

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

**О. Ч. Ролич**

***ОСНОВЫ ПРОГРАММИРОВАНИЯ  
ИНФОРМАЦИОННЫХ СИСТЕМ.  
КУРСОВОЕ ПРОЕКТИРОВАНИЕ***

*Рекомендовано УМО по образованию в области информатики  
и радиоэлектроники в качестве пособия для специальности  
1-40 05 01 «Информационные системы и технологии (по направлениям)»  
направления специальности  
1-40 05 01-10 «Информационные системы и технологии  
(в бизнес-менеджменте)»*

Минск БГУИР 2020

УДК 004.415.2(076)  
ББК 32.973.202я7  
Р67

**Рецензенты:**

кафедра интеллектуальных и мехатронных систем  
Белорусского национального технического университета  
(протокол №11 от 20.03.2019);

доцент кафедры моделирования и проектирования  
учреждения образования  
«Белорусский государственный аграрный технический университет»  
кандидат технических наук, доцент Е. В. Галушко

**Ролич, О. Ч.**

Р67 Основы программирования информационных систем. Курсовое проектирование : пособие / О. Ч. Ролич. – Минск : БГУИР, 2020. – 84 с. : ил.  
ISBN 978-985-543-534-2.

Сформулированы основные положения, касающиеся выполнения курсовой работы по дисциплине «Основы программирования информационных систем», приведены требования к оформлению пояснительной записки и графического материала, методика построения пользовательского интерфейса, схем алгоритмов, диаграмм состояний и последовательности, даны рекомендации по решению типовых задач в рамках курсовой работы, а также рассмотрены типичные ошибки и проблемы, возникающие в процессе её выполнения

**УДК 004.415.2(076)**  
**ББК 32.973.202я7**

**ISBN 978-985-543-534-2**

© Ролич О. Ч., 2020  
© УО «Белорусский государственный университет информатики и радиоэлектроники», 2020

## Содержание

Введение.....	5
1 Цели, задачи и форма организации курсовой работы по дисциплине «Основы программирования информационных систем».....	7
2 Общие требования к курсовой работе.....	12
2.1 Требования к оформлению титульного листа.....	12
2.2 Требования к реферату курсовой работы.....	13
2.3 Требования к заданию по курсовой работе.....	14
2.4 Краткие требования к оформлению содержания.....	15
2.5 Требования к введению.....	15
2.6 Краткие требования к основному тексту пояснительной записки.....	15
2.7 Требования к оформлению заключения.....	16
2.8 Требования к оформлению списка использованных источников.....	16
2.9 Правила оформления приложений.....	16
2.10 Оформление ведомости документов.....	16
2.11 Исходные данные к курсовой работе.....	19
2.12 Требования к программной реализации.....	19
3 Оформление пояснительной записки.....	20
3.1 Оформление рубрикации, заголовков и содержания.....	20
3.2 Основные правила изложения текста.....	20
3.3 Основные правила написания математических формул.....	20
3.4 Оформление иллюстраций.....	21
3.5 Оформление таблиц.....	21
3.6 Оформление ссылок, сносок и примечаний.....	21
4 Оформление графического материала.....	23
4.1 Методика проектирования и требования к пользовательскому интерфейсу.....	23
4.2 Требования к оформлению линий чертежей.....	34
4.3 Методика построения и требования к схеме алгоритма.....	34
4.4 Методика построения и требования к диаграмме состояний.....	42
4.5 Методика построения и требования к диаграмме последовательности.....	47
5 Дополнительные указания по темам курсовой работы.....	52
5.1 Статистический анализ массивов данных.....	53
5.2 Регрессионный анализ массивов данных.....	54
5.3 Спектральный анализ массивов данных.....	56
5.4 Корреляционный анализ массивов данных.....	57
5.5 Логический анализ данных интерфейсов связи.....	60
5.6 Протоколы систем телемеханики и телеметрии.....	63

5.7	Формирование и преобразование массивов .....	64
5.8	Линейное программирование.....	65
5.9	Динамическое программирование.....	66
	Заключение.....	73
	Приложение А (обязательное) Образец задания по курсовой работе.....	75
	Приложение Б (обязательное) Образец титульного листа.....	77
	Приложение В (обязательное) Пример оформления приложения к пояснительной записке курсовой работы.....	78
	Приложение Г (обязательное) Пример оформления ведомости документов к курсовой работе.....	79
	Приложение Д (обязательное) Пример оформления страницы пояснительной записки с иллюстрацией, расположенной между абзацами.....	80
	Список использованных источников.....	81

Библиотека БГУМР

## ВВЕДЕНИЕ

Курсовая работа как вид курсового проектирования и одна из форм текущей аттестации студента является обязательным элементом подготовки специалистов с высшим образованием и подготовительным этапом к выполнению дипломного проекта [1]. Она представляет собой самостоятельный студенческий труд, заключающийся в решении учебной или реальной профессиональной задачи по изучаемой дисциплине. Курсовая работа отражает способности владения базовыми знаниями по изучаемой и смежным дисциплинам, творческие возможности студента, его аккуратность, целеустремлённость, умение думать, анализировать, излагать мысли в научно-техническом стиле и доводить дело до логического завершения. Курсовая работа содержит результаты теоретических и (или) экспериментальных исследований по учебной дисциплине и включает совокупность аналитических, расчётных, исследовательских, оценочных заданий, объединённых общностью рассматриваемого объекта, и предполагает выполнение отдельных элементов конструкторских, технологических, программных, организационно-управленческих, экономических и других работ и разработку графической документации, в том числе плакатов.

Целью курсовой работы является решение задачи, поставленной перед студентом кафедрой или ведущим преподавателем по изучаемой дисциплине. При этом студенту выдаётся заполненный бланк задания на курсовую работу, образец которого представлен в приложении А. Решение поставленной задачи сопровождается программным, печатным и графическим материалом. Печатный материал подкрепляется пояснительной запиской, в которой автор-студент последовательно и логично излагает этапы и обосновывает методы решения, поясняя его различными способами: вербально, таблично, графически с помощью диаграмм и схем.

В задачи курсовой работы как одного из этапов подготовки к дипломному проектированию входят [1]:

- освоение, углубление, обобщение и систематизация знаний, полученных студентом в процессе обучения;
- приобретение практических навыков и развитие творческих подходов к решению конкретной инженерной или инженерно-экономической задачи;
- формирование умений использовать справочную литературу, нормативную, правовую, нормативно-техническую документацию, осуществлять патентный поиск;
- приобретение навыков по оформлению текстовой и графической документации согласно требованиям государственных стандартов и стандарта предприятия «Дипломные проекты (работы). Общие требования. СТП 01-2017» [2].

Данное пособие соответствует Положению об организации и проведении курсового проектирования в БГУИР и уточняет специфику курсовой работы по дисциплине «Основы программирования информационных систем».

Основные документы, регламентирующие организацию курсовой работы по дисциплине «Основы программирования информационных систем», представлены следующим перечнем:

- Положение об организации и проведении курсового проектирования в БГУИР [1];
- стандарт предприятия БГУИР СТП 01-2017 [2];
- ГОСТ 2.104-2006 «Единая система конструкторской документации. Основные надписи» [3];
- ГОСТ 2.105-95 «Единая система конструкторской документации. Общие требования к текстовым документам» [4];
- ГОСТ 7.1-2003 «Библиографическая запись. Библиографическое описание» [5];
- ГОСТ 2.106-96 «Единая система конструкторской документации. Текстовые документы» [6];
- ГОСТ 2.004-88 «Единая система конструкторской документации. Общие требования к выполнению конструкторских и технологических документов на печатающих и графических устройствах вывода ЭВМ» [7];
- ГОСТ 7.9-95 «Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Реферат и аннотация. Общие требования» [8];
- учебная программа учреждения высшего образования по учебной дисциплине «Основы программирования информационных систем» для направления специальности 1-40 05 01-10 «Информационные системы и технологии (в бизнес-менеджменте)».

# **1 ЦЕЛИ, ЗАДАЧИ И ФОРМА ОРГАНИЗАЦИИ КУРСОВОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ «ОСНОВЫ ПРОГРАММИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ»**

Согласно учебной программе цель курсовой работы по учебной дисциплине «Основы программирования информационных систем» состоит:

- в систематизации и закреплении теоретических знаний студентов по основным разделам дисциплины;
- в углубленном изучении методов и алгоритмов генерирования, обработки и анализа потока многомерных массивов данных;
- в получении практических навыков работы с современными компьютерными средствами математического и статистического анализа и программирования на произвольном языке.

Тематика курсовой работы должна быть актуальной и соответствовать современному состоянию и перспективам развития науки, техники и образования. Темы могут быть посвящены разработке программ спектрального, статистического и корреляционного анализа потока многомерных массивов данных, решения транспортных задач, прокладки оптимальных путей, прогнозирования динамики финансово-экономической деятельности бизнес-системы, моделирования физических и информационных систем.

Тематика курсовой работы должна отвечать учебным задачам дисциплины и строиться на фактическом материале организаций и учреждений реального сектора экономики, научных исследований кафедры. В тематику могут быть включены работы, связанные с научными, проектно-конструкторскими, организационно-управленческими, экономическими, лабораторными и компьютерными исследованиями, которые отвечают требованиям учебных программ дисциплины «Основы программирования информационных систем», а также основополагающих и сопутствующих ей учебных дисциплин.

При выполнении курсовой работы решаются задачи генерирования массивов данных, анализа их качества, спектральной и статистической обработки, анализа финансово-экономической деятельности предприятия, оптимального планирования производства и распределения ресурсов, прокладки наилучшего пути и иные транспортные, экономические, технические и информационные задачи.

Темы курсовой работы разрабатываются и утверждаются кафедрой проектирования информационно-компьютерных систем, обеспечивающей изучение дисциплины «Основы программирования информационных систем». Темы курсовой работы разрабатываются и утверждаются до начала семестра, в котором предусмотрена курсовая работа по учебной дисциплине. Количество утверждённых тем должно быть достаточным для выдачи в учебной группе каждому студенту индивидуального задания.

По дисциплине «Основы программирования информационных систем» сформирован следующий перечень тем курсовой работы:

- 1 Программа спектрального анализа массива случайных чисел.
- 2 Программа спектральной обработки векторных изображений.
- 3 Программа масштабирования изображений.
- 4 Программа фильтрации изображений.
- 5 Программа корреляционного анализа векторных изображений.
- 6 Программа корреляционного анализа одномерных массивов данных.
- 7 Программа анализа гистограмм изображений.
- 8 Программа анализа гистограмм одномерных массивов данных.
- 9 Программа регрессионного анализа одномерных массивов данных.
- 10 Программа решения транспортной задачи симплекс-методом.
- 11 Программа прокладки оптимального пути.
- 12 Программа прогнозирования динамики финансово-экономической деятельности предприятия.
- 13 Программа оптимального планирования распределения ресурсов.
- 14 Программа оптимального планирования производства.
- 15 Программа генерации векторных аудиофайлов.
- 16 Программа генерации векторных изображений.
- 17 Программа декодирования одномерных штрих-кодов.
- 18 Программа декодирования QR-кодов.
- 19 Программа декодирования осциллограмм интерфейса IrDA.
- 20 Программа декодирования осциллограмм интерфейса USB.
- 21 Программа выделения замкнутых контуров на изображении.
- 22 Программа моделирования источника излучения.
- 23 Программа отображения кадров цифрового болометра.
- 24 Программа обмена текстовыми сообщениями средствами Ethernet.
- 25 Программа построения дерева Хаффмана по данным JPEG-файла.
- 26 Программа JPEG-компрессии.
- 27 Программа JPEG-декомпрессии.
- 28 Программа решения транспортной задачи методом потенциалов.
- 29 Программа решения транспортной задачи средствами венгерского алгоритма.
- 30 Программа стохастического динамического планирования оптимального производства.
- 31 Программа динамического планирования распределения ресурсов с мультипликативным критерием.
- 32 Программа обмена видеоданными средствами Ethernet.
- 33 Программа обмена текстовыми сообщениями по протоколу MQTT.
- 34 Программа обмена видеоданными по протоколу MQTT.
- 35 Программа обмена данными по протоколу MODBUS RTU.
- 36 Программа обмена данными по протоколу MODBUS TCP.
- 37 Программа обмена данными по протоколу МЭК-104.



- 38 Программа оптимального управления спортивным клубом.
- 39 Программа качественного анализа аудиофайлов с использованием спектрограмм.
- 40 Программа анализа работы нейронных сетей для распознавания объектов на изображениях.
- 41 Программа анализа метрических характеристик подобных фигур на изображениях.
- 42 Программа анализа спектрограмм речевых команд.
- 43 Программа анализа спектрограмм звучания музыкальных инструментов.
- 44 Программа анализа образов показаний измерительных приборов.
- 45 Программа анализа данных измерений инерциальных навигационных приборов.

Представленный перечень тем может ежегодно незначительно меняться и дополняться согласно решению кафедры.

Исходя из Положения об организации и проведении курсового проектирования в БГУИР [1], студент вправе либо выбрать тему курсовой работы из числа предложенных преподавателем (кафедрой), либо самостоятельно предложить тему курсовой работы с обоснованием её целесообразности.

Студент уточняет с руководителем задачи курсовой работы, вариант задания, исходные данные, оформляет задание по работе в соответствии с формой, приведенной в приложении А.

В задании руководитель чётко формулирует исходные данные для выполнения расчётов, устанавливает объём и содержание графической части и пояснительной записки и составляет календарный план работы.

Пояснения к заданиям по темам курсовой работы отражены в разделе 5 «Дополнительные указания по темам курсовой работы» настоящего пособия.

Задание распечатывается на одном листе с обеих сторон, подписывается студентом и руководителем работы, датируется днём выдачи, регистрируется преподавателем и утверждается заведующим кафедрой.

Задание по курсовой работе выдаётся студенту в сроки, установленные Положением Министерства образования Республики Беларусь от 29 мая 2012 г., №53, а именно:

- студенту очной и вечерней форм обучения в первые две недели после начала семестра, в котором курсовая работа предусмотрена учебными планами;
- студенту заочной формы обучения на лабораторно-экзаменационной (установочной) сессии, предшествующей семестру, в котором курсовая работа предусмотрена учебными планами.

После получения студентом задания по курсовой работе и утверждения его заведующим кафедрой руководитель:

- назначает график консультаций студента по всем вопросам, связанным с выполнением курсовой работы;
- контролирует ход выполнения курсовой работы студентом;

- оценивает выполнение студентом каждого этапа (в процентах) курсовой работы;
- оказывает помощь студенту в подборе необходимой литературы и проведении патентного поиска;
- в установленные сроки предоставляет в деканат данные о выполнении студентом графика курсовой работы (в процентах).

На консультациях руководитель проверяет состояние работы над проектом, даёт конкретные указания по преодолению затруднений, анализирует типовые ошибки, помогает студенту находить рациональные пути их устранения. График консультаций преподавателя доводится до сведения студентов (вывешивается на кафедральной доске объявлений).

Студент обязан самостоятельно выполнять курсовую работу, оформлять пояснительную записку и графическую часть в соответствии с требованиями действующих стандартов ЕСКД, ЕСТД, ЕСПД и после каждого этапа курсовой работы предоставлять руководителю результаты теоретического анализа, систематизированный материал литературно-патентного поиска, выполненные расчёты и другие материалы на проверку. Руководитель проверяет сделанную работу, указывает ошибки, разъясняет недоработанные места и даёт рекомендации по их исправлению.

Студент несёт персональную ответственность за принятые в работе решения и достоверность их обоснования.

Завершённая курсовая работа, подписанная студентом, предоставляется руководителю в срок, установленный календарным планом. Выполненная курсовая работа может быть сдана на проверку руководителю до срока, указанного в календарном плане.

Руководитель проверяет полноту представленных материалов, соответствие их заданию, выясняет готовность работы к защите и по согласованию со студентом устанавливает дату защиты.

В случае неготовности курсовой работы либо необходимости внести правки студенту предоставляется дополнительный срок (с конкретным указанием требуемых исправлений).

После внесения исправлений и доработки курсовой работы студент повторно предоставляет её руководителю для проверки и защиты, но не позднее чем за три дня до защиты. Устранение недостатков, отмеченных руководителем, контролируется комиссией в процессе защиты.

Текущая аттестация в форме защиты курсового проекта (работы) производится комиссией, которая формируется заведующим кафедрой в составе не менее двух человек с участием руководителя курсового проекта. На защите возможно присутствие студентов группы (потока).

По решению заведующего кафедрой в качестве курсовой работы может быть представлена к защите научно-исследовательская, опытно-конструкторская и творческая работа, успешно выполненная студентом и отвечающая требованиям учебной программы.

Защита курсовой работы по «групповому» заданию, предусматривающему работу нескольких студентов над одним проектом, должна в обязательном порядке осуществляться в один день и при участии всех исполнителей разработки проекта. Защиту подобных проектов целесообразно организовывать в строгой последовательности отдельных частей, логически вытекающих одна из другой. Порядок такой защиты должен быть оговорен заранее на стадии выдачи задания и доведён до каждого исполнителя.

Защита состоит в коротком (5–10 минут) докладе студента по выполненному проекту и в ответах на вопросы членов комиссии. Студент должен при защите работы дать чёткие объяснения по существу работы. Доклад может сопровождаться электронной презентацией, разработанной студентом.

Комиссия оценивает результаты защиты каждой курсовой работы методом взвешивания оценок в следующей пропорции «пояснительная записка : графический материал : программа : доклад = 3 : 3 : 3 : 1» и принимает решение об отметке, учитывая при этом полноту представленного материала, обоснованность принятых решений, содержание доклада, ответы на вопросы, соблюдение требований стандартов к графическим и текстовым документам. Формат взвешивания оценок может незначительно меняться по усмотрению комиссии.

Положительные отметки вносятся в зачётно-экзаменационную ведомость и зачётную книжку. Неудовлетворительные отметки вносятся в зачётно-экзаменационную ведомость, а в зачётную книжку не вносятся. Любая отметка, положительная или неудовлетворительная, записывается на титульном листе пояснительной записки, где также указывается дата аттестации, и ставятся подписи членов комиссии. Неявка студента в установленные сроки защиты курсовой работы отмечается в зачётно-экзаменационной ведомости словами «не явился». В случае неявки студента по уважительной причине, подтверждённой документально, декан факультета слова «не явился» дополняет словами «по ув. причине». Неявка студента без уважительной причины оценивается деканом отметкой «1 балл», а студент считается имеющим академическую задолженность.

Курсовую работу, имеющую теоретический и практический интерес, рекомендуется представлять на конкурсах научных работ студентов.

Итоги выполнения работ обсуждаются на кафедре и по мере необходимости или в соответствии с планом работы – на заседаниях совета факультета.

## **2 ОБЩИЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ**

Курсовая работа состоит из пояснительной записки и графической части. Она может дополнительно включать макеты, модели, образцы и т. д. Графический материал и пояснительная записка выполняются только с использованием средств вычислительной техники и оформляются в соответствии с требованиями стандартов ГОСТ 2.104-2006, ГОСТ 2.105-95, ГОСТ 7.1-2003, ГОСТ 2.106-96, ГОСТ 2.004-88, ГОСТ 7.9-95, стандарта предприятия БГУИР СТП 01-2017 и данного пособия [2–8].

Согласно пункту 1.2.4 СТП 01-2017 пояснительная записка к курсовой работе выполняется на листах формата А4 с применением печатающих и графических компьютерных устройств. Разрешается исключать рамки и элементы оформления листов пояснительной записки по ЕСКД.

Общее количество листов пояснительной записки без учёта приложений справочного или информационного характера, рисунков, таблиц и формул, как правило, составляет 30–50 страниц печатного текста, набранного в соответствии с требованиями, изложенными в пункте 2.1.1 СТП 01-2017.

Пояснительная записка должна быть переплетена (закреплена в твёрдой обложке) или помещена в стандартную папку для курсовой работы.

Согласно пункту 1.2.5 СТП 01-2017 пояснительная записка к курсовой работе должна содержать следующее: титульный лист; реферат; задание по курсовой работе; содержание; введение; разделы, содержание которых определяется заданием на курсовую работу; заключение; список использованных источников; приложения, если они необходимы; ведомость документов.

Стиль изложения текста пояснительной записки должен быть строгим, научно-техническим, в безличной форме.

Графический материал, в соответствии с заданием по курсовой работе, представляется на листах формата А4 или А3 отдельно от пояснительной записки. Графический материал одного вида оформляется в рамке с основной надписью. Дополнительные требования к графическому материалу изложены в разделе 4 «Оформление графического материала» настоящего пособия.

### **2.1 Требования к оформлению титульного листа**

**2.1.1** Пояснительная записка начинается с титульного листа, образец которого выдаётся кафедрой и представлен в приложении Б. Он создаётся только с применением компьютерных средств векторной графики и печатающего устройства.

**2.1.2** На титульном листе (см. приложение Б) вместо «учёная\_степень» записывается учёная степень руководителя курсовой работы: «канд. техн. наук» или «к.т.н.» – для учёной степени кандидата технических наук; «докт. техн. наук» или «д.т.н.» – для доктора технических наук; «канд. физ.-мат. наук» или «к.ф.-м.н.» – для кандидата физико-математических наук; «докт. физ.-мат. наук» или «д.ф.-м.н.» – для доктора физико-математических наук; «канд. экон.

наук» или «к.э.н.», «докт. экон. наук» или «д.э.н.» – соответственно для кандидата и доктора экономических наук и т. д. Вместо «учёное\_звание» записывается учёное звание руководителя: «доцент» или «профессор». Вместо «И.О.Фамилия\_рук», «ТЕМА\_КУРСОВОЙ\_РАБОТЫ», «№\_темы\_порядковый» и «И.О.Фамилия\_студента» записываются соответственно инициалы и фамилия руководителя, наименование темы в точности с выданным заданием, трёхзначный порядковый номер темы курсовой работы, например 003, и наконец, инициалы и фамилия студента.

**2.1.3** Наименование кафедры и факультета на титульном листе следует писать без сокращений.

**2.1.4** Наименование темы работы пишется прописными буквами. Оно должно в точности соответствовать теме задания по курсовой работе. Ниже наименования темы приводится обозначение пояснительной записки, которое состоит из шифра документа, включающего пятибуквенный код организации – БГУИР; код классификационной характеристики специальности 1-40 05 01 «Информационные системы и технологии»; код специализации 10 «Информационные системы и технологии (в бизнес-менеджменте)»; трёхсимвольный порядковый номер темы согласно перечню тем, утверждённых кафедрой ПИКС, например 008, с добавлением букв ПЗ.

Примеры обозначения пояснительной записки:

БГУИР КР 1-40 05 01-10 019 ПЗ; БГУИР КР 1-40 05 01-10 001 ПЗ.

**2.1.5** Важно заметить, что на титульном листе, кроме названий факультета, кафедры, дисциплины и темы курсовой работы, отмечается дата её представления на проверку, так как только после предварительной трёхдневной проверки руководителем курсовая работа допускается к защите при условии отсутствия в ней принципиальных ошибок и замечаний. В противном случае работа отправляется на доработку с повторными проверками.

**2.1.6** Титульный лист включается в общее количество страниц пояснительной записки, но номер страницы не проставляется.

## **2.2 Требования к реферату курсовой работы**

Реферат выполняется по ГОСТ 7.9-95. Слово «РЕФЕРАТ» записывается прописными буквами полужирным шрифтом по центру, страница не нумеруется, но включается в общее количество страниц пояснительной записки. В реферативной части приводятся ключевые слова, непосредственно касающиеся темы, и кратко излагается содержание курсовой работы. Основными аспектами в содержании должны быть: предмет проектирования (исследования); цель работы; данные, относящиеся к методам проектирования; результаты и выводы. Объём реферата ограничен текстом, который можно разместить на одной странице пояснительной записки. Например:

## РЕФЕРАТ

БГУИР КР 1-38 02 03 03 042 ПЗ

Иванов, И. И. Терморегулятор системы «Умный дом» на базе микроконтроллера семейства ARM Cortex-M0 / И. И. Иванов. – Минск : БГУИР, 2017. – 58 с.

Пояснительная записка 58 с., 11 рис., 7 табл., 12 источников, 2 приложения

УМНЫЙ ДОМ, СИСТЕМА АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ ТЕМПЕРАТУРЫ, ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ, ИНТЕРФЕЙСЫ СВЯЗИ, МИКРОКОНТРОЛЛЕР, АРХИТЕКТУРА ARM CORTEX-M0

Предмет: распределение температуры в помещении.

Объект: методы и средства регулирования температуры в помещении на базе контуров обратной связи.

Цель: разработка электронного модуля на базе микроконтроллера семейства ARM Cortex-M0, являющегося основой системы автоматического регулирования температуры в помещении, с возможностью удалённого контроля и управления через Internet.

Методология проведения работы: в процессе решения поставленных задач использованы принципы системного подхода, теория схемотехнического и конструкторско-технологического проектирования радиоэлектронных средств, аналитические и физико-математические методы, методы компьютерной обработки экспериментальных данных и компьютерного моделирования.

Результаты работы: выполнен анализ литературных источников и патентных исследований; рассмотрено общетехническое обоснование разработки электронного модуля; проведён схемотехнический анализ радиоэлектронного средства; рассчитаны параметры проектируемого изделия; осуществлено моделирование физических процессов, протекающих в объекте управления; разработана графическая часть проекта.

Модуль предназначен для регулирования температуры в помещении с возможностью дистанционного управления через Internet.

Область применения результатов: проектирование систем автоматического регулирования температуры в помещении и кондиционирования воздуха.

### 2.3 Требования к заданию по курсовой работе

**2.3.1** Задание по курсовой работе заполняется согласно стандартной форме. Пример оформления задания приведён в приложении А. Бланк задания заполняется с помощью печатающего устройства.

**2.3.2** Наименования факультета и кафедры пишутся сокращённо, специальность и специализация обозначаются кодами классификационных характеристик, например: специальность 1-40 05 01, специализация 10.

**2.3.3** В пункте 3 задания указываются исходные данные к работе: описание к выполнению, язык и среда программирования, характеристики массивов и структур, воздействий и т. д., основные показатели (параметры), которые должны быть достигнуты при применении разработки, назначение разработки.

**2.3.4** В пункте 4 отражаются наименования разделов пояснительной записки.

**2.3.5** Пункт 5 задания должен содержать перечень графического материала с точным указанием вида, формата и количества листов, а также точное наименование каждого плаката. Всего в перечне должно быть указано не менее трёх листов в пересчёте на формат А3.

**2.3.6** Лицевую и оборотную страницы листа задания не нумеруют, но включают в общее количество страниц пояснительной записки.

## **2.4 Краткие требования к оформлению содержания**

**2.4.1** Содержание помещается сразу после задания по курсовой работе. Слово «СОДЕРЖАНИЕ» пишется прописными буквами полужирным шрифтом по центру строки. В содержание включаются заголовки всех частей пояснительной записки, в том числе разделов и подразделов, приложений, спецификаций и ведомость документов.

