

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ПРОГРАММИРОВАНИЕ СЕТЕВЫХ ПРИЛОЖЕНИЙ

*Методические рекомендации к курсовому проектированию
для студентов специальности 6-05-0611-01
«Информационные системы и технологии»
очной и заочной форм обучения*



Могилев 2026

УДК 004.42
ББК 32.973-018
П78

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»
«18» ноября 2025 г., протокол № 4

Составитель канд. техн. наук, доц. И. В. Акиншева

Рецензент канд. техн. наук, доц. Ю. С. Романович

Даны методические рекомендации к выполнению курсового проекта по дисциплине «Программирование сетевых приложений». Приведены требования к содержанию и оформлению курсового проекта, список литературы для подготовки.

Учебное издание

ПРОГРАММИРОВАНИЕ СЕТЕВЫХ ПРИЛОЖЕНИЙ

Ответственный за выпуск	А. И. Якимов
Корректор	И. В. Голубцова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2026

Содержание

Введение.....	4
1 Общие положения	6
2 Цели и задачи курсового проектирования.....	6
3 Объем, архитектура и структура курсового проекта.....	8
4 Требования к оформлению курсового проекта.....	11
5 Организация выполнения курсового проекта	15
6 Защита курсового проекта.....	16
Список литературы	16
Приложение А. Примеры индивидуальных заданий	17
Приложение Б. Примеры используемых блоков кода.....	21
Приложение В. Пример оформления титульного листа	24
Приложение Г. Примеры графической части курсового проекта.....	25

Введение

Целью учебной дисциплины «Программирование сетевых приложений» является подготовка специалистов, владеющих базовыми знаниями, умениями и практическими навыками в области языков и средств разработки сетевых приложений, ориентированных на клиент-серверную архитектуру, программирования элементов такой архитектуры.

Задачей учебной дисциплины является формирование у будущих специалистов знаний о современных программных средствах, применяемых для программирования сетевых приложений.

В результате освоения учебной дисциплины обучающийся:

а) узнает:

- основы и особенности применяемого языка программирования;
- основы разработки простейших клиент-серверных программ, методы разработки программ серверного обслуживания и методы организации клиент-серверного взаимодействия;
- последовательное, многопоточное и многопроцессное серверное обслуживание клиентов;
- методы и средства разработки взаимодействия и отладки программ клиент-серверной архитектуры, порядок разработки протоколов взаимодействия и методы их использования в программах для разработки программ;
- технологию клиент-серверного взаимодействия, порядок разработки интерфейса пользователя;

б) научится:

- создавать приложения удаленного вызова процедур и методов, приложения в архитектуре «клиент-сервер» с различными методами и процедурами серверной обработки запросов клиентов;
- разрабатывать приложения с различными типами клиентов приложений и интерфейсов, используя современные технологии;
- использовать для разработки приложений наиболее распространенные сетевые протоколы обмена данными и другие средства передачи данных в клиент-серверных архитектурах;

в) овладеет:

- инструментальными средствами разработки сетевых приложений;
- программными средствами разработки распределенных информационных систем;
- техникой компонентно-ориентированной разработки клиент-серверных приложений;
- методами разработки приложений распределенной обработки данных и технологии удаленных вызовов;
- языками и техникой программирования серверных приложений и интерфейсов пользователя.

В основе сетевой поддержки языков программирования лежит концепция сокета (socket). Сокет идентифицирует конечную точку сети. Сокеты – основа современных сетей, поскольку сокет позволяет отдельному компью-

теру обслуживать одновременно множество ратных клиентов и множество различных типов информации. Это достигается за счет использования порта (port) – нумерованного сокета на определенной машине. Считают, что серверный процесс «слушает» порт до тех пор, пока клиент не соединится с ним. Сервер в состоянии принять множество клиентов, подключенных к одному и тому же номеру порта, хотя каждый сеанс является уникальным. Чтобы обработать множество клиентских соединений, серверный процесс должен быть многопоточным либо обладать какими-то другими средствами обработки одновременного ввода-вывода.

Сетевые коммуникации происходят по протоколу IP (Internet Protocol) – это низкоуровневый маршрутизирующий протокол, который разбивает данные на небольшие пакеты и посылает их через сеть по определенному адресу, что не гарантирует доставки всех этих пакетов по этому адресу.

