xp1++;
        if ((color==9)&&(getpixel(x,y-35)==11)&&(xp2<4)) xp2++;
        setcolor(0);
        setfillstyle(1,0);
        fillellipse(x,y,15,15);
        setfillstyle(1,color);
        y-=35;
        fillellipse(x,y,15,15);
        return 0;
    }
    /*-----DOWN-----*/
int move_down(int x, int &y, int color)
{
    if (y+35>getmaxy()-61) return 0;
    if
((getpixel(x,y+35)==6)||((getpixel(x,y+35)==8)||((getpixel(x,y+35)==4)||((getpixe
l(x,y+35)==9)||((getpixel(x,y+35)==7)) return 0;
        if ((getpixel(x,y+35)==13)&&(color==4)&&(up_long1<7))
up_long1++;
        if ((getpixel(x,y+35)==13)&&(color==9)&&(up_long2<7))
up_long2++;
        if ((color==4)&&(getpixel(x,y+35)==11)&&(xp1<4)) xp1++;
        if ((color==9)&&(getpixel(x,y+35)==11)&&(xp2<4)) xp2++;
        setcolor(0);
        setfillstyle(1,0);
        fillellipse(x,y,15,15);
        setfillstyle(1,color);
        y+=35;
        fillellipse(x,y,15,15);
        return 0;
    }
    /*-----LEFT-----*/
int move_left(int &x, int y, int color)
{
    if (x-35<36) return 0;
    if ((getpixel(x-35,y)==6)||((getpixel(x-35,y)==8)||((getpixel(x-
35,y)==4)||((getpixel(x-35,y)==9)||((getpixel(x-35,y)==7)) return 0;
        if ((getpixel(x-35,y)==13)&&(color==4)&&(up_long1<7))
up_long1++;
        if ((getpixel(x-35,y)==13)&&(color==9)&&(up_long2<7))
up_long2++;

```

```

    if ((color==4)&&(getpixel(x-35,y)==11)&&(xp1<4)) xp1++;
    if ((color==9)&&(getpixel(x-35,y)==11)&&(xp2<4)) xp2++;
    setcolor(0);
    setfillstyle(1,0);
    fillellipse(x,y,15,15);
    setfillstyle(1,color);
    x-=35;
    fillellipse(x,y,15,15);
return 0;
}
/*-----RIGHT-----*/
int move_right(int &x, int y, int color)
{
    if (x+35>getmaxx()-77) return 0;
    if
((getpixel(x+35,y)==6)||(getpixel(x+35,y)==8)||(getpixel(x+35,y)==4)||(getpixel(x+35,y)
==9)||(getpixel(x+35,y)==7)) return 0;
    if ((getpixel(x+35,y)==13)&&(color==4)&&(up_long1<7)) up_long1++;
    if ((getpixel(x+35,y)==13)&&(color==9)&&(up_long2<7)) up_long2++;
    if ((color==4)&&(getpixel(x+35,y)==11)&&(xp1<4)) xp1++;
    if ((color==9)&&(getpixel(x+35,y)==11)&&(xp2<4)) xp2++;

    setcolor(0);
    setfillstyle(1,0);
    fillellipse(x,y,15,15);
    setfillstyle(1,color);
    x+=35;
    fillellipse(x,y,15,15);
return 0;
}

```

Учебное издание

**КУРСОВОЕ ПРОЕКТИРОВАНИЕ
ПО ОСНОВАМ ПРОГРАММИРОВАНИЯ**

Методические рекомендации

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

Формат 60×90/16
Уч. изд. 7,2 п.л. Тираж 10 экз.