**2.4.2** Расположение заголовков в содержании должно точно отражать последовательность и соподчинённость разделов и подразделов в тексте пояснительной записки.

## **2.5 Требования к введению**

**2.5.1** Введение (предисловие) помещается на отдельной странице. Слово «ВВЕДЕНИЕ» или «ПРЕДИСЛОВИЕ» записывается прописными буквами полужирным шрифтом по центру. Введение (предисловие) должно быть кратким и чётким, не должно быть общих мест и отступлений, непосредственно не связанных с разрабатываемой темой. Объём введения не должен превышать двух страниц.

**2.5.2** Рекомендуется следующее содержание введения (предисловия):

- краткий анализ достижений в области, которой посвящена тема курсовой работы;
- цель курсовой работы;
- принципы, положенные в основу работы, научного исследования или поиска технического решения;
- краткое изложение содержания разделов пояснительной записки с обязательным указанием задач, решению которых они посвящены.

## **2.6 Краткие требования к основному тексту пояснительной записки**

**2.6.1** В основном тексте пояснительной записки анализируются существующие решения по теме курсовой работы, определяются пути достижения цели, составляются требования, на основании которых разрабатываются конкретные методики и решения задач, принимаются математические, алгоритмические и программные решения.

**2.6.2** Общими требованиями к основному тексту пояснительной записки являются чёткость и логическая последовательность изложения материала, убедительность аргументации, краткость и ясность формулировок, исключаящих неоднозначность толкования, конкретность изложения результатов, доказательств и выводов. Текст пояснительной записки должен излагаться в научно-техническом стиле, в безличной форме.

## **2.7 Требования к оформлению заключения**

Заключение оформляется в полном соответствии с пунктом 1.2.15 стандарта предприятия БГУИР СТП 01-2017 [2].

## **2.8 Требования к оформлению списка использованных источников**

**2.8.1** «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ» следует оформлять по ГОСТ 7.1-2003. Название «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ» записывается прописными буквами с новой страницы по центру строки полужирным шрифтом. В «СПИСКЕ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ» позиции располагаются и нумеруются в той последовательности, в которой расположены и пронумерованы ссылки в тексте пояснительной записки. Образцы описания источников приведены в пункте 2.8.6 стандарта предприятия БГУИР СТП 01-2017 [2].

**2.8.2** Касательно списка использованных источников следует отметить:

1 В списке фамилия и инициалы первого автора разделяются запятой, а непосредственно инициалы – пробелом.

2 Вид издания (учеб. пособие, пособие, справочник и т. п.) указывается со строчной буквы.

3 Библиографические знаки (: ; – /) с обеих сторон отделяются пробелами.

4 Место издания следует писать в соответствии с ГОСТ 7.1-2003 [5].

5 Не допускаются ссылки на системы подсказок (help), а также сайт «Википедия» и другие аналогичные источники.

## **2.9 Правила оформления приложений**

Правила оформления приложений изложены в ГОСТ 2.105-95 и в подразделе 2.7 СТП 01-2017 [2]. Пример оформления приложения к курсовой работе приведён в приложении В настоящего пособия.

## **2.10 Оформление ведомости документов**

**2.10.1** «ВЕДОМОСТЬ ДОКУМЕНТОВ» соответствует составу курсовой работы и является последним обязательным листом пояснительной записки. Форма ведомости и её оформление приведены в приложении Г, где обозначения и наименования для графического материала должны соответствовать графам 1 и 2 рисунка 1 основной надписи графической части.

**2.10.2** В графе 1 основной надписи графической части указывается наименование изделия, прибора или краткая тема работы, а также наименование документа. Наименование изделия, прибора или краткая тема работы записывается в именительном падеже единственного числа. Оно должно соответствовать принятой терминологии и быть по возможности кратким. В наименовании, состоящем из нескольких слов, на первом месте помещается имя существительное, например, «Спектроанализатор двухмерный».



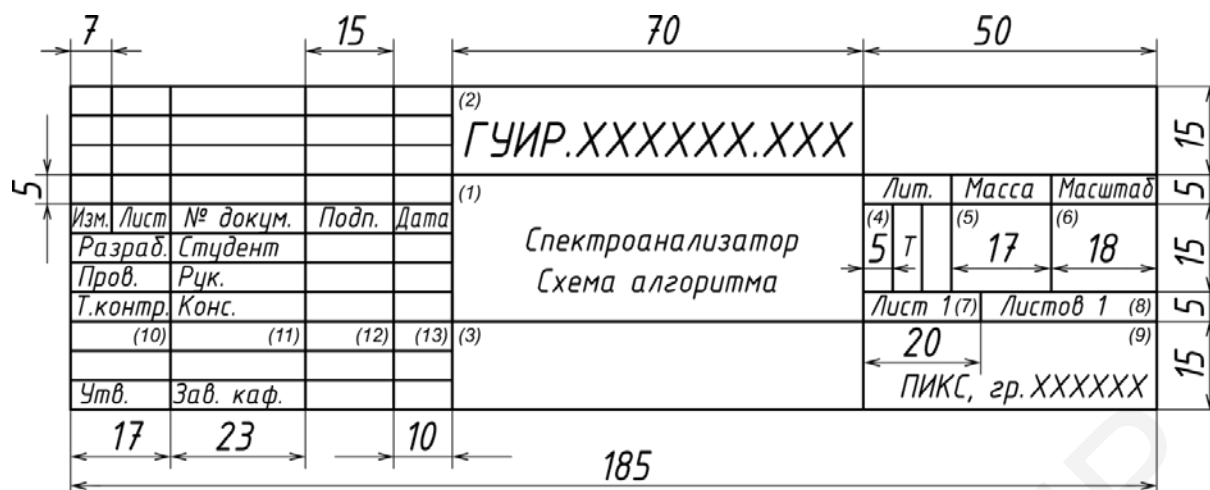


Рисунок 1 – Основная надпись для графических документов

**2.10.3** В графе 2 указывается обозначение документа по ГОСТ 2.201-80 [9]. Структура обозначения документа в основной надписи имеет шаблон «ХХХХ.ХХХХХХ.ХХХ». В данном шаблоне первые четыре символа обозначают код организации-разработчика, который состоит из четырёх букв (для курсовой работы, выполняемой в БГУИР – ГУИР). Следующие шесть символов определяют код классификационной характеристики, который выбирается по классификатору ЕСКД [9, 10]. Оставшиеся три символа – порядковый номер документа в соответствии с номером выбранной темы курсовой работы.

Код классификационной характеристики для программных средств выбирается исходя из следующего перечня [11]:

- 425113 «Программно-технические комплексы для автоматизации деятельности руководителя предприятия, организации»;
- 425124 «Программно-технические комплексы для автоматизации деятельности работника плановых органов и служб»;
- 425143 «Программно-технические комплексы для автоматизации работы диспетчера предприятия, производства»;
- 425162 «Программно-технические комплексы для автоматизации работы оператора склада»;
- 425196 «Программно-технические комплексы для автоматизации прочих работ и видов деятельности по управлению организационно-экономическими процессами»;
- 425514 «Программно-технические комплексы для автоматизации обмена данными (в интегрированных системах) с использованием локальных информационно-вычислительных сетей»;
- 425529 «Программно-технические комплексы для автоматизации обмена данными (в интегрированных системах) с использованием региональной вычислительной сети»;
- 425533 «Программно-технические комплексы для автоматизации обмена данными (в интегрированных системах) с использованием глобальной вычислительной сети»;

- 425590 «Программно-технические комплексы для автоматизации обмена данными (в интегрированных системах) прочие»;
- 425711 «Программно-технические комплексы для автоматизации обработки банковской информации»;
- 425726 «Программно-технические комплексы для автоматизации резервирования и продажи транспортных билетов»;
- 425745 «Программно-технические комплексы для автоматизации процессов обучения»;
- 425751 «Программно-технические комплексы для автоматизации обработки информации в торговле, материально-техническом обеспечении»;
- 425764 «Программно-технические комплексы для автоматизации обработки данных в системах связи»;
- 425798 «Программно-технические комплексы для автоматизации выполнения прочих работ в непромышленной сфере»;
- 501614 «Программные компоненты систем автоматизации технологии программирования»; в данную группу входят компоненты для проектирования интерфейсов пользователя, диагностики, тестирования и отладки, документирования, архивации, анализа;
- 501629 «Технологии автоматизации программирования»; в данную группу входят технологии для проектирования информационных систем, технических систем реального времени, программных систем;
- 501693 «Программные средства автоматизации технологии программирования прочие»;
- 501253 «Программные средства тестовые и диагностические»;
- 506113 «Программные средства для планирования и управления производством»;
- 506124 «Программные средства для управления сбытом»;
- 506139 «Программные средства для управления материально-техническим снабжением»;
- 506143 «Программные средства для управления техническим обслуживанием и ремонтом оборудования»;
- 506158 «Программные средства для контроля и управления качеством»;
- 506162 «Программные средства для управления трудовыми ресурсами»;
- 506188 «Программные средства для автоматизации делопроизводства, документооборота и управленческой деятельности»;
- 506196 «Программные средства для автоматизации управления предприятиями (организациями) прочие»;
- 506411 «Программные средства для автоматизации регулирования денежного обращения»;
- 506509 «Программные средства для автоматизации управления операциями с недвижимостью и арендой»;

- 507201 «Программные средства для управления учебным процессом»;
- 508810 «Мультимедиа-бизнес-приложения»;
- 508825 «Мультимедиа-развивающие программы».

За пояснениями граф 3–10 основной надписи графической части следует обратиться к пункту 3.1.4 стандарта предприятия БГУИР СТП 01-2017 [2].

## **2.11 Исходные данные к курсовой работе**

В качестве исходных данных к курсовой работе по дисциплине «Основы программирования информационных систем» выступают требования к пользовательской части программы, изложенные вербально, т. е. в виде текстового перечня базовой программной функциональности, доступной пользователю.

Дополнительно могут быть указаны тип операционной системы, под которой должна работать программа, язык и среда программирования, перечень используемых функций, библиотек и классов.

## **2.12 Требования к программной реализации**

**2.12.1** Программа курсовой работы должна быть написана на современном языке, с применением современных технологий программирования, в особенности объектно-ориентированных. Программная реализация должна иметь интуитивно понятный графический пользовательский интерфейс с количеством средств управления не менее десяти, включая компоненты просмотра изображений, построения графиков, таблиц, поверхностей. Взаимосвязи обработчиков событий, связанных с компонентами графического интерфейса, представляются в диаграмме состояний, имеющей не менее десяти узлов.

**2.12.2** Программа должна базироваться на вычислительных аппаратах и моделях, математически обоснованных в пояснительной записке и представленных в схемах алгоритмов. Количество блоков в отдельной схеме алгоритма должно быть как минимум пятнадцать и содержать ветвления.

**2.12.3** В реализации программного обеспечения следует учесть возможность анимации процессов генерирования, обработки и анализа массивов данных, заключающейся в визуальном движении объектов пользовательского интерфейса, изменении графиков, таблиц, изображений, «подсвечивании» наиболее информативных регионов.

### **3 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

Пояснительная записка выполняется с применением компьютерных печатающих и графических устройств вывода. При печати с помощью компьютерного текстового редактора используется гарнитура шрифта Times New Roman размером 13–14 пунктов с межстрочным интервалом, позволяющим разместить  $40 \pm 3$  строки на странице.

Номера разделов, подразделов, пунктов и подпунктов следует выделять полужирным шрифтом. Заголовки разделов рекомендуется оформлять полужирным шрифтом размером 14–16 пунктов, а подразделов – полужирным шрифтом 13–14 пунктов. Для акцентирования внимания на определённых элементах допускается использование курсивного и полужирного начертания.

Требования к расположению текста пояснительной записки представлены в пункте 2.1.2 и в приложении Л СТП 01-2017 [2]. Абзацы в тексте начинаются отступом 1,25 или 1,27 см, устанавливаемым в Word-редакторе в диалоговом окне «Абзац» (см. приложение Л СТП 01-2017 [2]).

Все части пояснительной записки необходимо излагать на одном языке – русском или белорусском, в научно-техническом стиле, в безличной форме. Для студентов, являющихся гражданами иностранных государств, получающих высшее образование на английском языке, допускается все части излагать на языке обучения.

Описки и графические неточности, обнаруженные в тексте пояснительной записки, не допускается исправлять подчисткой или закрашиванием белой краской.

#### **3.1 Оформление рубрикаций, заголовков и содержания**

Основные правила оформления рубрикаций, заголовков и содержания в контексте «курсовой работы» (или «дипломного проекта») отражены в подразделе 2.2 стандарта предприятия БГУИР СТП 01-2017 [2].

#### **3.2 Основные правила изложения текста**

При изложении текста пояснительной записки необходимо руководствоваться пунктом 2.3 стандарта предприятия БГУИР СТП 01-2017, откуда следует подчеркнуть, что текст пояснительной записки должен быть чётким, грамотным, логично изложенным, без допущения различных толкований [2]. Текстом должен быть связан весь иллюстрационный и табличный материал, формулы и программные фрагменты, с сохранением причинно-следственных связей в изложении.

#### **3.3 Основные правила написания математических формул**

Правила написания математических формул описаны в подразделе 2.4 СТП 01-2017 [2].

Следует отметить, что в связи с установленными требованиями к выполнению курсовой работы строго с применением печатающих и графических компьютерных устройств пункт 2.4.3 СТП 01-2017 сокращается и преобразуется к рекомендациям следующих межтекстовых промежутков для размещения формул:

- шесть интервалов для простейших однострочных формул;
- восемь интервалов для однострочных формул, содержащих знаки  $\Sigma$ ,  $\Pi$ ,  $\int$ .

В качестве дополнительного требования выступает запрет начинать формулой подпункт, пункт или раздел пояснительной записки. Завершать их формулами также настоятельно не рекомендуется. Следует обозначать вводную и завершающую текстовые части.

### **3.4 Оформление иллюстраций**

Правила оформления иллюстраций (рисунков) перечислены в подразделе 2.5 СТП 01-2017 [2].

Дополнительно к ним отмечается, что иллюстрации необходимо выполнять с помощью компьютерной техники и допускается цветное оформление отдельных иллюстраций, а разрешение каждой иллюстрации по горизонтали и вертикали должно быть не менее 300 dpi.

Запрещается начинать с иллюстрации пункт, подраздел или раздел пояснительной записки. Завершать их иллюстрациями также настоятельно не рекомендуется. До и после них следует обозначать вводную или завершающую, или поясняющую текстовые части так, как это сделано в приложении Д.

### **3.5 Оформление таблиц**

Требования к оформлению таблиц приведены в подразделе 2.6 стандарта предприятия БГУИР СТП 01-2017 [2].

Дополнительно к требованиям стандарта предприятия отмечается, что не следует начинать с таблицы пункт, подраздел или раздел пояснительной записки. Завершать их таблицей также настоятельно не рекомендуется. Следует обозначать вводную и завершающую текстовые части, приводить детальные пояснения зависимостей и закономерностей, представленных в табличном виде, делать конкретные выводы, замечания и краткое заключение.

### **3.6 Оформление ссылок, сносок и примечаний**

**3.6.1** Расшифровка ссылок на литературу, нормативно-техническую и другую документацию, иные источники, использованные при работе над курсовой работой, помещается в конце пояснительной записки перед приложениями в виде перечня «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ». Требования к оформлению списка использованных источников представлены в подразделе 2.8 СТП 01-2017 [2].

**3.6.2** В тексте пояснительной записки обязательно должны присутствовать ссылки на использованные литературные источники. Ссылки ставятся в

конце предложения с заимствованной из соответствующего литературного источника информацией.

**3.6.3** Ссылки на литературные источники оформляются в квадратных скобках, с указанием номера источника или их перечнем через запятую либо через тире. Последовательность ссылок определяется порядком их следования при изложении текста пояснительной записки от её начала. Частота ссылок на использованные литературные источники не должна превышать двух-трёх на абзац.

**3.6.4** В тексте пояснительной записки все ссылки на источники записываются арабскими цифрами в квадратных скобках в возрастающем порядке. Ссылки должны быть приведены на все без исключения источники, включенные в «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ».

**3.6.5** Без ссылок в тексте пояснительной записки разрешается использовать сведения, полученные на лекциях, семинарских, практических и лабораторных занятиях. Однако использованные учебные, учебно-методические материалы и пособия должны быть приведены и расположены в конце «СПИСКА ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ».

**3.6.6** Правилам оформления сносок и примечаний посвящён подраздел 2.9 стандарта предприятия БГУИР СТП 01-2017 [2].

## 4 ОФОРМЛЕНИЕ ГРАФИЧЕСКОГО МАТЕРИАЛА

Графическая часть курсовой работы выполняется и оформляется с использованием графических устройств компьютерной печати на листах чертёжной бумаги формата А3 или А4. Общий объём графической части указывается в техническом задании (см. приложение А).

Графический материал должен иметь рамку и основную надпись. Его форматы и правила выполнения должны соответствовать требованиям ЕСКД. На чертежах и схемах должны быть представлены все необходимые данные для однозначной передачи информации: условные графические обозначения элементов, их буквенно-цифровые позиционные обозначения, символы параметров в характерных точках схемы или диаграммы, поясняющие надписи и примечания.

Элементы, устройства и составные части на схемах изображаются в виде условных графических обозначений, установленных государственными стандартами ЕСКД, а их наименования и номера позиций должны соответствовать буквенным или буквенно-цифровым обозначениям по ГОСТ 2.701-2008 [12].

Листы формата А3, имеющие рамки и основную надпись, можно располагать горизонтально и вертикально. Листы формата А4 размером 210×297 мм располагаются только вертикально, а основные надписи располагаются внизу листа.

Формат листа и его расположение выбираются в зависимости от вида графического материала, его объёма, сложности и необходимости обеспечения на всех листах графической части курсовой работы единообразия выполнения.

Рамки на графических листах наносятся сплошной основной линией на расстоянии 5 мм от границы формата сверху, справа и снизу. Слева оставляется поле шириной 20 мм.

На листах формата А3 основная надпись, представленная на рисунке 1, располагается в правом нижнем углу. На листах формата А4 основная надпись располагается только вдоль короткой стороны листа.

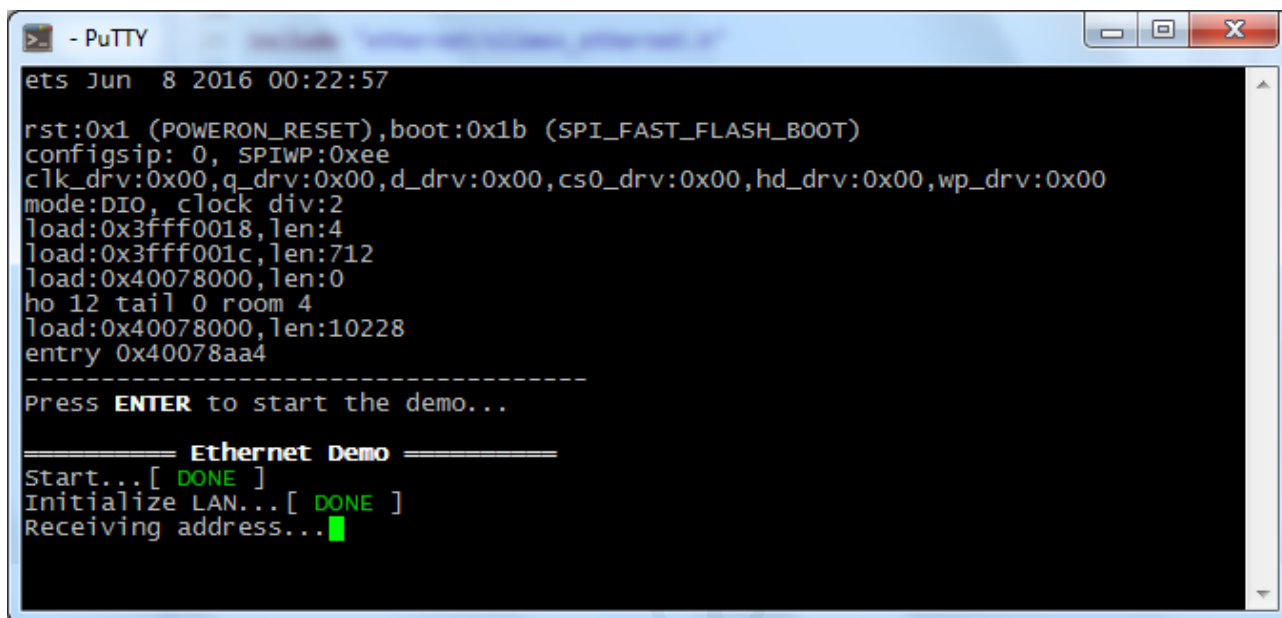
На документах, выполняемых в соответствии с ГОСТ 2.605-68 «ЕСКД. Плакаты учебно-технические. Общие технические требования», основная надпись помещается на оборотной стороне документа [13].

### 4.1 Методика проектирования и требования к пользовательскому интерфейсу

**4.1.1** Пользовательский интерфейс представляет собой средство взаимодействия программы с пользователем. Основная концепция системы управления пользовательским интерфейсом заключается в реализации принципа абстракции, а именно в отделении разработки пользовательского интерфейса от непосредственного приложения.

Различаются два вида пользовательского интерфейса: консольный и графический. Консольный интерфейс, пример которого представлен на рисунке 2,

связан с терминалом и командной строкой для обмена сообщениями, графический интерфейс – со стандартными визуальными средствами управления: кнопками, флажками, переключателями, полями редактирования, двухмерными и трёхмерными графиками, списками, таблицами и др. Таким образом, в графическом пользовательском интерфейсе элементы (меню, кнопки, значки, списки и т. п.) представлены пользователю в виде изображений, на которые пользователь воздействует определёнными способами.



```
- PuTTY
ets Jun  8 2016 00:22:57
rst:0x1 (POWERON_RESET),boot:0x1b (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0018,len:4
load:0x3fff001c,len:712
load:0x40078000,len:0
ho 12 tail 0 room 4
load:0x40078000,len:10228
entry 0x40078aa4

-----
Press ENTER to start the demo..

===== Ethernet Demo =====
Start... [ DONE ]
Initialize LAN... [ DONE ]
Receiving address... █
```

Рисунок 2 – Пример консольного пользовательского интерфейса

**4.1.2** В курсовой работе по дисциплине «Основы программирования информационных систем» требуются проектирование и обоснование пользовательского интерфейса графического типа.

**4.1.3** Графический пользовательский интерфейс создаётся, как правило, визуально посредством специализированных средств, позволяющих упростить его разработку путём спецификации компонентов пользовательского интерфейса с использованием графического и объектно-ориентированного языков спецификаций. В общем выделяются следующие основные способы спецификации интерфейса [14, 15]:

- языковой, когда применяются специальные языки для задания синтаксиса интерфейса (декларативные, объектно-ориентированные, языки событий);
- графический, связанный с определением интерфейса, как правило, средствами визуального программирования, программированием демонстраций и по примерам;
- объектно-ориентированный, основанный на принципе, называемом «непосредственное манипулирование», основное свойство которого заключается во взаимодействии пользователя с индивидуальными объектами, а не со всей системой как единым целым; типичными компонентами, используемыми для манипуляций с объектами и управляющими функциями, являются обработчики, меню, зоны диалога, кнопки различного вида;



– по спецификации прикладной задачи, когда интерфейс создаётся автоматически по спецификации семантики прикладной задачи.

**4.1.4** В разработке пользовательского интерфейса предполагается смешанное поведение интерфейса и прикладной программы. В целом выделяются три типа подобной связи, которые определяются характером взаимодействия с пользователем [14, 15]:

– инициатива диалога принадлежит пользователю, когда интерфейс предоставляет инициативу пользователю либо пользователь самостоятельно берёт на себя инициативу взаимодействия, а интерфейс поддерживает такую возможность;


– инициатива диалога принадлежит прикладной программе, это означает, что в случае, когда прикладной программе необходима некоторая информация, она запрашивает её у пользователя, а пользователь включается в процесс решения, когда необходимо ввести данные, требуемые системе;


– смешанная инициатива диалога, при которой пользователь определяет входные данные и прикладная программа, в случае необходимости дополнительных данных для решения задачи, запрашивает их у пользователя.


**4.1.5** Для проектирования графического пользовательского интерфейса рекомендуется использовать технологию визуального программирования как способа создания пользовательского интерфейса путём манипулирования графическими объектами вместо написания соответствующего кода в текстовом виде [14, 15].


**4.1.6** Технология визуального программирования основана на применении СОМ (Component Object Model)-объектов – объектной модели компонентов как технологического стандарта, предназначенного для создания программного интерфейса на основе взаимодействующих компонентов, каждый из которых может использоваться во многих программах одновременно. Достоинство СОМ-технологии заключается в возможности создания и использования СОМ-объектов вне зависимости от языка программирования. СОМ-объекты, как правило, содержащиеся в файлах с расширением DLL (например, в Windows-файлах OLE32.DLL и OLE-Aut32.DLL) или OCX, имеют полный набор свойств, методов и интерфейсов.


Среди наиболее распространённых простейших СОМ-объектов выделяются следующие средства управления пользовательским интерфейсом:


– статический текст  **Static Text** или **Label**, предназначенный только для чтения, который не может быть изменён пользователем и остаётся неизменным во время работы приложения или меняется лишь программно специальным методом соответствующего класса, например: методом `SetWindowText()` класса `CStatic` в языке Visual C++ или изменением свойства `Caption` класса `Label` в языке VBA или свойства `Text` класса `Label` в C#;

– поле редактирования  **Edit Box** или **TextBox** для ввода пользователем текстового сообщения или числовых данных, являющееся объектом класса `CEdit` в Visual C++ или класса `TextBox` в VBA или C#;


– триггерный переключатель типа **check**  (Check Box), предназначенный для множественного выбора и являющийся объектом класса CButton в языке Visual C++ или CheckBox в C#;

– многопозиционный переключатель типа **radio**  (радиопереключатель) (RadioButton или OptionButton), использующийся в выборе единственного объекта в группе и относящийся к классу CButton в языке Visual C++ или RadioButton в C#;


– кнопка  **Button, CommandButton, Push Button** как объект класса CButton языка Visual C++ или Button в C#;


– линейный список  **List Box** – объект класса CListBox в Visual C++ или ListBox в VBA (Visual Basic for Application) или C#;


– комбинированный список  **Combo Box** – объект класса CComboBox в Visual C++ или ListBox в VBA (Visual Basic for Application) или C#;



– полоса прокрутки  **Scroll Bar**, предназначенная для возможности вертикальной или горизонтальной прокрутки – объект класса CScrollBar в языке Visual C++ или класса HScrollBar/VScrollBar в C# либо класса ScrollBar в VBA (Visual Basic for Application);

– ползунок-регулятор  **Slider** или **TrackBar** как объект класса CSliderCtrl языка Visual C++ или TrackBar языка C#;

– прокрутик списка на один элемент  **Spin** или **NumericUpDown** – объект класса CSpinButtonCtrl языка Visual C++ или NumericUpDown языка C#;

– окно  **Picture**, или **Image**, или **PictureBox** для отображения и просмотра изображений – объект класса CStatic языка Visual C++, или PictureBox языка C#, или Image в VBA (Visual Basic for Application);

– группа визуальных средств управления  **Group Box** или **Frame** используется для размещения внутри его множества иных вышеперечисленных компонентов.