Протокол управления передачей TCP (Transmission Control Protocol) – является протоколом более высокого уровня, обеспечивающим надежную сборку этих пакетов, сортировку и повторную передачу, необходимую для надежной доставки данных. Третий протокол пользовательских дейтаграмм UDP (User Datagram Protocol), стоящий непосредственно за TCP. Может быть использован непосредственно для поддержки быстрой, не требующей постоянного соединения и ненадежной транспортировки пакетов.

Имя адреса в Интернете, называемое доменным именем, описывает местонахождение машины в пространстве имен. Например, `www.bru.by` относится к домену `by` (зарезервированному для сайтов Республики Беларусь), имеет имя `bru` (по названию организации), а `www` идентифицирует сервер, обрабатывающий Web-запросы. Доменное имя Интернет отображается на IP-адрес посредством службы доменных имен (Domain Name Service – DNS). Это позволяет пользователям работать с доменными именами, в то время как Интернет оперирует IP-адресами.

Все вышеописанные средства необходимы для написания сетевых приложений.

1 Общие положения

Курсовой проект представляет собой вид самостоятельной учебной работы и контроля качества обучения студента на определенных этапах выполнения учебного плана по специальности (направлению специальности).

Выполнение курсовых проектов предусматривается типовыми учебными планами в целях закрепления, углубления и обобщения знаний, полученных студентом в процессе обучения, развития способности самостоятельного и творческого мышления.

Основными требованиями, предъявляемыми к курсовому проекту, являются:

- целевая направленность;
- четкость построения;
- логическая последовательность изложения материал;
- глубина исследования и полнота освещения материала;
- убедительность аргументаций;
- краткость и точность формулировок;
- конкретное изложение результатов работы;
- доказательность выводов и обоснованность рекомендаций.

Курсовой проект является формой текущей аттестации студентов по дисциплине учебного плана специальности.

Зачет по курсовому проекту учитывается при определении общего числа экзаменов и зачетов на соответствующем этапе обучения.

Темы курсовых проектов вносятся в зачетные книжки студентов.

2 Цели и задачи курсового проектирования

Целями выполнения курсового проекта являются: развитие навыков реализации сетевых приложений с использованием прикладного интерфейса программирования сокетов; среда разработки прикладных сетевых приложений; получение опыта разработки сетевых программных средств.

При выполнении курсового проекта у студента должно формироваться представление об этапах проектирования и создания сетевых программных средств.

В ходе достижения цели решаются следующие задачи:

- развитие логического и алгоритмического мышления;
- выработка умения формулировать суждения и выводы, логически последовательно и доказательно их излагать;
- овладение навыками самостоятельной исследовательской работы;
- закрепление использования технологии сетевого программирования на стороне клиента и сервера.

Студент по согласованию с преподавателем выбирает тему курсового проекта. Далее приведен примерный перечень тем курсовых проектов по дисциплине «Программирование сетевых приложений».

- 1 Разработка сетевого приложения для автоматизации работы хозрасчетной поликлиники.
- 2 Разработка сетевого приложения для автоматизации работы издательства.
- 3 Разработка сетевого приложения для автоматизации работы учебного отдела университета.
- 4 Разработка сетевого приложения для автоматизации работы кондитерской фабрики.
- 5 Разработка сетевого приложения для автоматизации работы гостиницы.
- 6 Разработка сетевого приложения для автоматизации работы оптового склада.
- 7 Разработка сетевого приложения для автоматизации работы сети магазинов спорттоваров.
- 8 Разработка сетевого приложения для автоматизации работы склада обувной фабрики.
- 9 Разработка сетевого приложения для автоматизации работы хлебозаводов и сетевых магазинов.
- 10 Разработка сетевого приложения для автоматизации работы туристического агентства.
- 11 Разработка сетевого приложения для автоматизации работы мебельного склада.
- 12 Разработка сетевого приложения для автоматизации работы с метрологическим обеспечением предприятия.
- 13 Разработка сетевого приложения для автоматизации работы отдела сбыта металлургического завода.
- 14 Разработка сетевого приложения для автоматизации работы юридической фирмы.
- 15 Разработка сетевого приложения для автоматизации работы автотранспортного предприятия.
- 16 Разработка сетевого приложения для автоматизации работы библиотеки университета.
- 17 Разработка сетевого приложения для автоматизации работы сети мастерских автосервиса.
- 18 Разработка сетевого приложения для автоматизации работы со справочником электрика.
- 19 Разработка сетевого приложения для автоматизации работы оптового склада предприятия.
- 20 Разработка сетевого приложения для автоматизации работы сети автомобильных салонов.
- 21 Разработка сетевого приложения для автоматизации работы пассажирского автопредприятия.
- 22 Разработка сетевого приложения для автоматизации работы агентства «Пресса Беларуси».
- 23 Разработка сетевого приложения для автоматизации работы крестьянского фермерского хозяйства.