**4.1.7** Более развитые COM-объекты предоставляются специализированными программными средами автоматизированного проектирования и математическими пакетами. Так, средой Visual Studio либо редакторами Microsoft Office могут быть заимствованы из математического пакета MathCAD визуальные средства просмотра изображений, построения двухмерных графиков  и трёхмерных поверхностей  со всеми встроенными в них опциями и визуально настраиваемыми (на пользовательском уровне) режимами отображения. В качестве примера на рисунке 3 представлены настройки MathCAD-средства **Surface Plot** на фоне приложения, построенного в среде Visual Studio и разработанного на языке Visual C++.

Заимствованное из MathCAD средство **Surface Plot** в среде Visual Studio обладает теми же «правами», настройками и интерфейсом, что и в «родном» пакете (рисунок 4).

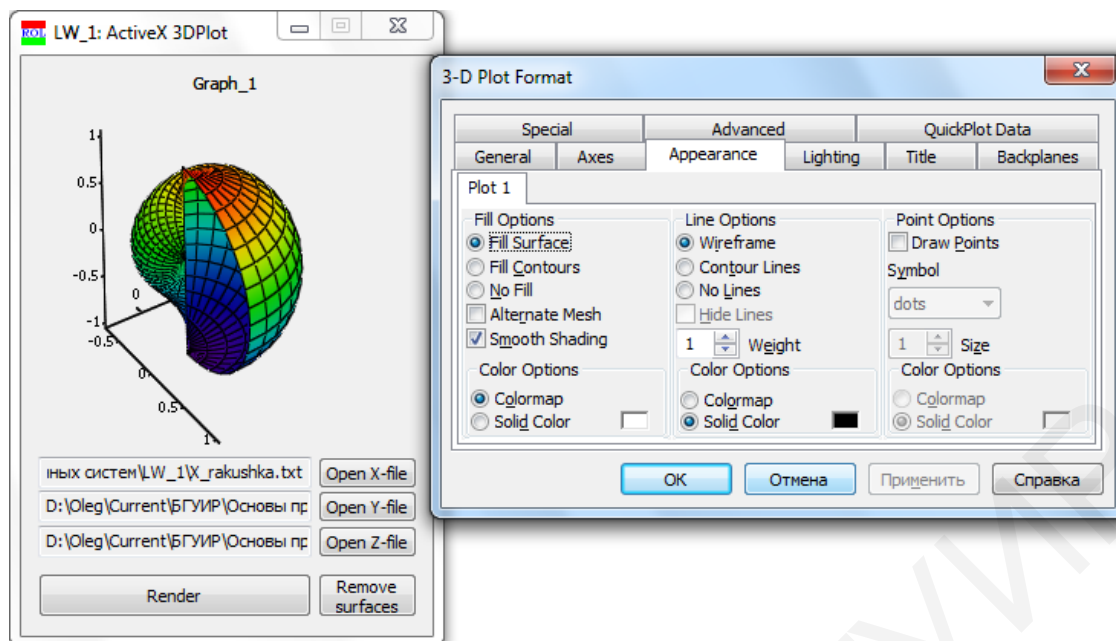


Рисунок 3 – Результат встраивания MathCAD-компонента **Surface Plot** в Visual Studio

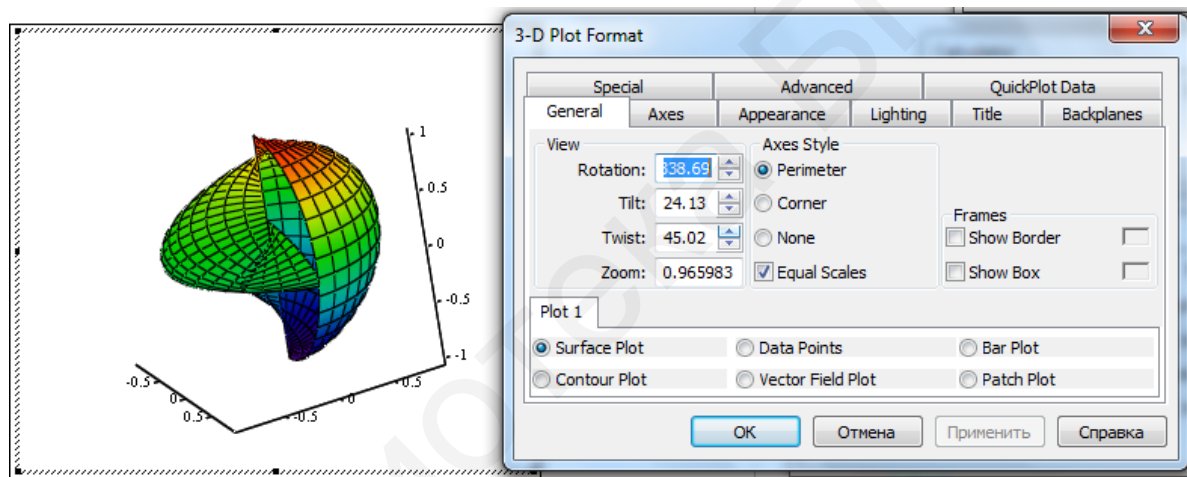


Рисунок 4 – Работа компонента **Surface Plot** в «родном» пакете MathCAD

При этом управление заимствованными COM-объектами на примере **Surface Plot** не требует от Visual C++-программиста дополнительных усилий по отношению к реализации математических операторов визуализации поверхностей, построения пространственной mesh-сетки, аффинных преобразований, моделированию источников света.

**4.1.8** Проектирование графического пользовательского интерфейса базируется на следующих этапах:

1 Анализ математической модели поставленной задачи в рамках курсовой работы.

2 Обоснование интерфейса программы путём выделения полного набора её входных данных из множества параметров математической модели.

3 Выбор из полученного набора входных программных данных пользовательской части, т. е. множества параметров, непосредственно зависящего от пользователя.

4 Выбор компонентов пользовательского интерфейса с точки зрения его компактности (отсутствия избыточности), удобства (интуитивной понятности), дизайна (формы, пространственной ориентации и цветовой гаммы), полноты, гибкости (возможности задания входных параметров разными способами и обеспечения заданной степени точности в решении задачи). Например, целочисленные параметры удобно задавать в виде ползунка-регулятора типа **Slider** или прокрутки **Spin** списка на один элемент, массивы – либо таблицами, либо файлами табличного вида, но текстового формата, впоследствии загружаемые в программу через указание полного пути к ним в поле редактирования **Edit Box/TextBox**.

5 Визуальное создание интерфейса пользователя.

В качестве примера рассматривается задача анализа поведения знаковой корреляции в модели сравнения векторного изображения-контейнера с эталонным векторным изображением.

Под изображением-контейнером понимается изображение, составленное из множества отдельных, аффинно преобразованных эталонных изображений, т. е. эталонных изображений, смещённых по горизонтали и вертикали, повернутых вокруг собственной перпендикулярной оси и изменённых в масштабе.

В качестве векторного изображения выступает изображение, распределение яркости и контуры объектов в котором задаются математически или в соответствии со строго определённым дискретным набором правил.

Дискретная двухмерная корреляционная функция двух изображений  $X$  и  $E$  размерами  $(M \times N)$  каждое представляется конечной суммой вида

$$R_{k,l} = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} X_{i+k,j+l} E_{i,j},$$

где  $X$  – фрагмент изображения-контейнера размером  $(M \times N)$ , левый верхний угол которого расположен в позиции с координатами  $(k, l)$ ;

$E$  – векторное изображение эталона,  $(m \times n)$  – размеры матрицы эталона  $E$ ;  
 $k, l$  – индексы:  $k = 0, \dots, (M - 1), l = 0, \dots, (N - 1)$ .

Так как в поставленной задаче анализируется поведение знаковой корреляции как меры сравнения объектов двух изображений (контейнерного и эталонного), то распределение яркости в эталонном изображении задаётся в бинарном виде парой  $\{1, -1\}$ . Значение «1» соответствует чёрному уровню, «-1» – белому. Например, бинарное векторное изображение эллипса с коэффициентом  $k$  вертикального сжатия и радиусом  $R$  большей полуоси выглядит следующим образом:

$$A_{i,j} = \begin{cases} -1, & \text{если } \left( \left( i - \frac{m}{2} \right)^2 + k \cdot \left( j - \frac{n}{2} \right)^2 \right) > R, \\ \text{иначе } 1, & \end{cases}$$

$i = 0, \dots, (m - 1), j = 0, \dots, (n - 1)$ .

Изображения треугольника и ромба с коэффициентом  $k$  вертикального сжатия соответственно:

$$B_{i,j} = \begin{cases} -1, \text{ если } \left( \left| j - \frac{n}{2} \right| - k \cdot i > 0 \right), \\ \text{иначе } 1; \end{cases}$$

$$C_{i,j} = \begin{cases} -1, \text{ если } \left( \left| i - \frac{m}{2} \right| + k \cdot \left| j - \frac{n}{2} \right| > \frac{\min(m,n)}{3} \right), \\ \text{иначе } 1. \end{cases}$$

Согласно поставленной задаче и её математической модели в пользовательском интерфейсе программы анализа поведения знаковой корреляции изображений выделяются компоненты, задающие тип (форму) объекта на изображении-контейнере, угол его поворота, измеряемый в градусах, коэффициент масштаба, горизонтальная и вертикальная координаты расположения объекта на результирующем изображении-контейнере. Форму объекта, в связи с относительно небольшим количеством вариантов, удобно задавать линейным или комбинированным списком. Угол поворота с точностью до  $1^\circ$  – ползунком-регулятором, координаты позиционирования, в связи с их дискретностью в один пиксель, – либо ползунком-регулятором, либо прокруткой списка на один элемент, либо полем редактирования, либо их совокупностью. Масштаб как вещественное число удобно устанавливать посредством поля редактирования.

Для задания формы эталонного образа и отображения результата расчётов предусмотрены отдельные области пользовательского интерфейса, включающие в том числе компоненты визуализации изображения-контейнера, построения результирующей трёхмерной поверхности знаковой корреляции и её интерпретации в виде линий уровней (рисунок 5).

На трёхмерной поверхности корреляционной функции рисунка 5 просматриваются локальные максимумы, соответствующие расположению отдельных объектов (ромба, треугольника, овала), с выраженным глобальным максимумом в месте локализации эталонного объекта (в данном случае таким является овал), с которым проводится операция корреляции.

На интерпретации трёхмерной поверхности знаковой корреляции в виде линий уровней отчётливо виден глобальный максимум, определяющий позицию эталонного изображения на изображении-контейнере, приводящую к совпадению эталонного изображения с соответствующим фрагментом изображения-контейнера.

**4.1.9** Проектирование визуального пользовательского интерфейса предполагает использование модели программирования, управляемого событиями. Эта модель программирования является основой программирования в современных операционных системах, в частности в Windows [16].

Пусть программа на языке Visual C++ пишется в некоторой операционной системе (рисунок 6) и описывает класс «Программный объект» объектов.

### Состав изображения-контейнера

	Угол поворота, град	Координаты (i,j) расположения
<input type="radio"/> Овал <input checked="" type="radio"/> Треугольник <input type="radio"/> Ромб	<input type="text" value="0"/>	<input type="text" value="15"/>
<input type="radio"/> Овал <input checked="" type="radio"/> Треугольник <input type="radio"/> Ромб	<input type="text" value="-180"/>	<input type="text" value="0"/>
<input type="radio"/> Овал <input type="radio"/> Треугольник <input checked="" type="radio"/> Ромб	<input type="text" value="180"/>	<input type="text" value="15"/>

### Эталонное изображение

 Овал  
 Треугольник  
 Ромб

Генерация изображения-контейнера

### Результирующее изображение-контейнер



### Поверхность корреляционной функции

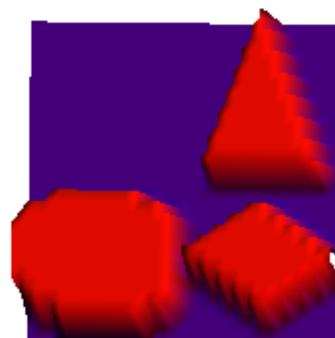
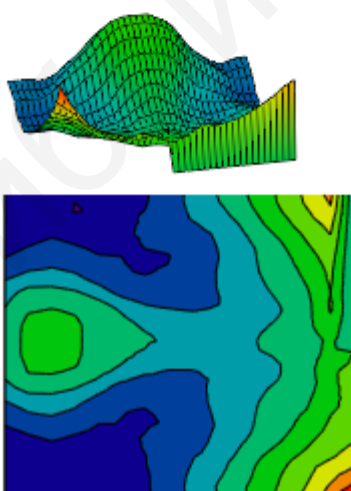


Рисунок 5 – Вариант пользовательского интерфейса

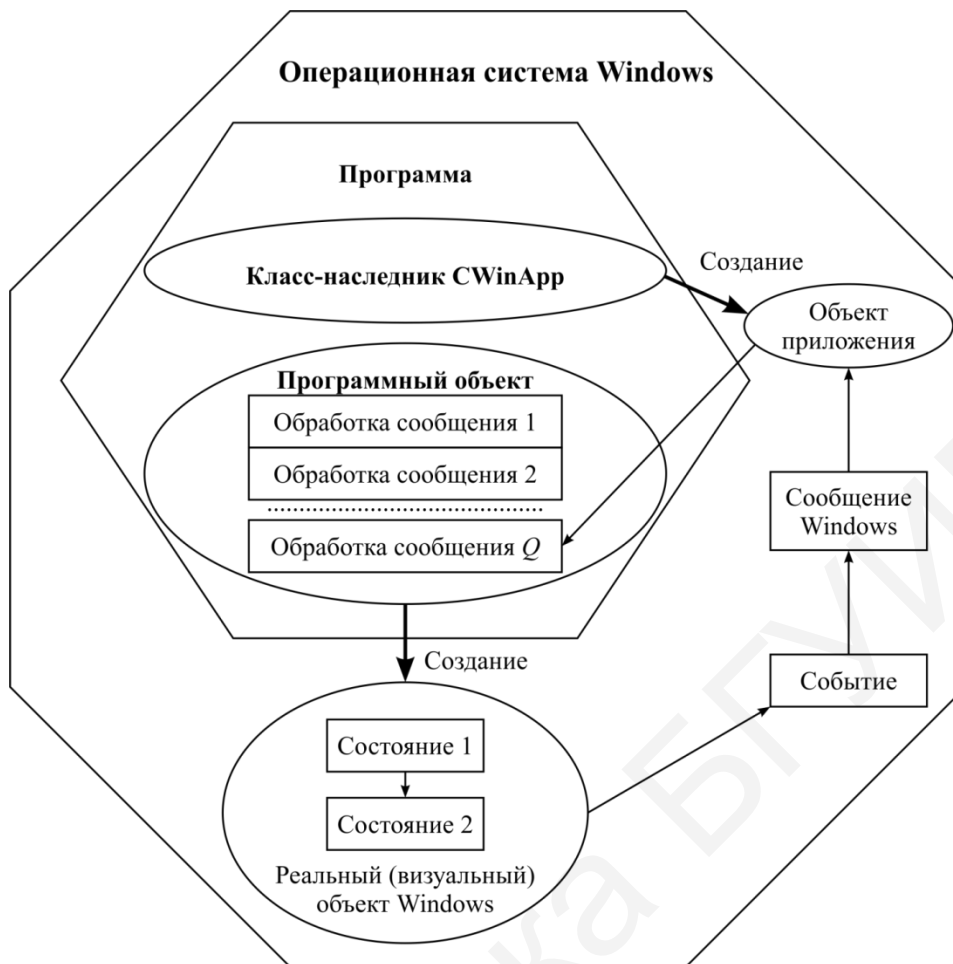


Рисунок 6 – Модель программирования, управляемого событиями

Во время её выполнения создаются реальные (или абстрактные, в зависимости от уровня абстракции) объекты, функции слежения за поведением которых операционная система полностью берёт на себя. Каждый объект характеризуется состоянием, поведением и паспортом. Состояние выражается текущими компонентами объекта, т. е. совокупностью его свойств; поведение определяет динамику изменения состояний и реакцию объекта на внешние воздействия; паспорт (или уникальное имя) позволяет отличить объект от других объектов того же или иного класса. Смена состояний объекта с течением времени порождает событие. Происхождение события фиксирует операционная система, которая генерирует соответствующее сообщение и передаёт его на обработку объекту приложения – наследнику класса CWinApp в Visual C++. В качестве конструктора класса-наследника CWinApp используется виртуальная функция

```
virtual bool InitInstance()
```

инициализации приложения. Объект приложения занимает руководящую роль во всей системе объектов программы. Он, анализируя принятое сообщение, передаёт его на обработку непосредственно исполнителю того объекта, который изменил состояние и породил сообщение Windows. В качестве исполнителей

обычно выступают функции, так называемые триггеры, задачей которых является обработка произошедшего события.

Таким образом, в программе явным или неявным образом присутствует объект приложения, который принимает поток сообщений, поступающих от операционной системы.

Неявное присутствие объекта приложения имеет место в «интегральных» средах, а именно в математических пакетах и системах автоматизированного проектирования, поддерживающих скриптовые сценарии обработки событий и использующих скриптовые языки, например: VBS (Visual Basic Script), JS (Java Script). В подобных случаях объектом приложения выступает непосредственно «интегральная» среда. Так, для пользовательского интерфейса на рисунке 5, созданного в математическом пакете MathCAD, триггеры `ListBox_SelChanged()` событий выбора пункта линейного списка, `ListBox_DblClick()` двойного щелчка левой кнопкой мыши на соответствующем пункте линейного списка, `Slider_ValueChanged()` изменения позиции ползунка-регулятора, `PushBtn_Click()` нажатия кнопки, `ScrollBarEvent::Exec(Inputs,Outputs)` выполнения сценария, связанного с полосой прокрутки, имеют скриптовую основу на языках VBS и JS и, в частности, могут выглядеть следующим образом:

```
Sub ListBox_SelChanged()  
    ListBox.Recalculate()  
End Sub  
Sub ListBox_DblClick()  
    ListBox.Recalculate()  
End Sub  
Sub Slider_ValueChanged()  
    Slider.Recalculate()  
End Sub  
Sub PushBtn_Click()  
    PushBtn.Recalculate()  
End Sub  
function ScrollBarEvent::Exec(Inputs,Outputs)  
{  
    // TODO: Add code here  
    Worksheet.SetValue("pdmPeriod", ScrollBar.Value);  
}
```

Если определения первых четырёх триггеров удовлетворяют синтаксису языка Visual Basic и отчасти напоминают синтаксис языка C#, то синтаксис последнего триггера соответствует языкам C++ и JavaScript.

Полный перечень событий, генерируемых простейшими COM-компонентами управления пользовательским интерфейсом, представлен в документации MSDN (Microsoft Development Network) [17]. Важно помнить, что имена триггеров-обработчиков стандартных событий имеют стандартизированные имена, не зависящие от языка программирования при реализации проекта пользовательского интерфейса.



Имена визуальных компонентов как программных объектов, посредством которых осуществляется доступ к свойствам и методам компонентов и управление ими, задаются в их свойствах «Properties», точнее в поле «Variable name» (связанного с «(Name)») языка Visual C++, «(Name)» языков C# и VBA, «Name used for this object in the script» пакета MathCAD.

**4.1.10** Проект пользовательского интерфейса должен содержать не менее десяти компонентов.

**4.1.11** Компоненты пользовательского интерфейса должны иметь размеры, удовлетворяющие закону «золотого сечения». Они не должны быть необоснованно большими и слишком малыми. Между размерами и информативной составляющей компонентов должны соблюдаться оптимальные соотношения.

**4.1.12** Проект пользовательского интерфейса детально обосновывается в соответствующем разделе «Проектирование графического пользовательского интерфейса средствами СОМ-объектов» пояснительной записки. Например,

Пользовательский интерфейс проекта программы декодирования штрих-кодов изображён на рисунке 7.

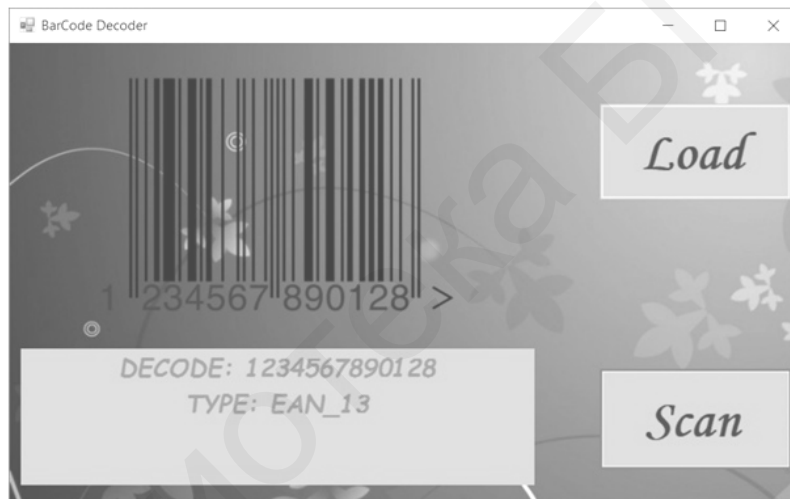


Рисунок 7 – Пользовательский интерфейс программы декодирования штрих-кодов

Он создан с помощью визуальной среды программирования Visual C# и использует тип взаимодействия, при котором инициатива управления принадлежит пользователю.

Интерфейс содержит две кнопки, которые пользователем могут применяться. Следовательно, программа содержит обработчик события «Нажатие кнопки Load» и «Нажатие кнопки Scan» с методами обработчиков

```
private void button1_Click()
и private void button2_Click()
```

соответственно. Также для взаимодействия с пользователем предусмотрены такие объекты, как «PictureBox» для загрузки и отображения изображения штрих-кода, «TextBox» для вывода на экран декодированного сообщения и «MessageBox» для уведомления пользователя об ошибках и помощи в использовании программы.

При проектировании графического пользовательского интерфейса выбраны холодные цвета для окна взаимодействия с пользователем, которые призваны снять усталость глаз и оказывать успокаивающее действие на пользователя. Шрифт приглушенного серого и синего цветов снижает мышечное напряжение, замедляет ритм дыхания, освежает, настраивает на терпеливую работу.

**4.1.13** В проектировании пользовательского и программного интерфейсов рекомендуется использование объектно-ориентированных технологий DCOM, .NET, OLE, OPC, ActiveX, DirectX, OpenGL, являющихся развитием COM-технологии.

Выпущенная в 1996 году технология DCOM позволяет COM-компонентам взаимодействовать друг с другом по сети.

В 2002 году была официально выпущена платформа Microsoft .NET, которая в настоящее время объявлена Microsoft рекомендуемой основой для создания приложений и компонентов под Windows. По этой причине в .NET включены и средства, позволяющие обращаться к компонентам COM из приложений .NET, и наоборот [18].

OLE (Objects Linking and Embedding – связывание и встраивание объектов) – технология, позволяющая передавать часть работы от одной программы редактирования к другой и возвращать результаты назад. Например, установленная на персональном компьютере издательская система может послать некий текст на обработку в текстовый редактор либо некоторое изображение в редактор изображений с помощью OLE-технологии [19].

OPC (OLE for Process Control) – семейство программных технологий, предоставляющих единый интерфейс для управления объектами автоматизации и технологическими процессами. Многие из OPC-протоколов базируются на Windows-технологиях: OLE, ActiveX, COM/DCOM. Такие OPC-протоколы, как OPC XML DA и OPC UA, являются платформо-независимыми.

## **4.2 Требования к оформлению линий чертежей**

В зависимости от назначения и типа схем в рамках курсовой работы по дисциплине «Основы программирования информационных систем» линиями изображаются функциональные и логические взаимосвязи, а также условные границы блоков схем и объектов диаграмм.

Оформлению линий чертежей и плакатов посвящён пункт 3.2 стандарта предприятия БГУИР СТП 01-2017 [2]. Но в нём, исходя из специфики курсовой работы по дисциплине «Основы программирования информационных систем», особое внимание необходимо обратить на подпункты 3.2.2–3.2.6 и 3.2.11.

## **4.3 Методика построения и требования к схеме алгоритма**

Следующий текст представляет акцентированную выдержку правил из пункта 3.12 «Правила выполнения схем алгоритмов, программ, данных и систем» стандарта СТП 01-2017 касательно только схем алгоритмов, минуя требования к схемам программ, данных и систем [2].

**4.3.1** Структура и состав схемы алгоритма программы определены единой системой программной документации (ЕСПД), частью которой является государственный стандарт ГОСТ 19.701-90 [20]. Согласно ГОСТ 19.701-90 схема алгоритма отображает последовательность выполнения операций для достижения поставленной цели.

**4.3.2** В состав схемы алгоритма входят символы, включающие линии потока информации и блоки начала и конца программы или функции, ввода-вывода данных, последовательного выполнения операций, ветвления (или условия), подготовки данных, начала и конца цикла, вызова внешней подпрограммы, соединителя, комментария.

Символы схемы алгоритма делятся на символы линий, данных, процесса, специальные символы. Их условные графические обозначения представлены в таблице 1.

**4.3.3** К линиям потока информации и линиям контуров условных графических обозначений блоков схемы алгоритма предъявляется требование, связанное с одинаковой толщиной линий.

**4.3.4** При переходе к УГО, расположенным в других местах схемы, используется УГО «Соединитель».

**4.3.5** Схемы алгоритмов выполняются с соблюдением пропорций размеров символов.

Таблица 1 – Условные графические обозначения символов схемы алгоритма

Символ	Наименование символа	Назначение
1	2	3
Символы данных		
	Данные	Ввод и вывод данных
Символы процесса		
	Процесс	Простое действие
	Предопределенный процесс	Комплексное действие в виде обращения к подпрограмме
	Подготовка	Указание диапазона действия параметра цикла
	Решение	Условие, ветвление
	Параллельные действия	
	Граница цикла	Обозначение границ тела цикла
Символы линий		
	Линия	Связь блоков и направление информационного потока
	Пунктирная линия	Используется в комментариях

Продолжение таблицы 1

1	2	3
Специальные символы		
	Соединитель	Связь удалённых друг от друга частей схемы
	Терминатор	Начало и конец алгоритма
	Комментарий	Пояснения к параметрам блока
	Пропуск	Пропуск части схемы

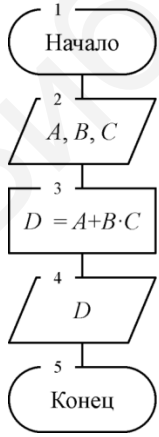
**4.3.6** Основное направление потока информации идёт сверху вниз и слева направо. В спорных местах направление на линиях указывается стрелками с развалом  $60^\circ$  [2, с. 47, рисунок 3.4, а].

**4.3.7** В основе процесса разработки схемы алгоритма лежит структурное программирование, заключающееся в разбиении (декомпозиции) алгоритма на множество таких структурных блоков, как последовательное действие, параллельное действие, условие, выбор (переключатель), цикл [21].

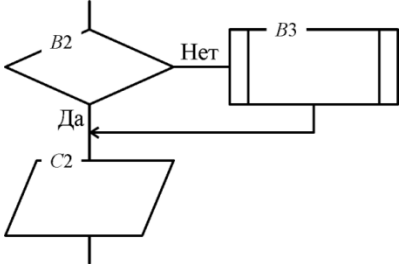
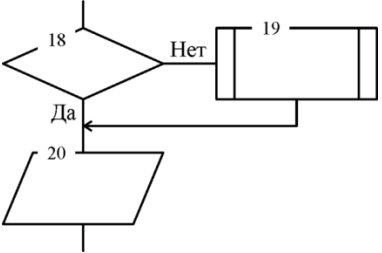
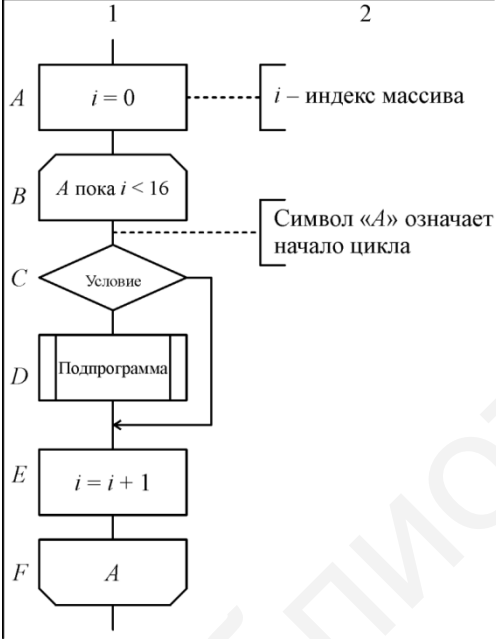
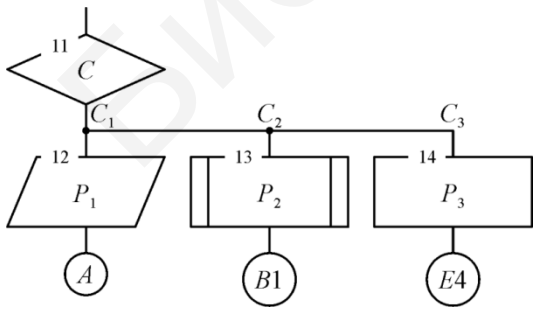
**4.3.8** Фрагменты схем, поясняющие применение структурных блоков, приведены в таблице 2 [22].

**4.3.9** Символы схемы алгоритма должны быть пронумерованы. Координаты зоны символа или порядковый номер проставляются в верхней его части в разрыве контура символа, как это реализовано в графах таблицы 2.

Таблица 2 – Фрагменты схем алгоритмов и структурные блоки

Фрагмент схемы	Содержание обозначения	Правила применения
1	2	3
	<p>Линейный алгоритм; последовательные действия</p>	<p>Координаты зоны символа или порядковый номер проставляются в верхней части символа в разрыве его контура</p>

Продолжение таблицы 2

1	2	3
	<p>Разветвляющийся алгоритм</p> <p>Возможные варианты обозначения символов в схемах:</p>	
	<p><math>B2, B3, C2</math> – координаты зоны листа, в которой размещён символ 18, 19, 20 – порядковые номера символов на схеме</p>	
	<p>Циклический алгоритм</p>	<p>Допускается не представлять координаты символов при наличии координатной сетки</p> <p>Комментарий применяется, если пояснение не помещается внутри символа (для пояснения характера параметров, особенностей процесса, линий потока и др.). Комментарий помещается в свободном месте схемы на данном листе и соединяется с поясняемым символом</p>
	<p>Разветвляющийся алгоритм с графическим изображением оператора «case» выбора (анализа признака <math>C</math>)</p> <p>Соединители:  <math>A, B1, E4</math> – идентификаторы соединителя в виде буквы при относительно простой схеме алгоритма или буквы и цифры, задающих координаты зоны листа</p>	<p>При большой насыщенности схемы символами отдельные линии потока между удалёнными друг от друга символами допускается обрывать. При этом в конце и в начале обрыва должен быть помещён символ «Соединитель»</p>

Продолжение таблицы 2

1	2	3
	Линии потока	Применяются для указания направления потока: без стрелки, если линия направлена слева направо или сверху вниз; со стрелкой – в остальных случаях
	Излом линии под углом 90°	Обозначает изменение направления потока
	Пересечение линий потока	Применяется в случае пересечения двух несвязанных потоков
	Слияние линий потока. Место слияния потоков обозначается точкой	Применяется в случае слияния линий потоков, каждая из которых направлена к одному и тому же символу на схеме

**4.3.10** В контексте основных типов алгоритмов (линейных, разветвляющихся и циклических) схема алгоритма в курсовой работе по дисциплине «Основы программирования информационных систем» должна иметь как минимум одно ветвление, т. е. относиться к разветвляющемуся или циклическому типу [22].

**4.3.11** Сложность схемы алгоритма определяется количеством символов, которое должно быть не менее 15, а рекомендуемое их количество варьируется в диапазоне от 15 до 30.

**4.3.12** Читательность и логическая завершенность схемы алгоритма заключается в возможности отслеживания путей изменений значений параметров, встречаемых в схеме, от момента их инициализации до завершения алгоритма. Иными словами, ни одна переменная, встречаемая в схеме алгоритма, не должна появляться «из ниоткуда» и существовать «сама по себе».

**4.3.13** Обоснование схемы алгоритма и деталильные пояснения к каждому её символу приводятся в подразделе «Разработка схемы алгоритма» раздела «Алгоритм функционирования программного обеспечения» пояснительной записки к курсовой работе.

В качестве примера схем алгоритмов и образца пояснений к ним ниже представлено решение задачи формирования образов текстовых символов при программировании бегущих строк на информационном табло прямоугольной

формы размером  $(M \times N)$  светодиодных матриц, каждая из которых состоит из  $(8 \times 8)$  светодиодов (иными словами, пикселей). То есть в общей сложности информационное табло имеет разрешение  $(8 \cdot M \times 8 \cdot N)$  светодиодов (пикселей).

Формирование образов текстовых символов занимает промежуточную позицию между операцией приёма текстовой информации, будучи отображённой на табло, и хранением её образа в видеопамати.

Текстовая информация, принимаемая контроллером табло от сервера, состоит из нескольких строк. Отдельная строка с точки зрения программирования является массивом символов. Внутреннее представление отдельного символа занимает 1 байт. В связи с требованием отображения на табло символов типов «знаки препинания», «разделители», «цифры», «латиница» и «кириллица», кодировка отдельного текстового символа, передаваемого сервером, выбирается в виде «Win1251 (cyrillic)». Строки текста, будучи отображаемого на информационном табло, стандартно разделяются символом «\r» с внутренним шестнадцатеричным кодом «0x0D» или двухсимвольной ESC-последовательностью «\r\n» с внутренними шестнадцатеричными кодами «0x0D, 0x0A» соответственно.

Предполагая ширину и высоту отдельного символа не более 8 пикселей таким образом, чтобы каждый символ не выходил за границы светодиодной матрицы размером  $(8 \times 8)$ , количество  $M$  текстовых строк информационного табло равно шести. Соответственно, со стороны сервера контроллером табло по интерфейсу Ethernet принимаются шесть строк текста. На общую длину текстовой информации накладывается ограничение, связанное со стандартной длиной TCP-пакета, равной 1500 байт. Следовательно, исходя из предположения вместимости текста, отображаемого на табло, в один TCP-пакет, длина отдельной строки не должна превышать  $Q = 1500/6 = 250$  символов. Каждую строку, согласно заданию на курсовой проект, контроллер табло способен визуализировать в бегущем виде.

Алгоритм бегущей строки в иерархии алгоритмов управления информационным табло, изображённой на рисунке 8, занимает более высокий уровень по отношению к алгоритму формирования образов текстовых символов.

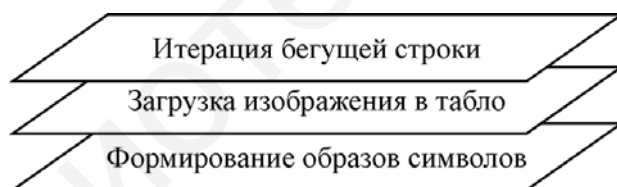


Рисунок 8 – Иерархия алгоритмов управления информационным табло

Для последующей реализации алгоритма бегущей строки в процедуре формирования образов символов необходимо учесть их горизонтальный сдвиг  $shifts[M]$  как влево, если  $shifts[i] > 0$ , так и вправо при  $shifts[i] < 0$  и наложение соседних символов в зависимости от заданного пробельного расстояния  $DISPLAY\_SPACE[M]$  между ними. Отдельные элементы массива  $DISPLAY\_SPACE[M]$  определяют степени разреженности или уплотнённости символов соответствующих строк. Посредством массива  $shifts[M]$  предусмотрена установка различных скоростей перемещения бегущих строк вплоть до их остановки или фиксации.

В качестве входных параметров данного алгоритма следует также предусмотреть координаты  $(x, y)$  расположения светодиодной матрицы  $(8 \times 8)$  в составе информационного табло. Начало координатной сетки  $(x, y)$  располагается в левом верхнем углу. Горизонтальная ось  $x$  направлена слева направо, координата  $x$  принимает целые значения от 0 до  $(N - 1)$ . Вертикальная ось  $y$  направлена сверху вниз, координата  $y$  принимает целые значения от 0 до  $(M - 1)$ .

Результирующая блок-схема алгоритма формирования образов символов на светодиодной матрице  $(8 \times 8)$  с координатами  $(x, y)$  представлена на рисунке 9.

В приведённом алгоритме массив `fonts[][8]` хранит образы-шрифты отдельных символов. Так, образ символа 'А' представляется последовательностью {56, 108, 198, 198, 254, 198, 198, 0} или в шестнадцатеричном виде {0x38, 0x6C, 0xC6, 0xC6, 0xFE, 0xC6, 0xC6, 0x00}, что визуально соответствует образу рисунка 10.

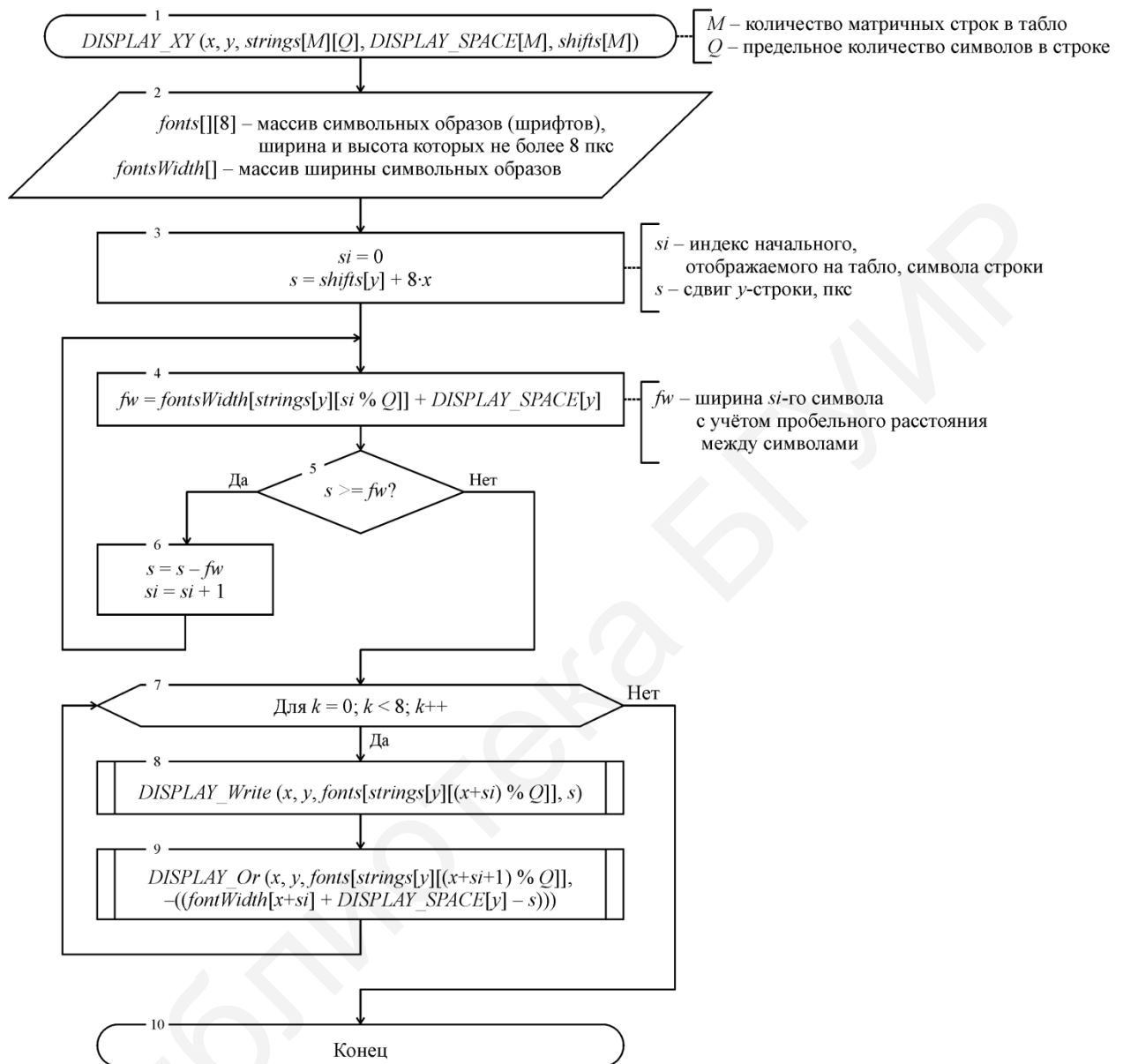


Рисунок 9 – Схема алгоритма формирования образов символов на светодиодной матрице

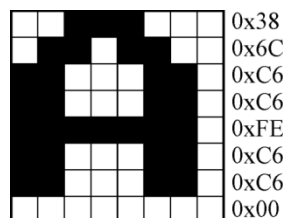


Рисунок 10 – Образ символа 'А'

Ниже следует фрагмент массива `fonts[][8]` шрифтов для символов кириллицы 'Б', 'В':

```
const unsigned char fonts[][8] = {
/*--- Шрифты символов кириллицы, расположенных по алфавиту ---*/
/* символ 'Б' */
/* символ 'В' */
```



```

0xFC, // 11111100
0xC0, // 11000000
0xF8, // 11111000
0xCC, // 11001100
0xCC, // 11001100 1
0xCC, // 11001100
0xF8, // 11111000
0x00, // 00000000

```

```

0xF8, // 11111000
0xCC, // 11001100
0xCC, // 11001100
0xF8, // 11111000
0xCC, // 11001100 2
0xCC, // 11001100
0xF8, // 11111000
0x00, // 00000000 };

```

Массив *fontsWidth[]* рисунка 9 содержит значения ширины образа отдельного символа. Так, символ 'А' кириллицы имеет ширину 7 пикселей, символы 'Б' и 'В' – 6 пикселей. Наибольшей шириной в 8 пикселей наделён символ 'Ц', наименьшей – в 2 пикселя – символ ','.

Процедуры *DISPLAY\_Write()* и *DISPLAY\_Or()* отвечают соответственно за отображение основного символа в светодиодной матрице с координатами (x, y) и за наложение следующего за ним символа строки с учётом вычисленного в блоках 5 и 6 сдвига *s* (влево при  $s > 0$  и вправо при  $s < 0$ ) и заданного фиксированного пробельного интервала *DISPLAY\_SPACE[y]* между символами. Графические схемы процедур *DISPLAY\_Write()* и *DISPLAY\_Or()* представлены на рисунке 11.

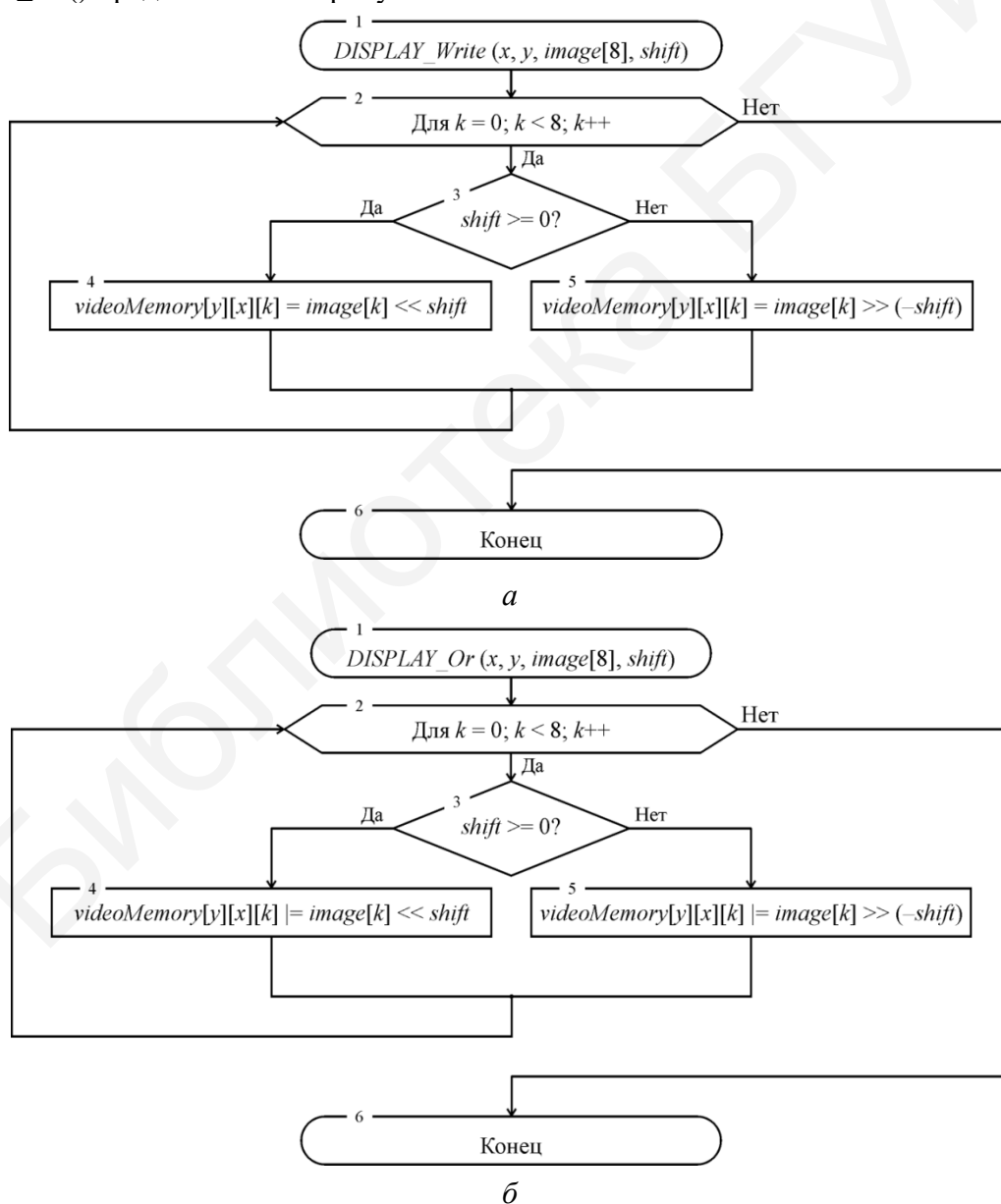


Рисунок 11 – Схемы алгоритмов процедур *DISPLAY\_Write()* и *DISPLAY\_Or()*

Операция '%' на рисунке 9 – это целочисленная бинарная операция вычисления остатка от деления первого (левого) операнда на второй (правый), предназначенная для предотвращения выхода за пределы массива *strings[][]* отображаемых строк и создания эффекта повторяющегося (или кругового) движения бегущих строк.

Алгоритмы процедур *DISPLAY\_Write()* и *DISPLAY\_Or()* очень схожи между собой. Отличие составляет применение операции '|' поразрядного логического сложения в процедуре *DISPLAY\_Or()* в противовес простому присваиванию '=' в аналогичных строках блоков 4, 5 процедуры *DISPLAY\_Write()*. Операции '<<' и '>>' блоков 4 и 5 на рисунке 11, *a* и *b* означают сдвиг двоичного представления байта *videoMemory[M][N][8]* видеопамати информационного табло на *|shift|* разрядов влево и вправо соответственно.

С учётом вышеописанных алгоритмов схема формирования образов символов на информационном табло представится в виде рисунка 12.

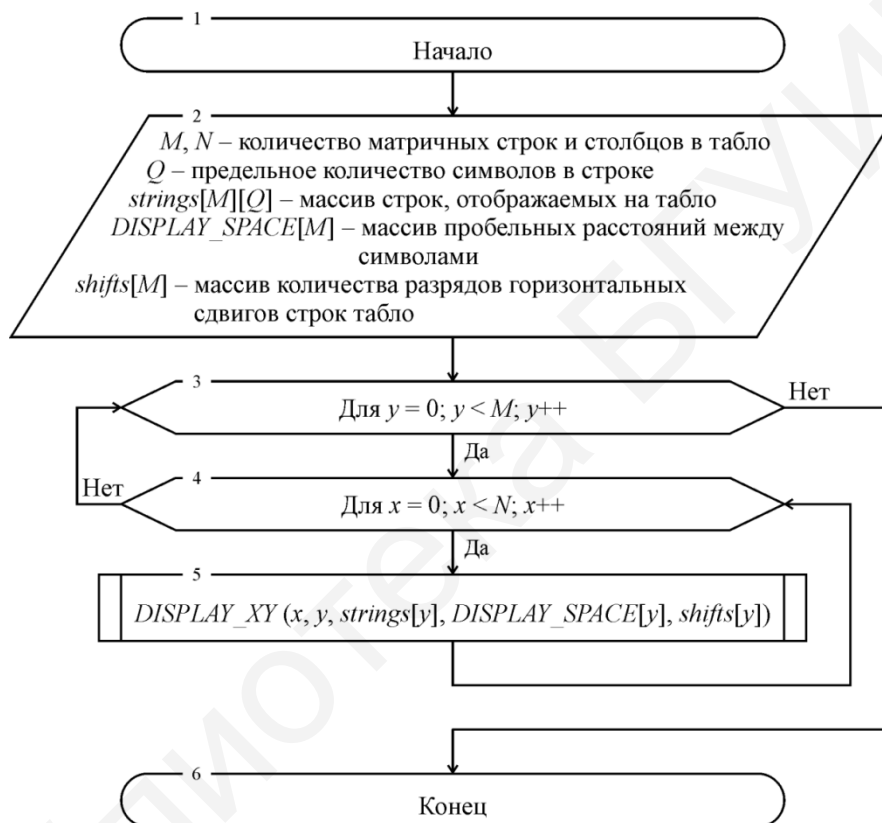


Рисунок 12 – Схема алгоритма формирования образов символов на информационном табло

Благодаря детальной декомпозиции алгоритм рисунка 12 формирования образов символов для всего информационного табло выглядит достаточно просто и заключается в циклическом проходе по всем светодиодным матрицам информационного табло и вызове базовой процедуры *DISPLAY\_XY()* формирования образов символов для отдельной светодиодной матрицы (*x, y*).

#### 4.4 Методика построения и требования к диаграмме состояний

**4.4.1** Комплект диаграмм состояний программы представляет собой модель поведения, позволяющую описать основные состояния программы, возможные переходы между ними в процессе функционирования, вызывающие данные переходы события, а также некоторые взаимодействия между программными объектами, возникающие при переходах [23, 24]. Такая модель от-

вечает основному назначению объектно-ориентированного программирования – создавать объекты, наделённые определённым поведением, которые будут адекватно представлять процессы предметной области.


**4.4.2** Основное назначение диаграммы состояний заключается в описании возможных последовательностей состояний и переходов, которые в совокупности характеризуют поведение программы в течение её жизненного цикла. Диаграмма состояний отражает динамическое поведение сущностей на основе спецификации их реакций на восприятие конкретных событий.

**4.4.3** Диаграмму состояний рекомендуется использовать для описания поведения некоторого объекта в нескольких различных вариантах использования. Интерфейсы и управляющие объекты классов, обладающие сложным поведением, также целесообразно представлять с помощью диаграммы состояний. Диаграмма состояний обычно не применяется при описании ряда взаимодействующих объектов, так как для этих целей используются специальные диаграммы взаимодействий.

**4.4.4** К оформлению диаграммы состояний, касательно толщины и формы линий, предъявляются требования, аналогичные требованиям к оформлению линий чертежей (см. подраздел 4.2) и символов схемы алгоритма (см. подраздел 4.3).

**4.4.5** В диаграммах состояний используются условные обозначения символов, перечисленные в таблице 3.

Таблица 3 – Условные графические обозначения символов диаграммы состояний

Символ	Наименование символа	Назначение
	Круг	Начальное состояние
	Окружность с кругом внутри	Конечное состояние
	Эллипс или окружность	Устойчивое (длительное) состояние
	Скруглённый прямоугольник	Кратковременное состояние
	Стрелка	Переход
	Толстая горизонтальная линия	Объединение или разветвление состояний

Верхушка скруглённого прямоугольника, обозначающего состояние, содержит название состояния. В середине может быть горизонтальная линия, под которой записываются активности, происходящие в данном состоянии.

Над стрелкой, обозначающей переход из одного состояния в другое, или рядом с ней отмечается название события (если есть), вызывающего переход. Вместе с названием события в формате «название\_события[охраняющее\_выражение]/действие» может записываться охраняющее выражение и действие, производимое при переходе.

Толстая горизонтальная линия с множеством входящих линий и одной выходящей либо одной входящей линией и множеством выходящих обозначает объединение и разветвление состояний соответственно.

**4.4.6** С состоянием можно связать данные пяти типов: входное действие, выходное действие, деятельность, событие и история состояния.

Для указания действий (actions), выполняемых при входе в состояние и при выходе из него, используются метки «entry» и «exit» соответственно. Действие, которое должно выполняться, когда программа находится в данном состоянии, указывается после метки «do». Считается, что такое действие начинается при входе в состояние и заканчивается при выходе из него (рисунок 13).

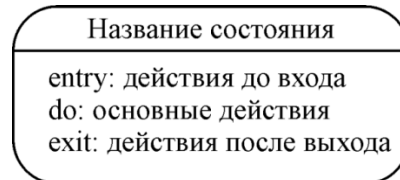


Рисунок 13 – Обозначение символа кратковременного состояния с действиями

Частные случаи состояний, начальное «start state» и конечное «end state», не содержат внутренних действий. В начальном состоянии программа находится по умолчанию в начальный момент времени. Графически оно обозначается в виде закрашенного кружка (см. таблицу 3). В конечном состоянии объект будет находиться по умолчанию после завершения работы программы в конечный момент времени. Графически оно обозначается в виде закрашенного кружка, помещенного в окружность.