24 Разработка сетевого приложения для автоматизации работы агентства по продаже авиабилетов.

25 Разработка сетевого приложения для автоматизации работы торгово-закупочного предприятия.

Преподаватель в соответствии с выбранной темой выдает студенту индивидуальное задание. Примеры индивидуальных заданий приведены в приложении А.

3 Объем, архитектура и структура курсового проекта

Общий объем курсового проекта – минимум 20 страниц машинописного текста. Приложения формируются при необходимости, если имеющиеся таблицы, рисунки, расчеты затрудняют восприятие материала курсового проекта.

3.1 Архитектура курсового проекта

Приложение должно быть выполнено в архитектуре «клиент-сервер» с многопоточным сервером с организацией взаимодействия с базой данных на объектно-ориентированном языке программирования.

Уровни архитектуры.

Серверное приложение может быть реализовано в виде консольного приложения или GUI-приложения. Настройки сервера должны меняться без изменения исходного кода (аргументы командной строки, конфиг-файлы и т. д.).

Клиентское приложение – оконное приложение с использованием стандартных библиотек пользовательского интерфейса.

В рамках работы над курсовым проектом студентами должны быть использованы следующие техники:

- 1) разработка и использование собственной иерархии классов, расширение базовых классов выбранного языка программирования;
- 2) реализация паттернов проектирования на выбор студента;
- 3) использование сокрытия данных (инкапсуляция), перегрузки методов, переопределения методов, сериализации, абстрактных типов данных (интерфейсы, абстрактные классы), статических методов, обработки исключительных ситуаций.

Обязательные требования к функционалу системы:

- 1) в разрабатываемом приложении необходимо обеспечить добавление, редактирование и удаление записей из базы данных, сохранение табличных результатов в файле (создание текстового отчета), предоставление пользователю аналитической информации (в зависимости от индивидуального задания);
- 2) предусмотреть возможность сохранять информацию в любой момент на сервере и загружать ранее сохраненные данные;
- 3) предусмотреть механизм авторизации пользователей (использовать роли Администратор и Пользователь).

Разрабатываемая согласно индивидуальному заданию база данных (не менее трех связанных таблиц) должна быть приведена к третьей нормальной форме.

При разработке приложения с помощью объектно-ориентированного языка программирования Java возможно использование следующих блоков кода, представленных в приложении Б.

3.2 Структура курсового проекта

Структура проекта включает: титульный лист; лист задания на курсовое проектирование; содержание; введение; основную часть (состоящую как минимум из трех разделов); заключение; список использованных источников и приложения. В курсовом проекте предусмотрена графическая часть, выполняемая, в зависимости от размера изображения, на листах форматов А1–А4.

Титульный лист является первым листом курсовой работы. Пример оформления титульного листа приведен в приложении В. Лист задания на курсовое проектирование является вторым листом курсовой работы.

Содержание включает наименование всех разделов, подразделов и пунктов (если они имеют наименование) с указанием номеров страниц, на которых размещается начало материала разделов (подразделов, пунктов). Содержание должно включать все заголовки, имеющиеся в проекте, в том числе «введение», «заключение», «список использованных источников». В содержании перечисляют все приложения с их заголовками. В содержании все номера подразделов могут быть смещены вправо относительно номеров разделов.

Во *введении* следует раскрыть основное содержание и значение избранной темы, необходимо показать актуальность темы, цель и задачи проекта. Также во введении показывается структура проекта, т. е. краткое содержание всех разделов курсового проекта. Объем введения составляет 1–2 листа формата А4.

1 Описание предметной области.

1.1 Общие положения. Обосновывается актуальность темы; формулируется объект (предметная область, в которой проводится исследование), предмет (средства разработки программного изделия, проектированию которого посвящен проект), цель (вытекает из темы проекта) и задачи проекта (задачи формулируются как этапы достижения цели).

1.2 Сведения из теории. Общие сведения по предметной области в объеме, необходимом для написания курсового проекта.