**4.4.7** Между состояниями возможны различные типы переходов. Переход «transition» представляет собой отношение между двумя последовательными состояниями, которое указывает на факт безусловной смены одного состояния другим. Переходы между состояниями отображаются стрелками, как показано на рисунке 14.

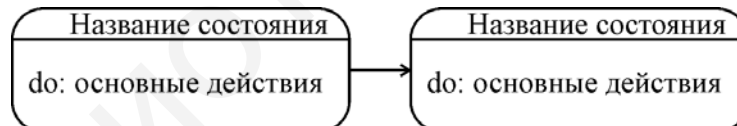


Рисунок 14 – Обозначение перехода между состояниями

**4.4.8** Обычно переход инициируется событием. Событие «event» представляет собой спецификацию некоторого факта, имеющего место в пространстве и во времени. Допускаются переходы без событий. Также разрешены условные переходы. Сторожевое (ограждающее) условие «guard condition», если оно есть, представляет собой булево выражение, для записи которого может использоваться синтаксис языка объектных ограничений. Порядок выполнения условного перехода следующий:

- 1) происходит событие;
- 2) вычисляется условие перехода;
- 3) если условие перехода истинно, выполняется переход и активизируется действие, в противном случае переход не выполняется.

**4.4.9** Выражение действия «action expression» выполняется только в том случае, когда переход срабатывает. Выполняемое действие не может быть пре-

рвано никаким другим действием, пока оно не закончится. Данное действие может оказывать влияние как на сам объект, так и на его окружение, если это с очевидностью следует из контекста модели.

**4.4.10** Одной из важных характеристик диаграммы состояний является подсостояние «substate». Подсостояние позволяет значительно упростить моделирование сложного поведения. Подсостояние – это состояние, вложенное в другое состояние. На рисунке 15 показано составное состояние «composite state», содержащее в себе два подсостояния.

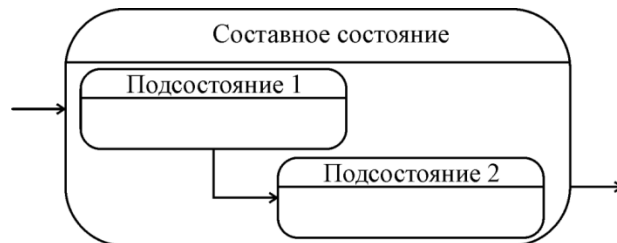


Рисунок 15 – Составное состояние с вложенными в него подсостояниями

В свою очередь в подсостояние могут вкладываться другие подсостояния. Степень вложенности подсостояний не ограничивается.

**4.4.11** Составное состояние может содержать несколько последовательных подсостояний или несколько параллельных. Последовательные подсостояния «sequential substates» используются для моделирования такого поведения программы, при котором в каждый момент времени она может находиться только в одном подсостоянии (рисунок 16). Поведение объекта в этом случае представляет собой последовательную смену подсостояний, начиная начальным и заканчивая конечным подсостояниями.

**4.4.12** Возможно наличие параллельных подсостояний «concurrent substates», позволяющих специфицировать несколько программных подпроцессов, которые могут выполняться параллельно внутри составного состояния. Каждый из подпроцессов занимает некоторую область внутри составного состояния, которая отделяется от остальных горизонтальной пунктирной линией, как показано на рисунке 16. Если на диаграмме состояний имеется составное состояние с вложенными параллельными подсостояниями, то программа может одновременно находиться в каждом из этих подсостояний.

**4.4.13** Сложность диаграммы состояний определяется количеством символов, которое должно быть не менее десяти, а рекомендуемое их количество варьируется в диапазоне от 10 до 20.

**4.4.14** Читабельность и логическая завершённость диаграммы состояний заключается в возможности отслеживания путей изменений значений параметров, встречаемых в диаграмме, от момента их инициализации до завершения работы программы. Иными словами, ни одна переменная, используемая диаграммой, не должна появляться «из ниоткуда» и существовать «сама по себе».

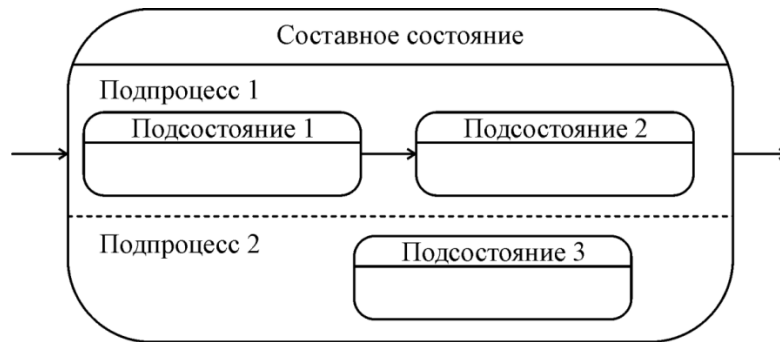


Рисунок 16 – Составное состояние с вложенными последовательными и параллельными подсостояниями

**4.4.15** Обоснование диаграммы состояний и детальные пояснения к каждому её символу и переходу приводятся в подразделе «Разработка диаграммы состояний» раздела «Алгоритм функционирования программного обеспечения» пояснительной записки к курсовой работе.

В качестве примера далее рассматривается диаграмма состояний и приводятся к ней пояснения при решении задачи организации бегущих строк на информационном табло прямоугольной формы размером  $(M \times N)$  светодиодных матриц. Данная задача связана с задачей формирования образов текстовых символов при программировании бегущих строк из предыдущего пункта.

Важной особенностью алгоритма управления информационным табло является возможность организации множества  $M$  бегущих строк с различными скоростями. При этом следует учесть факт, что стоячая строка является частным случаем бегущей строки со значением сдвига, равным нулю.

Место процедуры организации бегущих строк отражено в диаграмме состояний программы управления информационным табло на рисунке 17.

Она основывается на следующих двух алгоритмах:

- формирования в видеопамяти *videoMemory[]* образов текстовых символов информационных строк *strings[]* в зависимости от элементов массива *shifts[]* их соответствующих сдвигов;
- загрузки изображения из видеопамяти *videoMemory[]* в табло.

Данные алгоритмы работают на каждой итерации бесконечного цикла управления информационным табло.

В блоке 1 инициализации информационного табло рисунка 17 проводится чтение из карты памяти начальной строковой информации, определение массивов *strings[][]* бегущих строк и *delta[]* их скоростей, инициализация видеопамяти *videoMemory[]* и очистка табло.

Основная работа контроллера табло осуществляется по событиям таймера. При этом с «периодом обращения к серверу», равным порядка 30 с, контроллер связывается с информационным сервером с целью получения новых данных в текстовом формате и обновления массива *strings[][]* строк. На рисунке 17 это действие представлено составным состоянием «Загрузка информации», включающим два последовательных подсостояния «Связь с сервером» и «Обновление текста». В подсостоянии «Связь с сервером» проводится загрузка новой строковой информации с сервера, а в «Обновлении текста» – обновление строкового массива *strings[][]*.

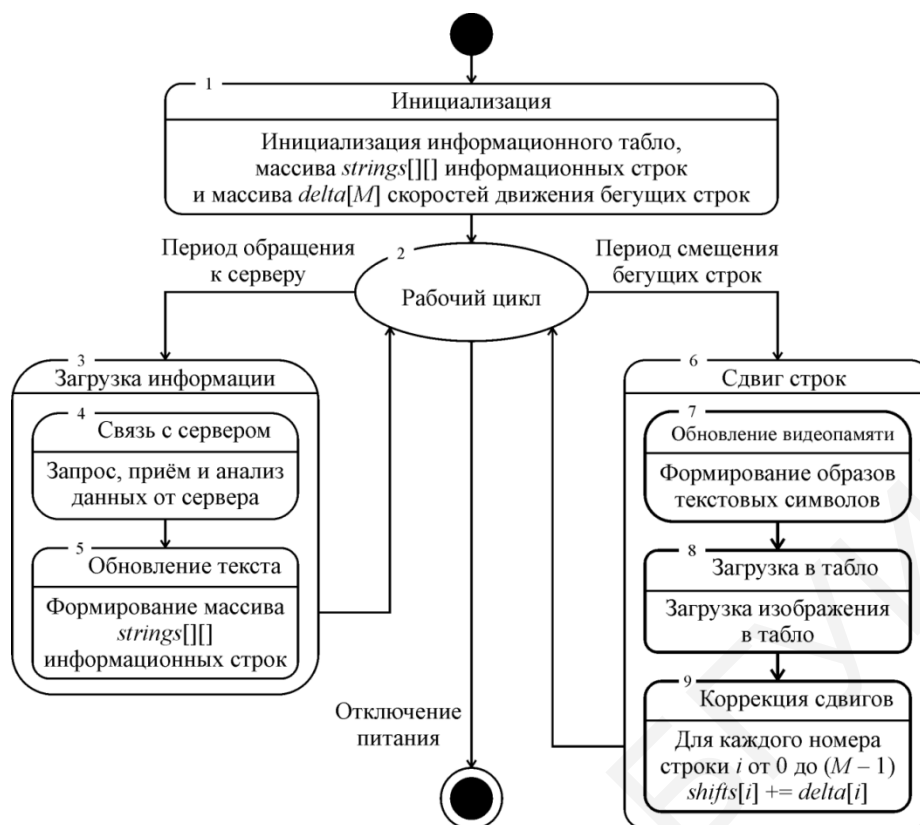


Рисунок 17 – Диаграмма состояний программы организации бегущих строк

Короткий «период смещения бегущих строк», примерно равный 10 мс, отвечает за очередной сдвиг строк табло, реализуя таким образом эффект движения. Сдвиг строк реализуется посредством этапов обновления видеопамати *videoMemory[]* (блок 7 рисунка 17) в зависимости от элементов сдвигового массива *shifts[]* с последующим изменением элементов данного массива на значения *delta[]* (блок 9 рисунка 17). Обновлённый образ в видеопамати, согласно блоку 8, загружается в табло и отображается пользователю.

## 4.5 Методика построения и требования к диаграмме последовательности

**4.5.1** Диаграмма последовательности, являющаяся видом диаграммы взаимодействия, предназначена для моделирования взаимодействия программных объектов во времени и для представления временных особенностей передачи и приёма сообщений между объектами [23, 25].

**4.5.2** Взаимодействующие объекты программы обмениваются между собой некоторой информацией. При этом информация принимает форму законченных сообщений. Таким образом, сообщение, имеющее информационное содержание, приобретает дополнительное свойство оказывать направленное влияние на своего получателя.

**4.5.3** Диаграмма последовательности выделяет упорядочение сообщений по времени, отражает сценарий поведения в программе и обеспечивает наглядное представление порядка передачи сообщений. Посредством диаграммы последовательности также раскрываются динамические аспекты программы. Они

показывают взаимодействие, включающее набор объектов и их отношений, и пересылаемые между объектами сообщения.

**4.5.4** Элементами диаграммы последовательности являются объекты, связи и сообщения.

**4.5.5** На диаграмме последовательности объекты, участвующие во взаимодействии, помещаются на вершине диаграммы, вдоль горизонтальной оси. Слева размещается объект, инициирующий взаимодействие, а справа – объекты по возрастанию подчинённости. Сообщения, посылаемые и принимаемые объектами, помещаются вдоль вертикальной оси в порядке возрастания времени от вершины к основанию диаграммы. Таким образом, обеспечивается визуальное представление потока управления во времени.

**4.5.6** К оформлению диаграммы последовательности, касательно толщины и формы линий, предъявляются требования, аналогичные требованиям к оформлению линий чертежей (см. подраздел 4.2) и символов схемы алгоритма (см. подраздел 4.3).

**4.5.7** В диаграммах последовательности используются условные обозначения символов, перечисленные в таблице 4.

Таблица 4 – Условные графические обозначения символов диаграммы последовательности

Символ	Наименование символа	Назначение
	Прямоугольник с именем объекта внутри	Обозначение объекта программы
	Вертикальная пунктирная линия	Линия жизни объекта
	Высокий тонкий прямоугольник	Фокус управления
	Стрелка синхронного сообщения	Связь как канал для передачи синхронных сообщений
	Стрелка асинхронного сообщения	Связь как канал для передачи асинхронных сообщений
	Пунктирная стрелка	Маркер сообщения возврата в момент потери фокуса объекта
	Линия	Абстрактный канал передачи сообщения
	Перекрестие	Уничтожение объекта

**4.5.8** Синтаксис представления имени имеет вид

*ИмяОбъекта : ИмяКласса*

**4.5.9** Диаграмма последовательности имеет две важные характеристики: линию жизни объекта и фокус управления.

Линия жизни объекта – это вертикальная пунктирная линия, которая обозначает период существования объекта. Большинство объектов существуют на протяжении всего взаимодействия, их линии жизни тянутся от вершины до ос-



нования диаграммы. Объекты могут создаваться в ходе взаимодействия. В этом случае их линии жизни начинаются с момента приёма сообщения «create». Если объекты уничтожаются в ходе взаимодействия, их линии жизни заканчиваются с момента приёма сообщения «destroy» (рисунок 18).

Фокус управления – это высокий тонкий прямоугольник, отображающий период времени, в течение которого объект выполняет действие (свою или подчинённую процедуру). Вершина прямоугольника отмечает начало действия, а основание – его завершение. Момент завершения может маркироваться сообщением возврата, которое отображается пунктирной стрелкой (см. таблицу 4).

Фокусы управления могут быть вложены друг в друга, например при рекурсивном вызове собственной операции. В этом случае второй фокус управления отображается немного правее первого, как показано на рисунке 19.



Рисунок 18 – Создание и уничтожение объекта Рисунок 19 – Вложение фокусов управления

**4.5.10** Для отображения «условности» линия жизни может быть разделена на несколько параллельных линий жизни. Каждая отдельная линия соответствует условному ветвлению во взаимодействии. Далее в некоторой точке линии жизни могут быть снова соединены (рисунок 20).

**4.5.11** Ветвление показывается множеством стрелок, идущих из одной точки. Каждая стрелка отмечается сторожевым условием, подобно рисунку 21.

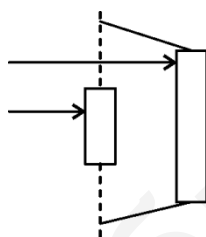


Рисунок 20 – Параллельные линии жизни

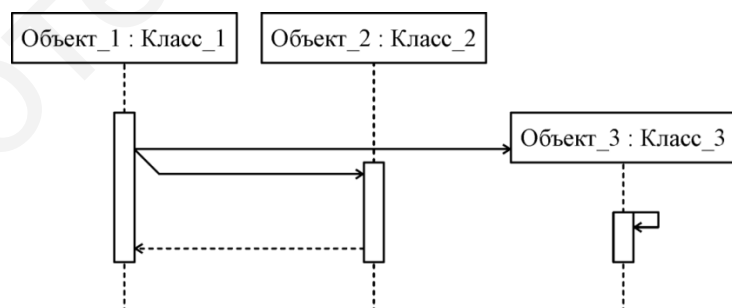


Рисунок 21 – Ветвление потока управления на диаграмме последовательности

**4.5.12** На диаграмме последовательности взаимодействие объектов друг с другом изображается в виде связей как каналов для передачи (или путей для пересылки) сообщений.

Сообщение – это спецификация передачи информации между объектами в ожидании того, что будет обеспечена требуемая деятельность. Приём сообщения рассматривается как событие. Кроме того, объект может посылать сообщение самому себе – рефлексивное сообщение.

Для полной записи сообщений принят следующий синтаксис:

*ВозвращаемаяВеличина := ИмяСообщения (Аргументы)*

Результатом обработки сообщения обычно является действие. Моделируются следующие разновидности действий:

- вызов – в объекте запускается операция;
- возврат – при возврате значения в вызывающий объект;
- посылка «send» сигнала в объект;
- создание объекта по стандартному сообщению «create>»;
- уничтожение объекта выполняется по сообщению «destroy».

**4.5.13** Когда объект посылает сообщение другому объекту (делегируя некоторое действие получателю), объект-получатель, в свою очередь, может послать сообщение в третий объект и т. д. Таким образом формируется поток сообщений, т. е. последовательность управления. Сообщения в последовательности должны быть пронумерованы. Номера записываются перед сообщениями, которые размещаются над линиями связей. Направления сообщений указываются стрелками.

**4.5.14** Наиболее общую форму управления задаёт процедурный или вложенный поток (поток синхронных сообщений). Процедурный поток изображается стрелками с заполненными наконечниками (см. таблицу 4). Работа с синхронным сообщением подчиняется следующему правилу: передатчик ожидает ответа от получателя до тех пор, пока получатель не примет и не обработает сообщение.

**4.5.15** Менее общую форму управления задаёт асинхронный поток управления. Он изображается обычными стрелками с развалом  $60^\circ$  (см. таблицу 4). Сообщения считаются асинхронными, если передатчик не ожидает реакции от получателя сообщения. Получатель принимает сообщения по мере готовности.

**4.5.16** Помимо рассмотренных линейных потоков управления, в диаграммах последовательностей моделируются и более сложные формы: итерации и ветвления.

Итерация представляет повторяющуюся последовательность сообщений. После номера сообщения итерации добавляется выражение

$$*[i:=1..n],$$

означающее, что сообщение итерации будет повторяться заданное количество ( $n$ ) раз.

Для моделирования ветвления после номера сообщения добавляется выражение условия, например  $[x > 0]$ . Сообщение альтернативной ветви помечается таким же номером, но с другим условием:  $[x \leq 0]$ .

**4.5.17** Для формирования диаграммы последовательности выполняются следующие действия:

- отображаются объекты, которые участвуют во взаимодействии;
- устанавливаются связи этих объектов;
- связи помечаются сообщениями, которые посылают и получают выделенные объекты.

В итоге формируется ясное визуальное представление потока управления в контексте структурной организации сотрудничающих программных объектов.

**4.5.18** Сложность диаграммы последовательности определяется количеством символов, которое должно быть не менее 10, а рекомендуемое их количество варьируется в диапазоне от 10 до 20.

**4.5.19** Обоснование диаграммы последовательности и деталильные пояснения к каждому её символу и сообщению приводятся в подразделе «Разработка диаграммы последовательности» раздела «Алгоритм функционирования программного обеспечения» пояснительной записки к курсовой работе (см. приложение А).

В качестве примера далее рассматривается диаграмма последовательности программы сканирования штрих-кодов (см. также рисунок 7) и приводятся к ней пояснения.

Судя по диаграмме рисунка 22, видно, что в программе присутствует один субъект и три объекта взаимодействия, причём инициатива управления принадлежит пользователю. Пользователь может загружать изображение с диска и запускать процесс декодирования с помощью сканера штрихового кода.

Сканер штрих-кода – это основной объект программы, подразумевающий обработку изображения и непосредственной области штрих-кода на изображении. Чтобы перейти к работе напрямую со штрих-кодом, сканер обрабатывает изображение, выделяя область локализации штрихового кода. Затем происходит распознавание и декодирование штрих-кода. Декодированное сообщение «Текст штрих-кода» доставляется пользователю. Этот цикл продолжается до тех пор, пока пользователь не прекратит работу с программой.

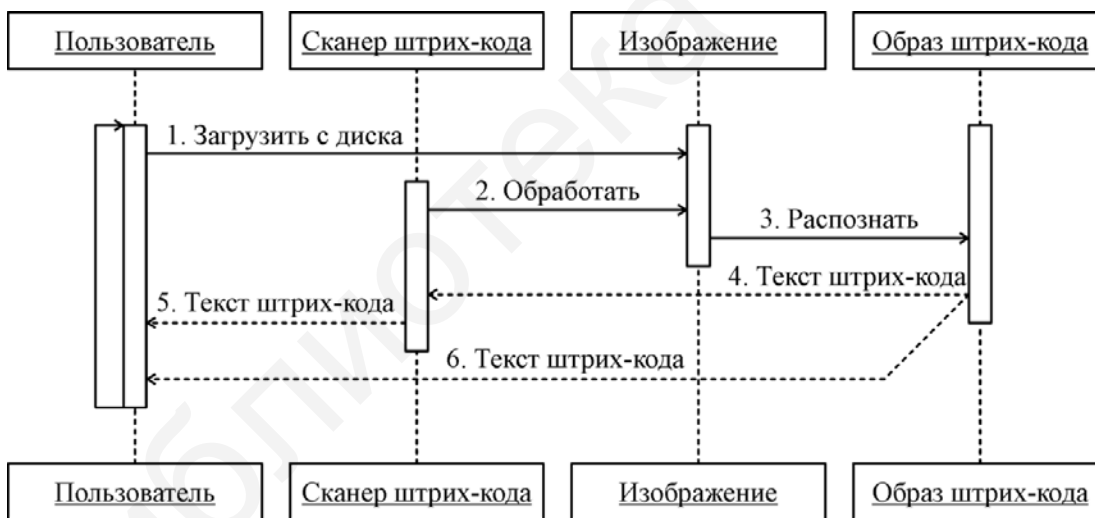


Рисунок 22 – Диаграмма последовательности программы сканирования штрих-кодов

Сканер штрих-кода в основном использует асинхронные сообщения, что обусловлено его взаимодействием с изображением и образом штрих-кода, которые, в свою очередь, являются объектами, не предусматривающими выполнения каких-либо действий, – действия выполняются над ними только пользователем и сканером.

## 5 ДОПОЛНИТЕЛЬНЫЕ УКАЗАНИЯ ПО ТЕМАМ КУРСОВОЙ РАБОТЫ

Тематика курсовой работы по дисциплине «Основы программирования информационных систем» обусловлена как узким, так и широким понятиями «программирования».

В узком смысле слова программирование представляет реализацию алгоритма действий для решения поставленной задачи средствами конкретных механизмов, коими, как правило, выступают компьютер и иные технические системы, управляемые с помощью специального языка – языка программирования. В подобном «программировании» изучаются структуры данных, форматы их представления в памяти и хранения на информационных носителях, кодирование, шифрование, интерфейсы связи и протоколы взаимодействия в системах обмена информацией.

Под программированием в широком смысле слова понимается не просто реализация некоторого алгоритма решения задачи в контексте заданного языка программирования, не просто объединение данных и команд, манипулирующих данными, а прикладное применение методов решения алгоритмических задач в планировании, прогнозировании, оптимизации. Не случайно в синонимическом ряду к слову «программирование» стоят «план», «содержание», «пакет», «проект», а термин «план-программа» является основным в системе планирования и прогнозирования [26, 27]. Даже на программирование как простейшее оперирование данными могут налагаться условия наискорейшего достижения результата или нахождения наиболее короткого пути реализации алгоритма.

Таким образом, в рамках дисциплины «Основы программирования информационных систем» «программирование» неразрывно связано с форматами хранения структур данных, их оперированием, преобразованием и анализом, планированием и прогнозированием, способами обмена данными в каналах связи информационных систем, а также с дополнительными условиями, налагаемыми на действия, приводящие к результату решения поставленных задач.

В связи с вышесказанным темы курсовой работы группируются по следующим направлениям: статистический анализ массивов данных, регрессионный анализ массивов данных, спектральный анализ массивов, корреляционный анализ массивов, логический анализ данных интерфейсов связи, протоколы систем телемеханики и телеметрии, формирование и преобразование массивов, линейное программирование, динамическое программирование.

Независимо от темы курсовой работы, в структуре пояснительной записки выделяются следующие основные разделы (см. приложение А): математическое обоснование и формализация задачи курсовой работы; структуры и способы кодирования и представления данных, форматы файлов хранения входных и выходных массивов; применение технологий программирования, в особенности объектно-ориентированных при решении задачи курсовой работы; обоснование проекта графического пользовательского интерфейса; описание алгоритма ра-

боты программы; демонстрация работы программы с учётом входных массивов разных типов; анализ результатов работы программы.

Листинг программного кода приводится в приложении к пояснительной записке. Касательно непосредственной программной части, вне зависимости от темы курсовой работы, к ней, относительно корректности вводимых пользователем данных, предъявляются требования предотвращения сопутствующих ошибок пользователя и обработки всевозможных исключений операционной системы.

Далее приводятся пояснения и детализация структуры курсовой работы в зависимости от тематического направления. Приведённые пояснения рассматриваются совместно с образцовым заданием приложения А, в котором пункты 4.3–4.7 являются общими для всех тем курсовой работы, а в пункте 4.7 «Анализ результатов решения поставленной задачи» сосредотачиваются вербальные выводы автора-студента, полученные в результате сравнительного анализа работы его программы над исходным материалом различной природы.

## **5.1 Статистический анализ массивов данных**

Из полного перечня предлагаемых в первом разделе тем к статистическому анализу массивов данных относятся темы №7 и 8.

**5.1.1** В контексте тем №7 и 8 предполагается рассмотрение массивов случайных чисел, которые являются предметом изучения дисциплины «Теория вероятностей и математическая статистика». Обе темы оперируют «гистограммой» как практическим аналогом «функции распределения» из теории вероятностей. Поэтому в работе над упомянутыми темами необходимо определиться с понятием «гистограммы», освоить способы генерации массивов детерминированных и случайных чисел, а также алгоритмы построения гистограмм [28].

**5.1.2** В рамках темы №7 в качестве пункта «Описание к выполнению» задания по курсовой работе выступает «Проектирование графического пользовательского интерфейса для анализа гистограмм изображений, хранимых в файлах и генерируемых функционально». Исходя из этого, дополнительно к «гистограмме» и алгоритмам её вычисления необходимо определиться с понятием «изображения», рассмотреть используемые цветовые системы и модели представления изображений, в том числе растровые и векторные, форматы графических файлов, в частности наиболее распространённые bmp, jpeg, gif, png, способы работы с ними в программных средах, алгоритмы построения покомпонентных и цветовых гистограмм.

Пункт 4.1 задания по теме №7 звучит как «Математическое описание алгоритмов формирования векторных изображений». В его содержание входит рассмотрение следующих вопросов: понятие изображения, принципы формирования изображений, цветовые системы RGB, YCbCr и MKO, анализ литературных источников по математическому обоснованию алгоритмов формирования векторных изображений.

Векторное изображение как матрица, распределение яркости и/или форма контуров в котором заданы математически, является частным случаем детерминированного массива. Оператор «векторизации» незаменим в задачах моделирования сцен, распознавания образов и идентификации объектов. Поэтому в статистическом анализе существенную роль играет анализ формы гистограмм векторных изображений и их принципиальные отличия от гистограмм растровых изображений.

Пункт 4.2 «Форматы представления и хранения файлов изображений» задания включает вопросы анализа литературных источников по описанию форматов хранения растровых и векторных изображений, а также функциям работы с файлами изображений в выбранной среде программирования.

Пункт 4.3 непосредственно связан с алгоритмами построения гистограмм изображений и включает вопросы анализа литературных источников и определения гистограммы изображения, физического смысла гистограммы изображения, анализа алгоритмов построения покомпонентных гистограмм изображения и алгоритмов построения гистограммы изображения на плоскости цветового треугольника системы МКО.

Таким образом, программа статистического анализа гистограмм изображений должна выполнять функции открытия файла одного из графических форматов bmp, jpeg, gif, png или wmf, отображения изображения на экране, вычисления и построения покомпонентных R, G, B, Y и цветовой гистограмм считанного из файла изображения, предотвращения сопутствующих ошибок пользователя и обработки исключений операционной системы.

Пункт «Анализ результатов решения поставленной задачи» содержит вербальные выводы автора-студента, полученные в результате сравнительного анализа гистограмм множества изображений различных типов (векторных и растровых; «портрет», «пейзаж», «текст» или «сцена») по форме, шероховатости и полимодальности гистограмм.

## **5.2 Регрессионный анализ массивов данных**

К данному направлению, кроме основной темы №9, можно отнести и тему №3, основанную на операции интерполяции двухмерных массивов, тесно связанной с аппроксимацией и регрессией.

**5.2.1** В теме №9 пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для регрессионного анализа одномерных массивов данных с возможностью задания полного пути к файлу исходного одномерного массива, функциональной зависимости тренда и функции распределения случайных величин, аддитивно добавляемых к функции тренда, а также подбора вида и параметров функции тренда в процессе непосредственного анализа». Данное описание к выполнению отчасти перекликается с аналогичным описанием по теме №8 «Программа анализа гистограмм одномерных массивов данных», поэтому в темах №8 и 9 схожими будут и пункты задания. В частности, пункты 4.1 и 4.2 за-

дания по теме №9 идентичны соответствующим пунктам темы №8. Основное отличие текущей темы от темы №8 состоит в изучении методов аппроксимации и интерполяции данных одномерного массива.

К пользовательскому интерфейсу темы №9 «Программа регрессионного анализа одномерных массивов данных» предъявляются следующие требования: выбор пользователем использования файлов с готовыми массивами данных, представленных в бинарном, текстовом, wav или xls форматах; возможность выбора пользователем вида (линейного, параболического, эллиптического, гиперболического, экспоненциального, показательного, степенного, логарифмического, гармонического) детерминированной функции как функции тренда; ввод параметров детерминированной функции с соответствующими подсказками, определяющими смысл параметров; выбор пользователем вида функции распределения случайного массива; ввод параметров функции распределения с соответствующими подсказками, определяющими смысл параметров; выбор способа добавления (аддитивно или мультипликативно) случайной составляющей к детерминированному массиву; ввод весовых коэффициентов детерминированной и случайной составляющих в результирующем массиве; аппроксимация результирующего массива функциями вышеперечисленных типов (линейный, параболический, эллиптический, гиперболический и т. д.) методами наименьших квадратов, наибольшего правдоподобия, согласования и др.; отображение всех аппроксимирующих кривых в одном графическом окне с явно выделенным графиком исходно сгенерированного массива; выбор наиболее подходящей аппроксимирующей функции по минимальной оценке среднеквадратичного отклонения аппроксимирующего массива от сгенерированного.

Дополнительно рекомендуется обратиться к постановке задачи и требованиям, касающимся темы №8 в плане генерации детерминированных и случайных массивов.

**5.2.2** Тема №3 «Программа масштабирования изображений» отличается от темы №9 главным образом предметом анализа и обработки. В качестве предмета темы №3 выступают изображения как двухмерные массивы. Объектом служат методы аппроксимации и интерполяции двухмерных массивов и их фрагментов.

Пункт «Описание к выполнению» задания по теме №3 формулируется в виде «Проектирование графического пользовательского интерфейса с возможностью задания коэффициента масштабирования (уменьшения или увеличения) изображения в диапазоне от 10 до 1000 %». Иными словами, в рамках курсовой работы по теме №3 необходимо изучить методы уменьшения и увеличения изображений с минимальными потерями их качества, в частности методы билинейной и бикубической интерполяции.

В контексте темы №3 функциональные требования, предъявляемые к пользовательскому интерфейсу программы, выглядят следующим образом: возможность выбора пользователем файла исходного изображения; отображение исходного (выбранного) изображения на экране компьютера; задание для

выбранного изображения коэффициента масштаба в диапазоне от 10 до 1000 % с шагом 10 %; визуализация результирующего масштабированного изображения; сохранение масштабированного изображения в файле одного из графических форматов bmp, jpeg, png, gif.

Кроме перечисленных выше требований в рамках темы №3 необходимо разработать авторскую функцию (метода класса) с открытым кодом использования математических операторов для интерполяции двухмерной матрицы изображения.

### **5.3 Спектральный анализ массивов данных**

К направлению спектрального анализа и спектральной обработки массивов относятся темы №1, 2, 39, 42, 43, 45.

**5.3.1** Базовым оператором спектрального анализа является преобразование Фурье (одномерное или двухмерное), а для цифровых сигналов или массивов данных – дискретное преобразование Фурье [29]. В контексте спектрального анализа массив выступает в качестве «выборки», движущейся вдоль «контейнера» как потока данных, причём на каждом шаге формирования выборки вычисляется спектр в виде пары амплитудно-частотной (АЧХ) и фазочастотной (ФЧХ) характеристик. Изменение АЧХ во времени представляет спектрограмму как двухмерный массив или изображение. Поэтому в освоении спектрального анализа первоочередными терминами выступают «амплитудно-частотная характеристика», «фазочастотная характеристика», «амплитудно-фазочастотная характеристика (АФЧХ)», «спектрограмма».

**5.3.2** В теме №1 «Программа спектрального анализа массива случайных чисел» пункт «Описание к выполнению» задания по курсовой работе формулируется как «Проектирование графического пользовательского интерфейса с возможностью задания функции распределения, ввода её основных параметров, генерирования случайного цифрового сигнала в виде массива случайных чисел, построения и анализа его амплитудно-частотной (АЧХ), фазочастотной (ФЧХ) и амплитудно-фазочастотной характеристик, а также гистограмм АЧХ и ФЧХ». Данная тема, согласно описанию, перекликается с темами №7, 9 в плане методов генерации случайных массивов с заданными законами распределения (равномерный, нормальный, экспоненциальный, треугольный, параболический, Пуассона, Максвелла, Больцмана, Коши, Рэлея) и алгоритмов вычисления гистограмм. Поэтому в работе над темой №1 рекомендуется отчасти обратиться и к требованиям по темам №7 и 9, в особенности задуматься над требованиями, касающимися вычисления гистограмм двухмерных массивов.

Требования к программе по теме №1: выбор пользователем вида функции распределения массива случайных чисел (равномерная, нормальная, экспоненциальная, треугольная, параболическая, Пуассона, Максвелла, Больцмана, Коши, Рэлея); ввод параметров функции распределения с соответствующими подсказками, определяющими смысл параметров; отображение сгенерированного



исходного массива в виде графика; построение АЧХ, ФЧХ и АФЧХ исходного массива; построение гистограммы АЧХ, ФЧХ и АФЧХ.

**5.3.3** Базовым отличием тем с номерами 39, 42, 43 от темы №1 и основным их предметом выступает понятие «спектрограмма» – изображение, отражающее изменение АЧХ потока данных во времени. Так как спектрограммы представляют собой изображения, то в работе с ними применяются методы и алгоритмы обработки, анализа и преобразования изображений [30, 31].

В теме №39 «Программа качественного анализа аудиофайлов с использованием спектрограмм» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для качественного анализа аудиофайла с возможностью задания полного пути к аудиофайлу, его проигрыванию, анимации перемещения сигнальной выборки по аудиофайлу, визуализации изменения спектрограммы и вывода на экране показателей качества текущего фрагмента аудиофайла».

Согласно описанию к выполнению, далее следуют требования, предъявляемые к пользовательскому интерфейсу программы по теме №39: возможность выбора пути к аудиофайлу; визуализация в РСМ-формате сигналов аудиоканалов выбранного файла; возможность задания начальной и конечной координат фрагмента сигнала одного из аудиоканалов для последующего построения его спектрограммы; проигрывание выделенного фрагмента аудиосигнала; задание длины выборки для построения спектрограммы; вычисление текущих мгновенных качественных показателей спектрограммы – граничной частоты в АЧХ; координат глобального максимума на изображении спектрограммы; координат глобального максимума на АЧХ; координат локальных максимумов на текущей АЧХ; метрических характеристик (энергетического и геометрического центров) спектрограммы; построение и визуализация спектрограммы ранее выделенного пользователем фрагмента с отображением на её кадрах номера текущего смещения выборки относительно начала файла и параметров спектрограммы, выделением разными цветами кривых, точки которых соединяют граничные частоты и глобальные максимумы АЧХ отдельных выборок, энергетических и геометрических центров; возможность сохранения спектрограммы в видеофайле.

#### **5.4 Корреляционный анализ массивов данных**

Корреляционный анализ необходим для поиска подобных или эталонных фигур и сигналов в контейнере [29, 32, 33]. В качестве контейнера выступает массив с большой длиной или изображение с идеально однородным фоном, с градиентным фоном или с шумом. Поиск «эталона» заключается в последовательном перемещении его образа вдоль контейнера «навстречу» потоку, т. е. слева направо в одномерном случае и в направлении слева направо, сверху вниз в случае двумерной матрицы, например изображения. При этом на каждой итерации сдвига эталона относительно контейнера вычисляется скалярное произведение «эталонного» вектора с соответствующим вектором как фрагментом

контейнера с размерами, равными размерам вектора эталона. Если фрагмент контейнера совпадает с эталоном, их скалярное произведение имеет максимальное значение.

Основная цель курсовой работы по корреляционному анализу данных состоит в освоении способов практического применения дискретной корреляционной функции на эталонных объектах относительно простой формы и исследовании характера изменения корреляционной зависимости от аффинных преобразований эталонной фигуры.

Весомую практическую значимость имеет понятие «знаковой корреляции». Поэтому в качестве выводов по курсовой работе данного направления выступают, в частности, зависимости чувствительности знаковой корреляции от аффинных преобразований.

Из полного перечня предлагаемых тем с корреляционным анализом данных связаны темы №5, 6, 41, 44.

**5.4.1** В теме №5 «Программа корреляционного анализа векторных изображений», кроме «корреляции», рассматриваются «векторные изображения», также упоминавшиеся в теме №7. Поэтому вопросы и требования, касающиеся темы №5, перекликаются с темой №7, и для их детализации рекомендуется обратиться к пункту 5.1.2.

В теме №5 «Программа корреляционного анализа векторных изображений» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для корреляционного анализа двух векторных изображений, одно из которых (исходное) хранится на диске, а второе генерируется функционально, с возможностью задания полного пути к файлу исходного изображения, коэффициента масштабирования, углов поворотов и функционального типа поверхности распределения яркости отдельных цветовых составляющих генерируемого векторного изображения».

В требования к пользовательскому интерфейсу по теме №5 входят: возможность выбора пользователем изображения «векторного» типа плоской и объёмной фигур: прямоугольника, ромба, параллелограмма, трапеции, эллипса, кардиоиды, трилистника, конуса, параболоида, сферы; ввод параметров функции распределения яркости и формы контуров для генерируемого векторного изображения с соответствующими подсказками, определяющими смысл параметров; возможность выбора типа фона (однородный или градиентный) «векторного» изображения; настройка аффинных преобразований путём задания начального положения фигуры, её масштаба и угла поворота в плоскости изображения примерно так, как показано на рисунке 5; визуализация сгенерированного векторного изображения; возможность ввода полного пути к файлу эталонного изображения или иного способа выбора «эталона»; вычисление корреляционной и знаковой корреляционной функций сгенерированного векторного изображения с «эталоном» с их отображением в двух интерпретациях: в виде графиков поверхности и в виде линий контуров.

**5.4.2** Корреляционный анализ одномерных массивов, которому посвящена тема №6 «Программа корреляционного анализа одномерных массивов данных», представляет собой достаточно простой случай. В качестве одномерных эталонов в теме №6 выступают сигналы следующих форм: прямоугольной, треугольной симметричной, треугольной асимметричной, трапецеидальной симметричной, трапецеидальной асимметричной, параболической, эллиптической, колоколообразной симметричной, колоколообразной асимметричной. Для полноты картины учитывается зашумлённость контейнера.

В теме №6 «Программа корреляционного анализа одномерных массивов данных» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для корреляционного анализа двух одномерных сигналов с возможностью задания их типа (детерминированный или случайный), вида (для детерминированного сигнала) или функции распределения (для случайного сигнала), и коэффициента масштаба».

Требования к пользовательскому интерфейсу по теме №6 перекликаются с требованиями к программе по теме №5 (см. пункт 5.4.1): задание длины  $N$  массива эталона или, иными словами, длины одномерного массива-окна, перемещающегося вдоль контейнера; возможность выбора пользователем формы импульса (прямоугольной, треугольной симметричной, треугольной асимметричной, трапецеидальной симметричной, трапецеидальной асимметричной, параболической, эллиптической, колоколообразной симметричной, колоколообразной асимметричной) как одномерного массива длиной  $N$ , составляющего генерируемый контейнер; ввод параметров функции импульса (в частности, ширины и высоты) с отображением соответствующих подсказок, определяющих смысл параметров; настройка аффинных преобразований путём задания начального положения импульса и его масштаба в генерируемом контейнере; визуализация предварительно сгенерированного контейнерного сигнала; выбор пользователем вида функции распределения массива случайных чисел длиной, совпадающей с длиной контейнера; ввод параметров функции распределения с соответствующими подсказками, определяющими смысл параметров; выбор способа добавления (аддитивно или мультипликативно) случайной составляющей к контейнеру; ввод весовых коэффициентов предварительно сгенерированного контейнерного сигнала и случайной составляющей в результирующем массиве-контейнере; визуализация результирующего сгенерированного контейнерного сигнала; возможность создания эталона на базе прямоугольной, треугольной симметричной, треугольной асимметричной, трапецеидальной симметричной, трапецеидальной асимметричной, параболической, эллиптической, колоколообразной симметричной и колоколообразной асимметричной форм с настройкой параметров функции для соответствующей формы; вычисление корреляционной и знаковой корреляционной функций сгенерированного контейнера с эталоном, с отображением их в виде графиков и информированием пользователя об их локальных максимумах.

## 5.5 Логический анализ данных интерфейсов связи

Логический анализ изучает методы получения информации путём декодирования сигналов, передаваемых по каналам связи информационных систем, и структуру кодов, зависящую от интерфейса. Интерфейс как совокупность средств и методов взаимодействия между элементами системы имеет физические (или аппаратные) и логические (или программные) уровни, которые, в свою очередь, согласно модели OSI взаимодействия открытых систем, «раскладываются», как правило, в общем на семь подуровней [34].

В непосредственной передаче информации по каналу связи участвует физический уровень интерфейса, иначе именуемый физическим интерфейсом. Биты информации, кодируемые уровнем напряжения (высокий/низкий), типом фронта (передний/задний), длительностью, способом NRZI, в физическом интерфейсе объединяются в посылки или пакеты с определённым форматом.

В данном подразделе анализируются сигналы и коды битов для физических интерфейсов стандартов USB, IrDA, CAN, I2C, SPI, USART/UART, соответствующие им форматы коротких посылок, текстовых сообщений небольшой длины и видеофайлов.

Из полного перечня тем к логическому анализу относятся темы №17, 18, 19, 20, 23, 25. Темы №17, 18, 23 и 25 выступают как «декодеры» форматов данных и файлов, передаваемых по каналам связи информационных систем. Темы №19 и 20 посвящены декодированию сигналов коротких фрагментов (транзакций, пакетов и посылок) информационных сообщений.

**5.5.1** Интерфейс IrDA применяется в системах дистанционного беспроводного управления на базе инфракрасного канала связи.

В теме №19 «Программа декодирования осциллограмм интерфейса IrDA» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для декодирования хранящейся в заданном пользователем файле осциллограммы пакета данных интерфейса связи IrDA».

Осциллограмма декодируемого сигнала хранится в текстовом файле, в формате двумерной матрицы вещественных чисел вида

0.01232006	0
0.01240006	-3.0859375

где первый столбец означает текущее время, измеряемое в секундах, второй – отвечает за величину мгновенного напряжения. Для приведения исходного, считанного из текстового файла, сигнала к идеализированному импульсному виду необходима его пороговая обработка с предварительным вычислением порогового напряжения.

Требования к реализации пользовательского интерфейса по теме №19: возможность выбора пользователем и загрузки с диска осциллограммы исходного сигнала; визуализация загруженной осциллограммы исходного сигнала; автоматическая пороговая обработка исходного сигнала; совместное отображение исходного и порогово обработанного сигналов; возможность повышения

разрешения графиков сигналов путём визуального масштабирования (растягивания) осциллограмм; распознавание стандарта (RC или SIRC) инфракрасной связи, которому принадлежит выбранный сигнал-пакет; вывод декодированных данных и основных параметров (длительности импульсов и целого пакета) сигнала в поле редактирования.

**5.5.2** По аналогии с предыдущей темой №19 в теме №20 «Программа декодирования осциллограмм интерфейса USB» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для декодирования осциллограммы связи интерфейса USB, хранящейся в текстовом виде в заданном пользователем файле».

Осциллограммы сигналов USB-пакетов и транзакций, наподобие темы №19, хранятся в текстовом файле в формате двумерной вещественной матрицы.

Требования к программе и пользовательскому интерфейсу по теме №20 в основном аналогичны требованиям по теме №19, за исключением пунктов распознавания отдельных пакетов в USB-транзакции, декодирования пакетов в соответствии с их форматами, вывода в окне редактирования двоичных декодированных данных транзакции (в виде 0000000110100101...), типов пакетов (например, SOF, IN, OUT, SETUP) и их полей, основных временных параметров (длительности импульсов и отдельных пакетов) исходного сигнала.

**5.5.3** Тема №23 «Программа отображения кадров цифрового болометра» располагается на границе логического анализа и протоколов систем телемеханики и телеметрии. В ней требуется разобраться с форматом передачи видеоданных по физическому интерфейсу SPI (VoSPI, Video over SPI).

В теме №23 пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для отображения кадров цифрового болометра, хранящихся в текстовом виде «PLAIN TEXT» в формате VoSPI в заданном пользователем файле».

Исходные файлы по данной теме хранят в формате «PLAIN TEXT», последовательность изображений такая, в которой они представлены в памяти микропроцессора после считывания из цифрового болометра. Иными словами, в исходных файлах хранится копия памяти микропроцессора с кадрами-теплогограммами, полученными от цифрового болометра как инфракрасной (тепловой) камеры или прибора ночного видения.

Требования к реализации пользовательского интерфейса по теме №23: возможность выбора пользователем и загрузки с диска файла, представляющего последовательность кадров цифрового болометра в формате «PLAIN TEXT»; подсчёт количества кадров в файле; возможность выбора номера кадра для визуализации; выбор цветовой палитры: полутоновой, «colormap» или «rainbow» для стандартного представления тепловизионных изображений [35]; визуализация выбранного кадра в соответствии с указанной цветовой палитрой.

Примерный пользовательский интерфейс по данной теме изображён на рисунке 23.

**5.5.4** Тема №25 «Программа построения дерева Хаффмана по данным JPEG-файла», как и тема №23, граничит с «протоколами систем телемеханики и телеметрии». Она применяется, например, в формате MPEG-2 передачи кадров по каналам Ethernet, WiFi и Bluetooth в системах видеонаблюдения и видеоконтроля. В данной теме необходимо разобраться с форматом JPEG хранения изображений, в частности, с местом расположения в JPEG таблицы Хаффмана и алгоритмом взаимного её преобразования в дерево Хаффмана, применяемого в кодировании битового потока квантованного, дискретно-косинусно-преобразованного образа фрагмента исходного изображения, представленного в системе YCbCr.

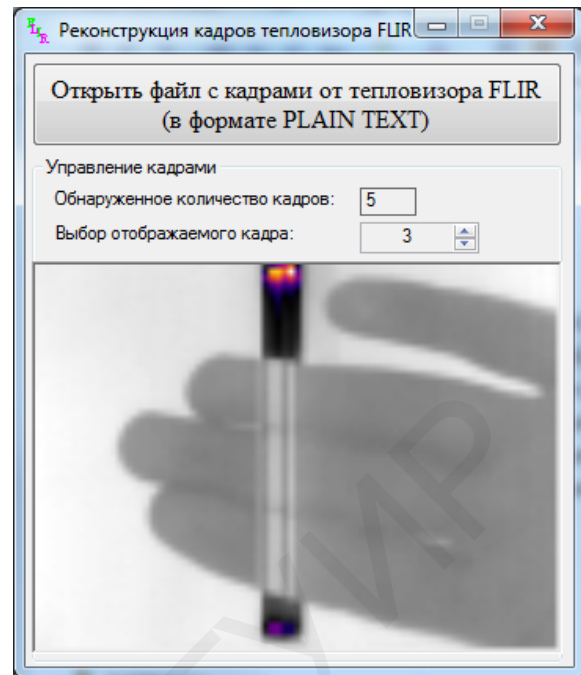


Рисунок 23 – Примерный пользовательский интерфейс программы отображения кадров цифрового болометра

В теме №25 пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для визуализации деревьев Хаффмана, хранящихся в заданном JPEG-файле».

Так, для таблицы Хаффмана «FFC4 0018 00 00030101000000000000000000000000 0301000204» соответствующее дерево должно выглядеть, как показано на рисунке 24.

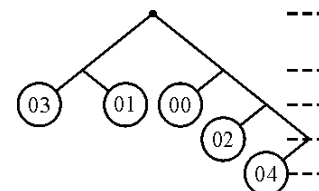


Рисунок 24 – Пример дерева Хаффмана

Требования к программе и реализации пользовательского интерфейса по теме №25 звучат следующим образом: возможность выбора пользователем и загрузки с диска JPEG-файла; анализ выбранного пользователем файла на наличие таблиц Хаффмана; отображение в статическом окне количества найденных таблиц Хаффмана; возможность задания номера интересующей таблицы Хаффмана с отображением её исходных шестнадцатеричных кодов в поле редактирования; наличие кнопки запуска построения дерева Хаффмана для заданной пользователем таблицы; возможность указания полного пути и имени файла для хранения изображения результирующего дерева; индикация процесса построения дерева Хаффмана посредством бегущего индикатора; формирование текстового файла (с тем же именем, но с расширением «txt»), и расположенного по тому же пути, что и только что построенное изображение дерева) со сгенерированными двоичными кодами вершин заданного пользователем дерева Хаффмана, например, в виде «00 – 03\r\n01 – 01\r\n10 – 00\r\n110 – 02\r\n1110 – 04» для дерева на рисунке 24.

## 5.6 Протоколы систем телемеханики и телеметрии

Подобные протоколы используются для передачи измерительной информации, текстовых и видеосообщений в сетевых системах сбора и анализа данных (другими словами, в системах телемеханики и телеметрии). В модели OSI эти протоколы представлены канальным, сетевым, транспортным и сеансовым уровнями, а также уровнем представления данных.

Основной целью выполнения курсовой работы по данному направлению является изучение, приобретение и укоренение навыков работы с инструментарием, поддерживающим сетевые телеметрические протоколы, и разработки программ взаимодействия с ним (инструментарием, включающим специализированные приложения и утилиты) в автоматическом режиме. В качестве подобных протоколов рассматриваются TCP, UDP, MQTT, MODBUS TCP, MODBUS RTU, МЭК-104.

Из предлагаемого перечня к темам, связанным с протоколами систем телемеханики и телеметрии, относятся №24, 32, 33, 34, 35, 36, 37.

**5.6.1** В теме №24 «Программа обмена текстовыми сообщениями средствами Ethernet» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для обмена текстовыми сообщениями между клиентами сети Ethernet по заданному протоколу».

Требования к программе по теме №24 выглядят примерно так: организация TCP-сервера как концентратора и коммутатора сетевых сообщений с минимальным пользовательским интерфейсом и портом №24 для прослушивания; организация независимых TCP-клиентов, подключающихся к серверу по порту №24; возможность ввода имени клиента в поле редактирования при его запуске; наличие линейного или комбинированного списка информирования присутствующих в сети клиентов, с возможностью выбора адресуемого клиента; возможность ввода в поле редактирования текстового сообщения для адресуемого клиента; наличие окна отображения текстового диалога с выбранным клиентом; возможность загрузки с сервера истории диалога для вновь выбранного клиента.

**5.6.2** В теме №33 «Программа обмена текстовыми сообщениями по протоколу MQTT» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для обмена текстовыми сообщениями между клиентами сети Ethernet по протоколу MQTT».

Протокол MQTT находит широкое применение в технологии «интернет-вещей». Под MQTT подразумевается не только протокол взаимодействия поверх, например, TCP, но и архитектура во главе с «брокером». Поэтому для освоения MQTT необходимо в первую очередь изучить его архитектуру и идеологию «подписчик – публицист». Кроме этого, следует провести аналитический обзор имеющегося множества MQTT-брокеров, например: RSMB, Mosquitto, Amazon, выявить их отличия и научиться их конфигурировать.

Требования к программе MQTT-клиента и пользовательскому интерфейсу по теме №33: возможность задания параметров MQTT-сессии: идентификатора клиента, тайм-аута бездействия, качества сервиса, периода пингования, аутентификации и др.; связь по TCP с RSMB; наличие поля редактирования для ввода темы подписки, совпадающей с идентификатором клиента; наличие поля для ввода текстового сообщения и кнопки для его отправки; наличие окна диалога с переданными и принятыми сообщениями.

### **5.7 Формирование и преобразование массивов**

К направлению «формирования и преобразования массивов» относятся темы №4, 15, 16, 21, 22, 26, 27, 40, связанные с генерацией массивов, например векторных аудиофайлов и изображений и их фильтрацией или сжатием.

Фильтрация и JPEG-компрессия массивов, в связи с преобразованием их спектров, имеет общие черты со спектральным анализом. Задания по генерации файлов также тесно связаны со спектральным и корреляционным анализом, где требуется формирование детерминированных и случайных массивов.

**5.7.1** В теме №4 «Программа фильтрации изображений» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса с возможностью выбора типа фильтра изображения».

В фильтрации изображений применяются как линейные, так и нелинейные (рангово-порядковые) двумерные фильтры [33]. Базовыми операциями в процессе фильтрации выступают тесно связанные между собой «свёртка» и «корреляция». Поэтому, в изучении фильтрации основное внимание уделяется освоению алгоритмов вычисления двумерной свёртки и корреляции (см. пункт 4.1.8).

В требования к программе по теме №4 входят: возможность выбора пользователем файла исходного изображения; визуализация исходного изображения; выбор типа фильтра (линейный, сглаживающий, медианный) и задание его параметров (импульсной характеристики или весовых коэффициентов) с отображением соответствующих подсказок и пояснений; применение настроенного пользователем фильтра с визуализацией результирующего отфильтрованного изображения в отдельном окне.

**5.7.2** В теме №15 «Программа генерации векторных аудиофайлов» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для генерации аудиопотока с функционально заданной спектрограммой».

Согласно описанию предметом и объектом работы выступают спектрограмма в виде функционально задаваемого изображения и обратное преобразование Фурье. Сопутствующим инструментарием для освоения темы являются информация о форматах аудиофайлов (в частности, wav) и функции для работы с ними (в частности, сохранения в файловой системе).

Примерные требования к функциональности программы и пользовательскому интерфейсу по теме №15 формулируются следующим образом: возмож-



ность выбора пользователем функции задания спектрограммы как изображений изменения во времени амплитудно-частотной и фазочастотной характеристик; настройка параметров выбранной функции с отображением соответствующих подсказок и пояснений; визуализация спектрограммы в виде контрастированных изображений изменения во времени АЧХ и ФЧХ; генерация wav-файла по заданной спектрограмме.

## **5.8 Линейное программирование**

Линейное программирование представляет математическую дисциплину, посвящённую теории и методам решения экстремальных задач на множествах векторов многомерного пространства, задаваемых системами линейных уравнений и неравенств [36, 37].

Из полного перечня предлагаемых тем к линейному программированию отнесены темы №10, 28, 29, 38, в основном (за исключением темы №38) связанные с методами решения транспортной задачи как задачи оптимального распределения товаров и услуг между поставщиками и потребителями.

Методы линейного программирования, эффективные в решении задач из области исследования операций, используют преобразования массивов, интерпретируемых в виде таблиц. Поэтому общей чертой всех перечисленных выше тем является изучение форматов хранения таблиц в памяти компьютера и в файловой системе.

**5.8.1** В теме №10 «Программа решения транспортной задачи симплекс-методом» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для решения транспортной задачи симплекс-методом с возможностью задания полного пути к файлу таблицы (матрицы) перевозок».

В описании к данной теме фигурирует термин «матрицы перевозок», который будет упоминаться в заданиях и требованиях к остальным темам, связанным с транспортной задачей.

Требования к программе по теме №10: ввод количества поставщиков и потребителей, возможность ручного ввода и коррекции матрицы перевозок, представленной в виде таблицы, возможность загрузки матрицы перевозок из указанного пользователем текстового (txt) или xls-файла, ввод матрицы стоимостей перевозок (матрицы затрат), расчёт наименьшей стоимости перевозки симплекс-методом, отображение хода решения задачи в отдельном окне.

**5.8.2** В теме №28 «Программа решения транспортной задачи методом потенциалов» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для решения транспортной задачи методом потенциалов с возможностью задания полного пути к файлу таблицы (матрицы) перевозок».

Требования к программе по теме №28 также совпадают с соответствующими требованиями по теме №10, за исключением метода решения задачи.

## 5.9 Динамическое программирование

С динамическим программированием как разделом математики, решающим задачи планирования производственных многоэтапных процессов, связаны темы №11, 12, 13, 14, 30, 31 [38, 39].

**5.9.1** В теме №11 «Программа прокладки оптимального пути» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для решения задачи выбора оптимального пути между назначенными пунктами в транспортной системе, представляющей определённую топологическую схему дорожных участков, перекрёстков и развязок с заданными среднестатистическими временами задержек или простоев». Исходный материал, а именно топология дорог для данной курсовой работы, заимствован из литературного источника [38] и в незначительном модифицированном виде представлен на рисунке 25.

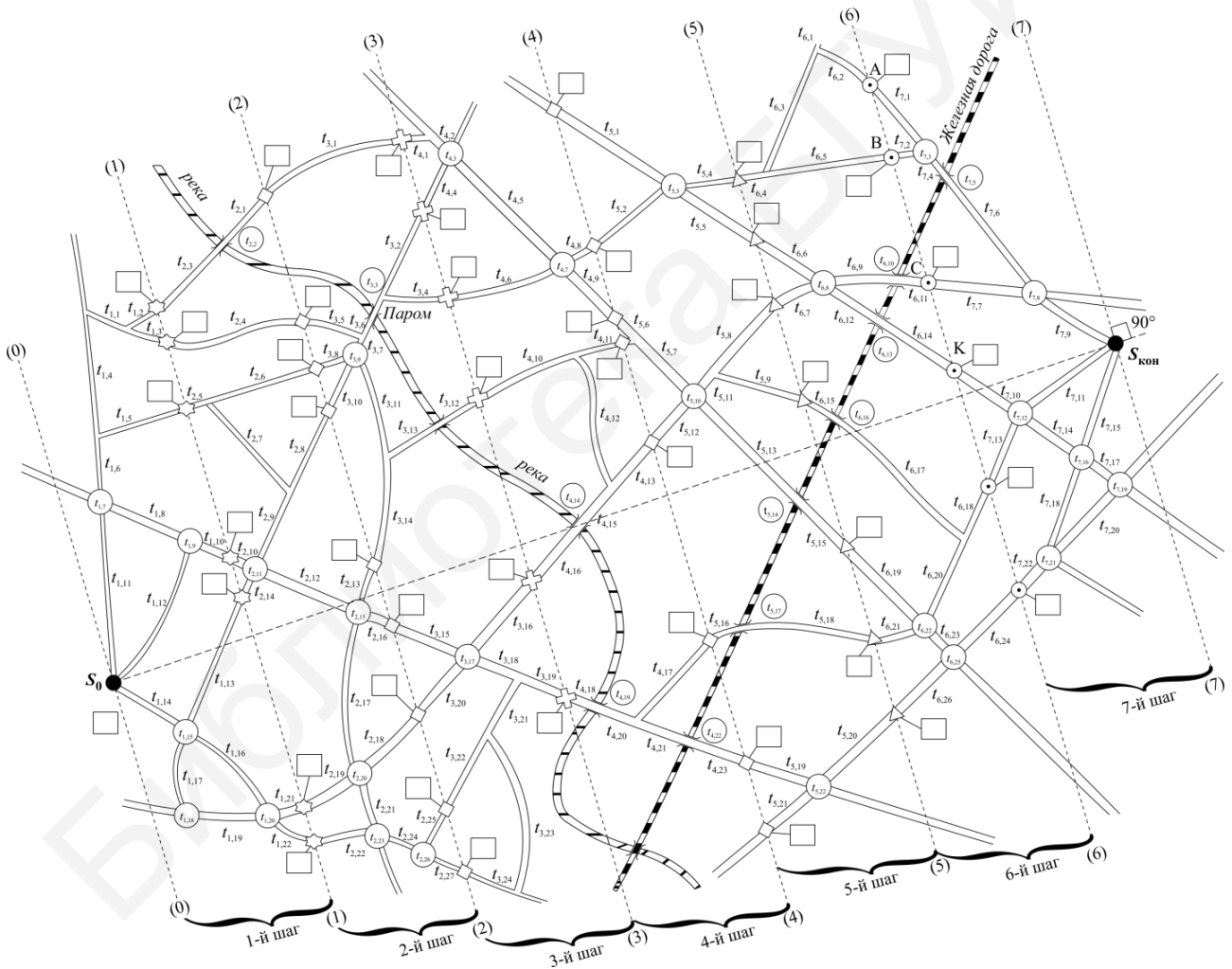


Рисунок 25 – Топология дорог для задачи по теме №11

Более детальная постановка задачи «звучит» следующим образом.

Для задачи автомобильной грузоперевозки методом динамического программирования (динамического планирования) необходимо вычислить наименее затратный по времени путь из пункта  $S_0$  в пункт  $S_{\text{кон}}$  с представленной на рисунке 25 топологией дорог.

Времена, затрачиваемые на каждом отрезке автомобильных дорог и при простое на перекрёстках или в очередях, обозначены как  $t_{ij}$ , где  $i = 1, 2, \dots, 7$  – номер шага или этапа планирования,  $j$  – номер обозначенного на топологии участка дороги для  $i$ -го шага. Время проезда отрезка пути указано на рисунке рядом с интересующим отрезком, а время, затрачиваемое при простое на перекрёстке или в очереди к переправе или переезду, указано в кружке рядом с интересующим перекрёстком, переездом или переправой.

Числовые значения времён  $t_{ij}$  (в минутах) задаются таблично в пользовательском интерфейсе программы в формате таблицы 5.

Таблица 5 – Распределение времени (в минутах) проезда отрезков дорог, перекрёстков, переездов или переправ для рисунка 25

Номер отрезка, $j$	Номер шага, $i$						
	1	2	3	4	5	6	7
1	4	4	15	1	10	2	10
2	2	5	8	1	8	2	2
3	2	6	20	3	5	15	2
4	10	14	6	3	3	2	1
5	5	1	3	11	7	10	5
6	1	10	2	10	3	5	10
7	4	11	1	3	5	2	4
8	1	10	2	2	4	1	2
9	2	8	5	3	8	3	5
10	20	2	3	12	3	5	3
11	16	1	10	3	1	1	4
12	11	8	3	14	4	2	3
13	6	3	3	3	10	5	4
14	1	3	11	2	5	5	2
15	10	2	4	2	4	3	6
16	8	1	4	3	3	5	2
17	3	18	2	10	5	12	1
18	5	5	2	1	10	4	5
19	2	4	4	5	3	4	2
20	2	2	4	2	4	5	3
21	6	5	5	3	2	3	1
22	4	7	10	5	5	3	1
23		2	20	4		1	
24		4	3			4	
25		8				2	
26		1				2	
27		6					

Требования к пользовательскому интерфейсу и программе по теме №11 выглядят следующим образом: визуализация изображения топологии дорог; возможность ввода и коррекции пользователем вручную табличного распределения временных параметров движения на заданной топологии дорог; загрузка и чтение матрицы распределения временных параметров движения из указанного пользователем бинарного или текстового (формата «PLAIN TEXT»), или Excel-файла; отображение значений введённых пользователем времён на топо-

логической карте; автоматическое, т. е. программное заполнение результирующих времён в узловых точках изображения топологии; вычисление наименее затратного по времени пути с выводом результата в отдельном окне и подчёркиванием результирующего пути на исходном изображении заданной топологии.

**5.9.2** В теме №12 «Программа прогнозирования динамики финансово-экономической деятельности предприятия» пункт «Описание к выполнению» задания по курсовой работе выглядит как «Проектирование графического пользовательского интерфейса для анализа финансово-экономических операций, инвестиций или контрактов предприятия с возможностью задания общего количества лет инвестиционного проекта, годового распределения доходов, ставки сравнения, расчёта чистого приведённого дохода, внутренней нормы доходности, срока окупаемости и рентабельности инвестиционного проекта» [40].

Объектом и предметом курсовой работы по теме №12 являются соответственно модель трёхлетнего инвестиционного контракта и математические и программные средства моделирования инвестиционных процессов.

Объектом инвестирования выступает производственное предприятие.

Цель инвестиционного контракта заключается в получении инвестором наибольшей прибыли в зависимости от оптимального ежегодного распределения денежных потоков между инвестируемым предприятием и банком.

Функциональная схема инвестиционного контракта, описывающая пути движения денежных потоков инвестора, представлена на рисунке 26.

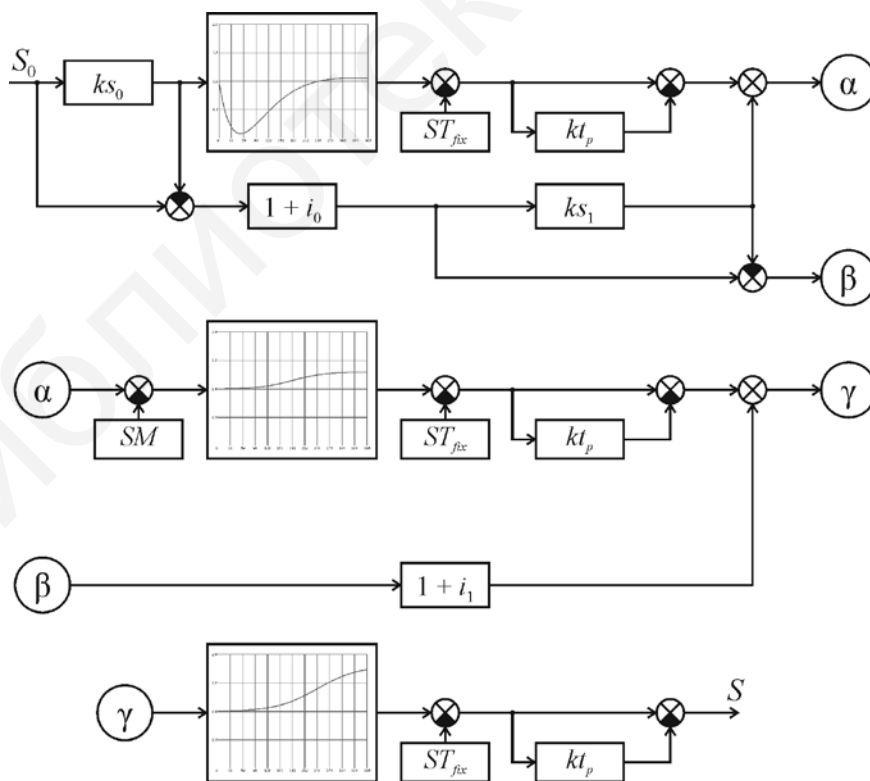


Рисунок 26 – Функциональная схема инвестиционного контракта

Согласно ей начальная сумма  $S_0$  инвестора распределяется между инвестируемым предприятием и банком, причём в первые два года предприятию перечисляется некоторая часть общей суммы инвестора, вырученная к текущему моменту, а оставшиеся денежные средства вкладываются в банк под «сложные» проценты. В последний третий год инвестиционного контракта вся сумма, вырученная за первые два года, полностью вкладывается в предприятие.

В задаче полагается, что производственное предприятие создаётся инвестором «с нуля» и в начале второго года будет проведена его модернизация, а к третьему году продукция предприятия успешно завоюет часть рынка, и в течение последнего третьего года его доходность будет расти только за счёт расширения рынков сбыта.

В связи с данными предположениями, функция доходности предприятия в течение первого года имеет выраженный спад, связанный с издержками в виде арендной платы, закупки оборудования, выплаты зарплаты и пр.

Динамика доходности предприятия в течение первого года математически описывается функцией

$$rr_1(t) = 1 + k(s_0) \cdot t \cdot (t - t_0) \cdot e^{-\frac{t}{T(s_0)}}, \quad (1)$$

где  $t$  – время, измеряемое в днях;

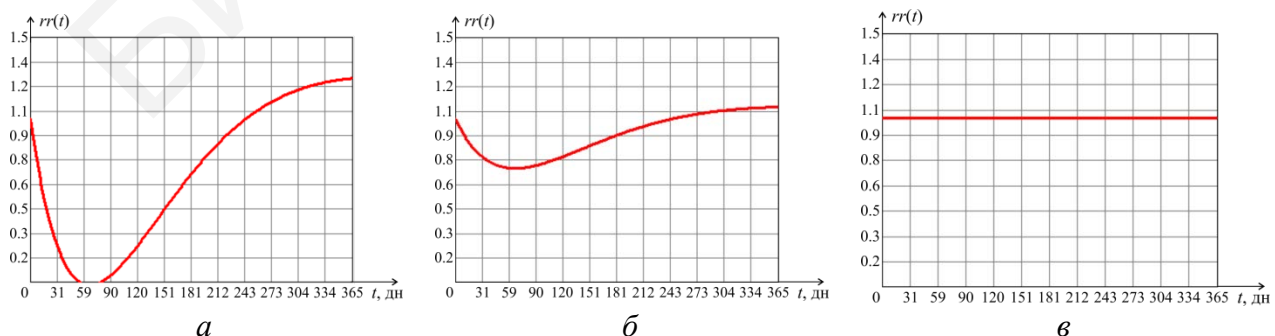
$k$  – коэффициент, определяющий крутизну спада функции доходности в начальный период функционирования предприятия и обратно пропорциональный зависящий от стартовых вложений  $s_0 = ks_0 \cdot S_0$ ;

$t_0$  – время выхода доходов предприятия на исходный уровень начальных вложений;

$T$  – постоянная времени, определяющая скорость изменения доходности и время минимального её значения, также наряду с коэффициентом  $k$  зависящая от стартовых вложений  $s_0$ .

Зависимость  $s_0 = ks_0 \cdot S_0$ , где  $S_0$  – общая сумма инвестиций, очевидно, следует из функциональной схемы рисунка 26. В приведённой зависимости  $ks_0$  – это коэффициент, определяющий часть общих инвестиционных средств, вкладываемых в предприятие в первый год.

В зависимости от коэффициента  $k = k(s_0)$  вид функции  $rr_1(t)$  выглядит так, как показано на рисунке 27.



$$a - k = 1,6 \cdot 10^{-4}; \quad б - k = 4,7 \cdot 10^{-5}; \quad в - k = 7,0 \cdot 10^{-8}$$

Рисунок 27 – Вид функции доходности  $rr_1(t)$  первого года в зависимости от коэффициента  $k$

Исходя из рисунка 27 следует заметить, что значительное уменьшение  $k$ , т. е. существенное завышение стартовых вложений  $s_0$  незначительно влияет на результирующую годовую прибыль и соответственно на повышение эффективности функционирования предприятия. Скорее, наоборот, при сильно завышенных стартовых вложениях эффективность предприятия снижается вследствие вероятного нерационального использования денежных средств. Таким образом, в величине суммы стартовых вложений имеется определённый оптимум.

Результирующая доходность по окончании первого года выходит в состояние «насыщения» вследствие неизменности технологического процесса.

После модернизации производства, отражённого на функциональной схеме рисунка 26 в виде суммы  $SM$ , предприятие переходит на более высоко-технологичный уклад с повышением производительности, функция доходности в течение второго года становится монотонно возрастающей и представляется в виде

$$rr_2(t) = \frac{a+b \cdot e^{-\frac{t-t_0}{T}}}{1+e^{-\frac{t-t_0}{T}}}, \quad (2)$$

где  $t$  – время, измеряемое в днях;

$a$  и  $b$  – уровни, определяющие входную (в начале года) и выходную (в конце года) доходности соответственно:  $a = 1$ ,  $b > 1$ ;

$t_0$  – время «середины» роста доходов;

$T$  – крутизна функции доходности.

Параметры  $t_0$  и  $T$  выражения (2) зависят от вложений  $s_1$  в предприятие в начале второго года. График функции (2) изображён на рисунке 28.

Замедленное развитие предприятия согласно рисунку 28 в начале второго года связано с устоявшимися новогодними каникулами, налоговыми издержками и иными денежными отчислениями. Состояние «насыщения» доходности в конце года объясняется полным освоением технологического потенциала предприятия и требованием либо очередной модернизации, либо расширения рынков сбыта.

Доходность  $rr_3(t)$  третьего года инвестиционного контракта также определяется функцией (2) с параметрами  $b$ ,  $T$  и  $t_0$ , значительно отличающимися от аналогичных параметров функции  $rr_2(t)$  доходности второго года за счёт расширения рынков сбыта продукции предприятия и более высокой квалификации персонала.

В итоге функция доходности предприятия в течение трёх лет инвестиционного контракта принимает примерный вид, изображённый на рисунке 29.

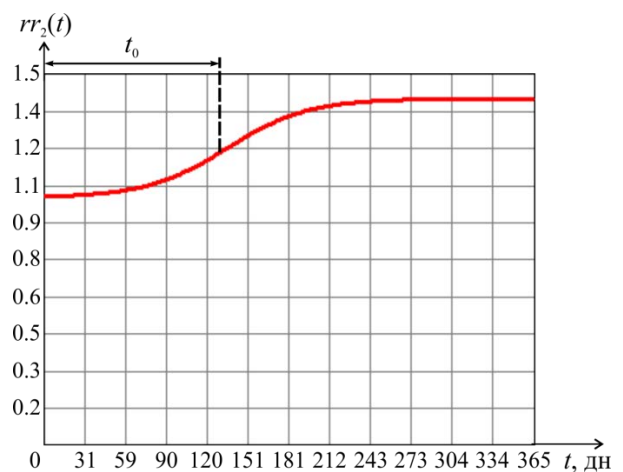


Рисунок 28 – График функции (2)

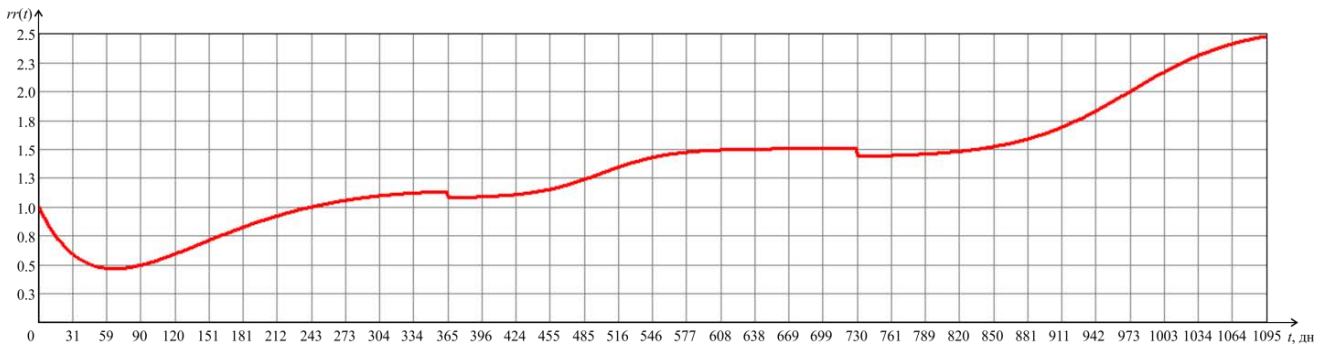


Рисунок 29 – Примерная функция доходности предприятия в течение периода инвестиционного контракта

Ступенчатые «разрывы» в начале второго и третьего годов обусловлены налогообложением и отчислением в соответствующие фонды.

Согласно функциональной схеме инвестиционного контракта на рисунке 26, у инвестора имеется некоторая денежная сумма  $S_0$ . Её часть, равная  $s_0 = ks_0 \cdot S_0$ , на год вкладывается в производство, а оставшаяся сумма  $(1 - ks_0) \cdot S_0$  – в банк под  $i_0$  процентов. За первый год предприятие приумножает часть вложенных средств инвестора в  $rr_1(t = 365)$  раз:  $rr_1(t = 365) \cdot s_0 = rr_1(t = 365) \cdot ks_0 \cdot S_0$ . По окончании первого года предприятие проводит фиксированные отчисления  $ST_{fix}$  и выплачивает налог на прибыль. В результате всех выплат после первого года функционирования предприятия в его активе имеется сумма

$$S_1 = (rr_1(t = 365) \cdot ks_0 \cdot S_0 - ST_{fix}) \cdot (1 - kt_p). \quad (3)$$

Банковский вклад инвестора к концу первого года составляет  $S_0 \cdot (1 - ks_0) \cdot (1 + i_0)$ . В начале второго года к активам предприятия добавляется часть суммы, снятая со счёта в банке, а именно  $S_0 \cdot (1 - ks_0) \cdot (1 + i_0) \cdot ks_1$ . На общую результирующую сумму

$$(rr_1(t = 365) \cdot ks_0 \cdot S_0 - ST_{fix}) \cdot (1 - kt_p) + S_0 \cdot (1 - ks_0) \cdot (1 + i_0) \cdot ks_1$$

проводится модернизация производства, отражённая в функциональной схеме рисунка 26 как  $SM$ . Таким образом, к началу второго года предприятие обладает финансовыми средствами, равными

$$S_{20} = (rr_1(t = 365) \cdot ks_0 \cdot S_0 - ST_{fix}) \cdot (1 - kt_p) + S_0 \cdot (1 - ks_0) \cdot (1 + i_0) \cdot ks_1 - SM.$$

Далее, следуя функциональной схеме рисунка 26, сумма средств предприятия по окончании второго года равна

$$\begin{aligned} & rr_2(t = 365) \cdot S_{20} = \\ & = rr_2(t = 365) \cdot ((rr_1(t = 365) \cdot ks_0 \cdot S_0 - ST_{fix}) \cdot (1 - kt_p) + S_0 \cdot (1 - ks_0) \cdot (1 + i_0) \cdot ks_1 - SM), \end{aligned}$$

и с учётом налогов и отчислений

$$\begin{aligned} S_2 &= (rr_2(t = 365) \cdot S_{20} - ST_{fix}) \cdot (1 - kt_p) = \\ &= [rr_2(t = 365) \times \\ &\times \{(rr_1(t = 365) \cdot ks_0 \cdot S_0 - ST_{fix}) \cdot (1 - kt_p) + S_0 \cdot (1 - ks_0) \cdot (1 + i_0) \cdot ks_1 - SM\} - \\ &\quad - ST_{fix}] \cdot (1 - kt_p). \end{aligned} \quad (4)$$



В последний третий год контракта инвестор добавляет к финансам предприятия все банковские сбережения, равные

$$S_0 \cdot (1 - ks_0) \cdot (1 - ks_1) \cdot (1 + i_0) \cdot (1 + i_1),$$

и по окончании последнего третьего года с учётом налогообложения объём итоговых финансов будет равен

$$S = S_3 = [rr_3(t = 365) \cdot \{ [rr_2(t = 365) \times \\ \times \{ (rr_1(t = 365) \cdot ks_0 \cdot S_0 - ST_{fix}) \cdot (1 - kt_p) + S_0 \cdot (1 - ks_0) \cdot (1 + i_0) \cdot ks_1 - SM \} - \\ - ST_{fix}] \cdot (1 - kt_p) + S_0 \cdot (1 - ks_0) \cdot (1 - ks_1) \cdot (1 + i_0) \cdot (1 + i_1) \} - ST_{fix}] \cdot (1 - kt_p). \quad (5)$$

Очевидным отражением инфляции в постановке задачи является рост банковского процента  $i$  в первый и второй годы функционирования предприятия:  $i_1 > i_0$ .

Таким образом, основная задача курсовой работы состоит в разработке программного модуля анализа и поиска оптимального распределения инвестиционных средств между банком и предприятием, т. е. поиска коэффициентов  $ks_0$  и  $ks_1$  для заданных функций  $rr_1(t)$ ,  $rr_2(t)$ ,  $rr_3(t)$  доходности предприятия соответственно в первый, второй и третий годы инвестиционного контракта; годовых банковских процентов  $i_0$ ,  $i_1$ ; параметров  $ST_{fix}$ ,  $kt_p$  налогообложения и суммы  $SM$  модернизации.

В требования к пользовательскому интерфейсу и функциональности программы по теме №12 входит следующее: визуализация функциональной схемы модели инвестиционного контракта; организация корректного ввода параметров модели: коэффициентов  $ks_0$  и  $ks_1$ , банковских процентов  $i_0$ ,  $i_1$ , параметров  $ST_{fix}$ ,  $kt_p$  налогообложения и суммы  $SM$  модернизации; организация и проверка корректности ввода необходимых параметров функций  $rr_1(t)$ ,  $rr_2(t)$ ,  $rr_3(t)$  годовых доходностей; визуализация непосредственно на функциональной схеме результирующих графиков функций  $rr_1(t)$ ,  $rr_2(t)$ ,  $rr_3(t)$  годовых доходностей; построение результирующей функции доходности предприятия в течение всего периода инвестиционного контракта, наподобие рисунка 29; ввод и проверка корректности ввода ставки  $d$  дисконтирования для расчёта чистого приведённого дохода; вычисление внутренней нормы доходности и чистого приведённого дохода с выводом результатов пользователю на экран; наличие кнопки вычисления оптимальных коэффициентов  $ks_0$  и  $ks_1$  как значений, соответствующих максимуму чистого приведённого дохода; наличие программного модуля оптимизации для поиска наилучшего распределения коэффициентов  $ks_0$  и  $ks_1$ , соответствующих максимуму чистого приведённого дохода; вывод оптимальных значений пользователю на экран.



## ЗАКЛЮЧЕНИЕ

В заключение следует перечислить наиболее частые заблуждения и ошибки, допускаемые в курсовой работе абсолютным большинством студентов касательно оформления, стиля и полноты изложения материала.

Следующий перечень замечаний, предназначенный для акцентирования внимания и устранения основных студенческих ошибок, обусловлен требованием высокой информативной плотности курсовой работы:

- страница пояснительной записки должна быть заполнена как минимум на треть, а лучше наполовину;

- при переносе рисунка или таблицы на следующую страницу (вследствие недостатка места для их полноценного размещения на текущей странице) на текущей странице разрешается оставлять не более пяти текстовых строк свободного места; в противном случае требуется заполнение «пустот» тематической информацией, в частности, дополнительными пояснениями к рисункам или иному сопутствующему материалу пояснительной записки;

- пункты или разделы пояснительной записки не должны завершаться рисунком, таблицей, формулой или фрагментом программного кода без последующих, сопутствующих им пояснений;

- следует предусматривать детальные пояснения к рисункам и графическому материалу, чтобы сторонние специалисты, читающие работу, интерпретировали их одинаково, без искажения основной авторской мысли;

- очевидность информации, изображённой на рисунке, и программного текста, якобы не требующих пояснений, является заблуждением, ибо представленный автором материал без сопутствующей детализации может быть интерпретирован не в его пользу;

- алгоритмы должны отражать глубокие знания автора, а не поверхностное понятие работы программы, поэтому необходимо составлять сложные алгоритмы как минимум разветвляющегося типа;

- схема алгоритма и диаграммы состояний или последовательности должны содержать подробные блоки работы программы, иметь полное и чёткое описание; так, наличие на схеме/диаграмме блока вида «Преобразование массива» при очевидно небольшом общем количестве блоков воспринимается сторонними специалистами как существенная недоработка, особенно в случаях, когда в курсовой работе по данному блоку отсутствуют пояснения, в частности детализация используемого в программе способа преобразования массива в виде фрагментов программного кода с сопутствующим описанием;

– схема алгоритма, диаграммы состояний и последовательности с сопутствующими пояснениями должны иметь логическую завершённость, быть читаемыми не только автором, но и сторонними квалифицированными специалистами; в частности, если в графическом материале имеется переменная, то его необходимо составить так, чтобы путь изменения данной переменной в схеме/диаграмме был чётким, логичным, легко отслеживаемым.

Следует отметить, что неясность и нечёткость изложения материала – это основной недостаток работы автора.

Немаловажным фактором является наличие в программном коде выходов из непредвиденных ситуаций, что обеспечивает безопасность работы с программой неподготовленных людей.

Внешний вид пользовательского интерфейса следует детально продумать таким образом, чтобы в нём присутствовали лаконичность, эргономичность и интуитивная понятность, при этом отсутствовала избыточность, а разработанная программа имела готовое практическое применение и была доступна любому пользователю.

Настоятельно рекомендуется наличие в программе информационного пункта «О программе» со сведениями о версии программы и информации об авторах.

# ПРИЛОЖЕНИЕ А

## (обязательное)

### Образец задания по курсовой работе

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники»  
Факультет компьютерного проектирования  
Кафедра проектирования информационно-компьютерных систем



«УТВЕРЖДАЮ»

Заведующий кафедрой

\_\_\_\_\_  
И.О. Фамилия  
(подпись зав. кафедрой) (И.О. Фамилия зав. кафедрой)

«\_\_» \_\_\_\_\_ 20\_\_ г.

### ЗАДАНИЕ

по курсовой работе по дисциплине «**Основы программирования информационных систем**»

Фамилия, имя, отчество \_\_\_\_\_  
(Фамилия Имя Отчество студента)

\_\_\_\_\_ группа \_\_\_\_\_  
(Номер группы студента)

1. Тема работы: \_\_\_\_\_

2. Срок сдачи студентом законченной работы: «\_\_» \_\_\_\_\_ 20\_\_ г.

#### 3. Исходные данные к работе:

3.1. Описание к выполнению \_\_\_\_\_

3.2. Язык и среда программирования – на выбор студента. Однако разработанное программное обеспечение должно выполняться в системе Windows 7 / 8 / 10 с возможной предустановкой библиотек или пакетов выбранной среды программирования.

3.3. В реализации программного обеспечения учесть возможность анимационного отображения процессов генерирования, обработки и анализа массивов данных (визуальное движение, изменение графиков, таблиц).

3.4. Пояснительную записку и графический материал выполнять по СТП БГУИР 01-2017.

3.5. Другие требования уточняются студентом в процессе работы.

#### 4. Содержание расчётно-пояснительной записки (перечень подлежащих разработке вопросов)

Титульный лист. Заполненный бланк задания с приложением. Содержание (4 – 6 стр.).

Введение (1 – 3 стр. Актуальность темы курсовой работы; цель и перечень задач, которые планируется решить; детальная постановка задачи).

4.1. Математическая модель задачи курсовой работы (10 – 12 стр. Основные понятия и определения. Сфера применения. Математическое описание физических и/или экономических основополагающих принципов решения задачи. Математическое обоснование способов решения задачи курсовой работы).

4.2. Интерфейсы курсовой работы (6 – 8 стр. Понятие интерфейса. Описание, краткая характеристика и модели используемых в курсовой работе интерфейсов).

4.3. Объектно-ориентированные технологии программирования (5–7 стр. Технологии программирования, используемые для решения поставленных задач. Реализация объектно-ориентированных технологий программирования в современных программно-математических средах).

4.4. Проектирование графического пользовательского интерфейса средствами СОМ-объектов (5–7 стр. Обоснование проекта пользовательского интерфейса).

4.5. Алгоритм функционирования программного обеспечения (5–7 стр. Разработка схемы алгоритма, диаграммы последовательности и диаграммы состояний (схемы в графической части к курсовой работе) с детальными пояснениями каждого компонента).

4.6. Программная реализация алгоритма (10–12 стр. Перечень используемых функций и библиотек. Описание форматов представления данных и программных фрагментов их обработки. Обоснование подпрограмм вычислительных алгоритмов).

4.7. Анализ результатов решения поставленной задачи (5–7 стр. Анализ графической информации, полученной в ходе решения поставленной задачи).

Заключение (1 стр. Выводы по курсовой работе).

Список использованных источников (1–2 стр. Перечень литературы и интернет-источников, которые были реально использованы при выполнении курсовой работы).

Приложения (3 и более стр. Листинг программного кода, ведомость документов и др.).

## 5. Перечень графического материала (с указанием обязательных чертежей и графиков):

5.1. Структура графического пользовательского интерфейса (формат А3 или несколько листов формата А4).

5.2. Схема алгоритма (формат А3 или несколько листов формата А4).

5.3. Диаграмма последовательности (формат А3 или несколько листов формата А4).

5.4. Диаграмма состояний (формат А3 или несколько листов формата А4).

## 6. Консультант по работе:

\_\_\_\_\_ (Фамилия Имя Отчество консультанта)

7. Дата выдачи задания: «\_\_» \_\_\_\_\_ 20\_\_ г.

## 8. Календарный план работы на весь период проектирования:

№ п/п	Наименование этапов курсовой работы	Срок выполнения этапов работы	Примечание
1	1-я опрoцентoвка (пункты 4.1, 4.2, 5.1)	«__» _____ 20__ г.	40 %
2	2-я опрoцентoвка (пункты 4.3 – 4.5, 5.2, 5.3)	«__» _____ 20__ г.	70...80 %
3	3-я опрoцентoвка (пункты 4.6, 4.7, 5.4, приложения)	«__» _____ 20__ г.	95 %
4	Сдача на проверку и защита курсовой работы	«__» _____ 20__ г.	100 %

Руководитель

\_\_\_\_\_ (подпись руководителя)

\_\_\_\_\_ (И.О. Фамилия руководителя)

Задание принял к исполнению

\_\_\_\_\_ (подпись студента)

\_\_\_\_\_ (И.О. Фамилия студента)

**ПРИЛОЖЕНИЕ Б**  
(обязательное)

**Образец титульного листа**

Министерство образования Республики Беларусь

Учреждение образования

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ**

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Дисциплина «Основы программирования информационных систем»

*К защите допустить:*

Руководитель курсовой работы  
учёная\_степень, учёное\_звание

\_\_\_\_\_ И.О. Фамилия\_рук.

\_\_\_\_.\_\_\_\_. 20\_\_

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе  
на тему

**ТЕМА\_КУРСОВОЙ\_РАБОТЫ**

БГУИР КР 1-40 05 01-10 №\_темы\_порядковый ПЗ

Студент

\_\_\_\_\_ (подпись студента)

И.О. Фамилия\_студента

Курсовая работа представлена  
на проверку \_\_\_\_ . \_\_\_\_ . 20 \_\_\_\_

\_\_\_\_\_ (подпись студента)

Минск 20\_\_

## ПРИЛОЖЕНИЕ В (обязательное)

### Пример оформления приложения к пояснительной записке курсовой работы

#### *Переходные характеристики системы управления по задающему и возмущающему воздействиям*

Передаточная функция системы по задающему воздействию с соответствующей переходной характеристикой на рисунке В.1:

$$H_{OY}(\omega) = \frac{K_3 K (1+i\omega T_D)}{(1+i\omega T_1)(1+i\omega T_D)(K_Y K_{ДВ} K_{ОС} + i\omega(1+i\omega T_{ДВ})) + K_{ДК}}$$

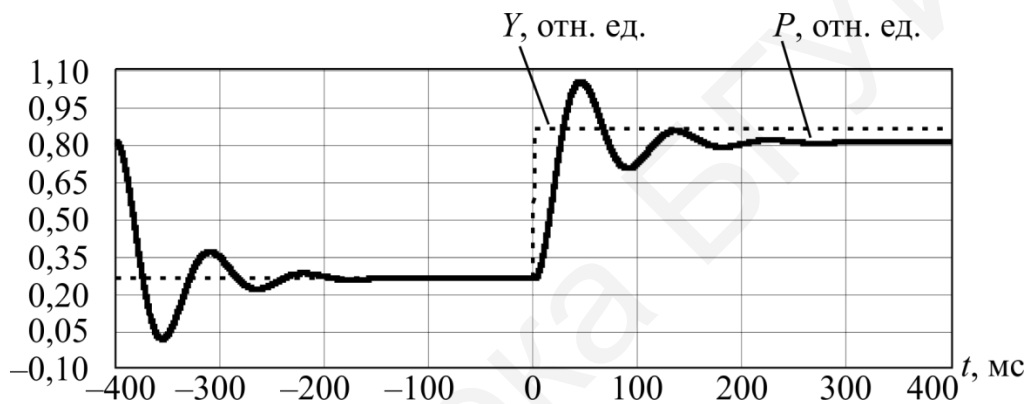


Рисунок В.1 – Переходная характеристика системы  
на задающее воздействие

Передаточная функция системы по возмущающему воздействию с соответствующей переходной характеристикой на рисунке В.2:

$$H_{OF}(\omega) = \frac{K_2 (1+i\omega T_D)(K_Y K_{ДВ} K_{ОС} + i\omega(1+i\omega T_{ДВ}))}{(1+i\omega T_1)(1+i\omega T_D)(K_Y K_{ДВ} K_{ОС} + i\omega(1+i\omega T_{ДВ})) - K_{ДК}}$$

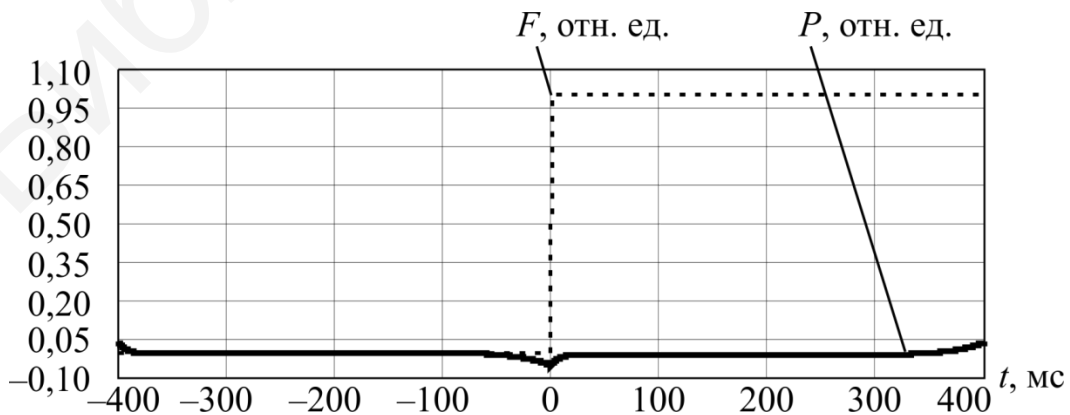


Рисунок В.2 – Переходная характеристика системы  
на возмущающее воздействие



## ПРИЛОЖЕНИЕ Д (обязательное)

### Пример оформления страницы пояснительной записки с иллюстрацией, расположенной между абзацами

Фрагмент схемы автоматизации технологического процесса регулирования давления пара в барабане котла представлен на рисунке <номер рисунка>. В процессе сгорания топливно-воздушной смеси, поступающей в топку котла, вода в подогретом барабане превращается в пар, который попадает в паропровод и далее – на вырабатывающую электроэнергию турбину или в систему отопления.

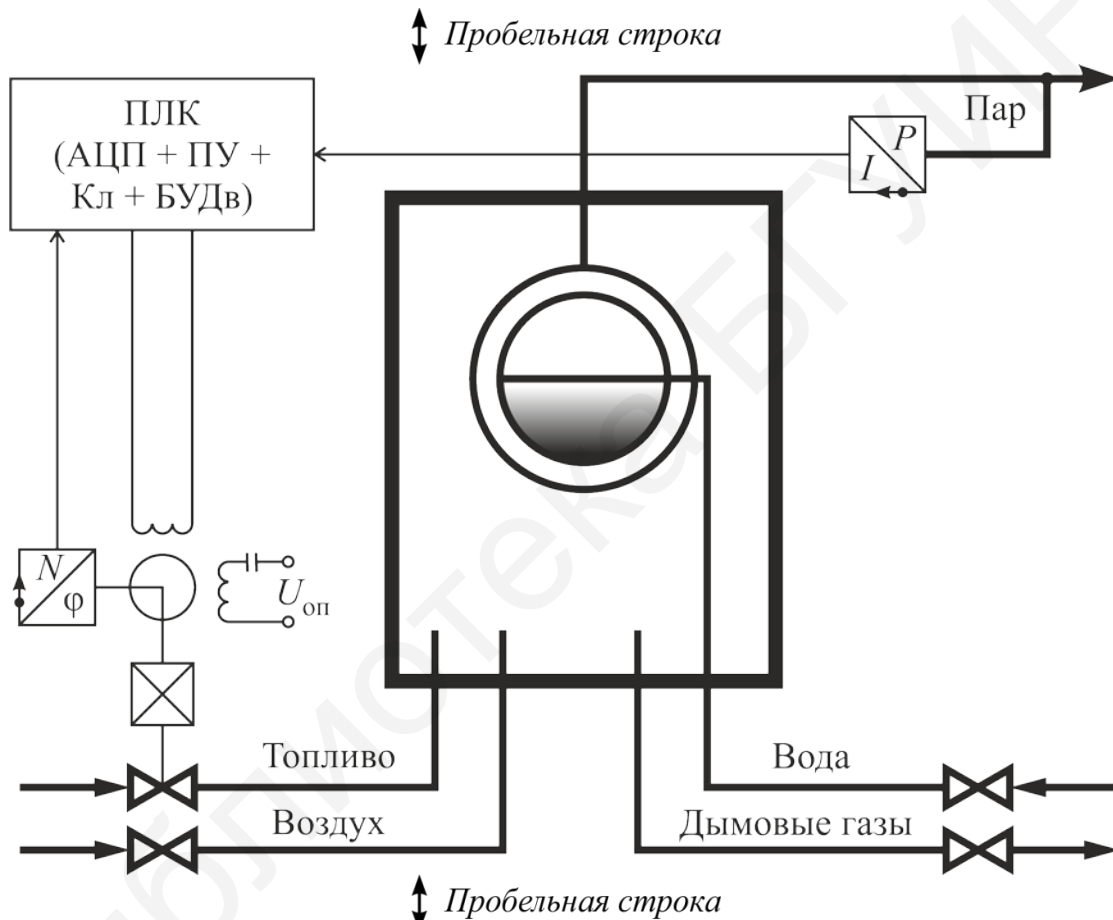


Рисунок <номер рисунка – Подрисуночная подпись>

Основная задача системы автоматического регулирования состоит в стабилизации давления пара в паропроводе путём управления подачей топлива. В данной задаче объектом управления выступает паровой котёл, а регулируемой величиной является давление  $P$  пара в паропроводе. Цель управления – поддержание давления пара на постоянном заданном уровне  $P_3$ , который устанавливается посредством клавиатуры программируемого логического контроллера (ПЛК). Управляющим воздействием на объект является угол  $\varphi$ в поворота вентильной заслонки, определяющий расход поступающего в топку котла топлива. В качестве возмущающего воздействия выступает изменение расхода  $Q_{п}$  пара от номинального значения.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Положение об организации и проведении курсового проектирования в БГУИР [Электронный ресурс]. – 2018. – Режим доступа : [https://libeldoc.bsuir.by/bitstream/123456789/25903/1/kurs\\_proekt.pdf](https://libeldoc.bsuir.by/bitstream/123456789/25903/1/kurs_proekt.pdf).
2. СТП 01-2017. Стандарт предприятия. Дипломные проекты (работы). Общие требования [Электронный ресурс]. – 2018. – Режим доступа : [https://libeldoc.bsuir.by/bitstream/123456789/29077/1/STP\\_01\\_2017.pdf](https://libeldoc.bsuir.by/bitstream/123456789/29077/1/STP_01_2017.pdf).
3. ГОСТ 2.104-2006. Единая система конструкторской документации. Основные надписи [Электронный ресурс]. – 2018. – Режим доступа : <http://docs.cntd.ru/document/1200045443>.
4. ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам [Электронный ресурс]. – 2018. – Режим доступа : <http://docs.cntd.ru/document/1200001260>.
5. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание [Электронный ресурс]. – 2018. – Режим доступа : [http://diss.rsl.ru/datadocs/doc\\_291wu.pdf](http://diss.rsl.ru/datadocs/doc_291wu.pdf).
6. ГОСТ 2.106-96. Единая система конструкторской документации. Текстовые документы [Электронный ресурс]. – 2018. – Режим доступа : <http://library.onaft.edu.ua/DSTU/2.106-96.pdf>.
7. ГОСТ 2.004-88. Единая система конструкторской документации. Общие требования к выполнению конструкторских и технологических документов на печатающих и графических устройствах вывода ЭВМ [Электронный ресурс]. – 2018. – Режим доступа : <http://www.library.onaft.edu.ua/DSTU/2.004-88.pdf>.
8. ГОСТ 7.9-95. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Реферат и аннотация. Общие требования [Электронный ресурс]. – 2018. – Режим доступа : [http://vestnik.mrsu.ru/content/files/GOST\\_7.9-95.pdf](http://vestnik.mrsu.ru/content/files/GOST_7.9-95.pdf).
9. ГОСТ 2.201-80. Единая система конструкторской документации. Обозначение изделий и конструкторских документов [Электронный ресурс]. – 2018. – Режим доступа : [http://www.robot.bmstu.ru/files/GOST/gost\\_2.201-80.pdf](http://www.robot.bmstu.ru/files/GOST/gost_2.201-80.pdf).
10. ОК 012-93. Классификатор ЕСКД. Общероссийский классификатор изделий и конструкторских документов [Электронный ресурс]. – 2018. – Режим доступа : <http://classinform.ru/ok-eskd/kod.html>.
11. ОКП 425100. Программно-технические комплексы для автоматизации/управления организационно-экономическими процессами [Электронный ресурс]. – 2018. – Режим доступа : <http://www.rospromtest.ru/kody-okp/425100/>.
12. ГОСТ 2.701-2008. Единая система конструкторской документации. Виды и типы. Общие требования к выполнению [Электронный ресурс]. – 2019. – Режим доступа : <http://files.stroyinf.ru/Data2/1/4293830/4293830651.pdf>.
13. ГОСТ 2.605-68. Единая система конструкторской документации. Плакаты учебно-технические. Общие технические требования

[Электронный ресурс]. – 2019. – Режим доступа : <https://meganorm.ru/Data2/1/4294849/4294849501.pdf>.

14. Грибова, В. В. Методы и средства разработки пользовательского интерфейса: современное состояние / В. В. Грибова, А. С. Клещёв [Электронный ресурс]. – 2019. – Режим доступа : <https://cyberleninka.ru/article/v/metody-i-sredstva-razrabotki-polzovatel'skogo-interfeysa-sovremennoe-sostoyanie>.

15. Грибова, В. В. Методы и средства разработки пользовательского интерфейса: современное состояние / В. В. Грибова, А. С. Клещёв [Электронный ресурс]. – 2019. – Режим доступа : <http://guimachine.ru/?p=937>.

16. Ролич, О. Ч. Технологии программирования : курс лекций / О. Ч. Ролич. – Минск : БГУ, 2008. – 144 с.

17. Microsoft Docs [Электронный ресурс]. – 2019. – Режим доступа : <https://docs.microsoft.com/ru-ru/>.

18. .NET Framework [Электронный ресурс]. – 2019. – Режим доступа : [https://ru.wikipedia.org/wiki/.NET\\_Framework](https://ru.wikipedia.org/wiki/.NET_Framework).

19. Object Linking and Embedding [Электронный ресурс]. – 2019. – Режим доступа : [https://ru.wikipedia.org/wiki/Object\\_Linking\\_and\\_Embedding](https://ru.wikipedia.org/wiki/Object_Linking_and_Embedding).

20. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов программ, данных и систем [Электронный ресурс]. – 2019. – Режим доступа : <http://gostexpert.ru/data/files/19.701-90/9c598f3f161aba120946a88becb47666.pdf>.

21. Пустоваров, В. И. Язык ассемблера в программировании информационных и управляющих систем / В. И. Пустоваров. – Киев : Век ; М. : Энтроп ; М. : Десс, 1998. – 304 с.

22. Блок-схемы. Основные алгоритмические конструкции [Электронный ресурс]. – 2019. – Режим доступа : [http://inf.susu.ac.ru/Klinachev/lc\\_sga\\_21.htm](http://inf.susu.ac.ru/Klinachev/lc_sga_21.htm).

23. Отвагин, А. В. Объектно-ориентированное проектирование и программирование : лабораторный практикум для студентов специальности 1-40 02 01 «Вычислительные машины, системы и сети» всех форм обучения / А. В. Отвагин, Н. А. Павлёнок. – Минск : БГУИР, 2011. – 44 с.

24. Диаграмма состояний [Электронный ресурс]. – 2019. – Режим доступа : <http://masters.donntu.org/2005/kita/shapovalova/library/uml1.html>.

25. Диаграмма последовательности [Электронный ресурс]. – 2019. – Режим доступа : <https://www.litmir.me/br/?b=133827&p=45>.

26. Синонимы к слову «программа» [Электронный ресурс]. – 2019. – Режим доступа : <https://sinonim.org/s/программа#>.

27. Синонимы к слову «программа» [Электронный ресурс]. – 2019. – Режим доступа : <https://kartaslov.ru/синонимы-к-слову/программа>.

28. Макаров, Е. Г. MathCAD: учебный курс / Е. Г. Макаров. – СПб. : Питер, 2009. – 384 с.

29. Гетманов, В. Г. Цифровая обработка сигналов : учеб. пособие / В. Г. Гетманов. – М. : НИЯУ МИФИ, 2010. – 232 с.

30. Бодянский, Е. В. Искусственные нейронные сети: архитектуры, обучение, применения / Е. В. Бодянский, О. Г. Руденко. – Харьков : ТЕЛЕТЕХ, 2004. – 369 с.
31. Пупков, К. А. Интеллектуальные системы / К. А. Пупков, В. Г. Коньков. – М. : Издательство МГТУ им. Н. Э. Баумана, 2003. – 348 с.
32. Бендат, Дж. Применение корреляционного и спектрального анализа / Дж. Бендат, А. Пирсол. – М. : Мир, 1983. – 312 с.
33. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
34. Таненбаум, Э. Компьютерные сети / Э. Таненбаум, Д. Уэзеролл. – СПб. : Питер, 2012. – 960 с.
35. Ролич, О. Ч. Тепловой контроль работы двигателя на основе статистического анализа сигналов болометра / О. Ч. Ролич, В. Е. Тарасенко, В. С. Ивашко // Изобретатель. – 2019. – №2–3 – С. 40 – 44.
36. Палий, И. А. Линейное программирование / И. А. Палий. – М. : Эксмо, 2008. – 256 с.
37. Линейное программирование в современных задачах оптимизации / Ю. В. Бородакий [и др.]. – М. : МИФИ, 2008. – 188 с.
38. Вентцель, Е. С. Элементы динамического программирования / Е. С. Вентцель. – М. : Наука, 1964. – 176 с.
39. Визгунов, Н. П. Динамическое программирование в экономических задачах с применением системы MATLAB / Н. П. Визгунов. – Нижний Новгород : ННГУ, 2006. – 48 с.
40. Аванесов, А. Т. Финансово-экономические расчёты: анализ инвестиций и контрактов / А. Т. Аванесов, М. М. Ковалёв, В. Г. Руденко. – Минск : БГУ, 1998. – 246 с.

*Учебное издание*

**Ролич Олег Чеславович**

***ОСНОВЫ ПРОГРАММИРОВАНИЯ  
ИНФОРМАЦИОННЫХ СИСТЕМ.  
КУРСОВОЕ ПРОЕКТИРОВАНИЕ***

ПОСОБИЕ

Редактор *Е. И. Костина*

Корректор *Е. Н. Батурчик*

Компьютерная правка, оригинал-макет *М. В. Касабуцкий*

Подписано в печать 13.01.2020. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».  
Отпечатано на ризографе. Усл. печ. л. 5,0. Уч.-изд. л. 5,0. Тираж 50 экз. Заказ 266.

Издатель и полиграфическое исполнение: учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники».  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий №1/238 от 24.03.2014,  
№2/113 от 07.04.2014, №3/615 от 07.04.2014.  
Ул. П. Бровки, 6, 220013, г. Минск