1.3 Постановка задачи. Название и назначение разрабатываемого приложения; спецификация входных и выходных данных; обосновывается и кратко описывается выбор средства (среды и языка) программирования; описывается требуемое оборудование и программное обеспечение.

2 Технологии разработки приложения.

В данной главе последовательно описывается решение сформулированных в первой главе задач, т. е. этапов работы.

2.1 Алгоритм решения. Описание общего алгоритма решения (словесное и/или с использованием блок-схем).

2.2 Макет приложения. Макет приложения может быть представлен в виде

графических схем оконного интерфейса пользователя.

2.3 Описание программы. Используемые классы, иерархия классов, библиотеки, пространства имен, обработчики событий, функции и т. д. в рамках требований к функционалу системы.

Допускается вставка фрагментов кода и их подробное описание согласно общему алгоритму решения задачи.

2.4 Результаты работы программы. Входные данные, выходные данные (при необходимости включить интерфейсные окна программы с результатами).

3 Руководство пользователя.

В данной главе дается подробное (для «чайников») описание технологии и описывается принцип использования разработанного приложения. Также в третьей главе должны содержаться основные выводы по результатам исследования, отражающие практическую значимость работы, предложения по использованию ее результатов.

Заключение должно содержать только те выводы, которые согласуются с целью, сформулированной в разделе «Введение», и должны быть изложены таким образом, чтобы их содержание было понятно без чтения текста проекта.

Выводы могут быть сформулированы по пунктам в такой последовательности, как они будут оглашены в конце доклада на защите курсового проекта. Объем заключения составляет 1–2 листа.

Список использованных источников должен содержать перечень источников, использованных при выполнении курсового проекта. Располагать их следует в алфавитном порядке или в порядке, в котором они встречаются по тексту проекта. В приложения следует включать листинги программы.

Листинги допускается оформлять с использованием иного типа и размера шрифта, чем основной текст пояснительной записки.

Графическая часть представлена в виде UML-диаграмм.

По согласованию с руководителем курсового проекта могут составляться следующие диаграммы: «Диаграмма классов», содержащая статическое представление, используемое в области программирования для описания структуры системы, представления ее классов, атрибутов, операций и отношений между объектами; «Диаграмма состояний», которая представляет интерфейс пользователя и содержит список внутренних действий, выполняемых моделируемыми элементами системы при нахождении их в том или ином состоянии; «Диаграмма последовательности», отражающая в динамике взаимодействие объектов информационной системы. Примеры выполнения графической части представлены в приложении Г.

4 Требования к оформлению курсового проекта

Общие требования

Курсовой проект должен быть выполнен на листах белой бумаги формата А4 (размер 210 × 297 мм). Разметка страницы (поля): верхнее – 2 см; нижнее – 2 см; левое – 2 см; правое – 2 см.

Расстояние от края до верхнего колонтитула – 2 см; до нижнего – 1 см.

Необходимо обратить внимание на то обстоятельство, что размещение текста на странице зависит от разрешения установленного в системе принтера, поэтому перед окончательной версткой работы необходимо установить в качестве принтера по умолчанию лазерный принтер, настроенный на разрешение 600 dpi. Необходимо соблюдать равномерную плотность, контрастность и четкость изображения.

4.1 Текст

4.1.1 Текст основной части делят на главы, пункты (параграфы) и – при необходимости – на подпункты. Каждую главу следует начинать с нового листа.

4.1.2 Основной текст работы должен быть напечатан шрифтом Times New Roman, 14 pt, межстрочный интервал – «Одинарный», выравнивание – по ширине, отступ – 1,0 см, запрет висячих строк.

4.1.3 Величина шрифта, необходимость выделения полужирным, курсивом и пр., а также формат абзацев для различных элементов работы указаны в ГОСТ 2.105–95.

4.1.4 Все неуказанные параметры форматирования должны быть равны нулю. Допускается вставка в текст специальных символов (с использованием шрифтов Symbol и Wingdings), а также полученных штатными средствами редактора формул, таблиц и рисунков.

4.1.5 Листинги должны соответствовать формату А4. Распечатки включаются в общую нумерацию страниц работы и помещаются в приложениях.

4.1.6 Опечатки, описки и графические неточности, обнаруженные в процессе подготовки, допускается исправлять закрашиванием белой краской и нанесением на том же месте исправленного текста (графики) черной пастой от руки. Повреждения листов, помарки, следы не полностью удаленного текста не допускаются.

4.1.7 Сокращение слов в тексте не допускается. Разрешается употреблять аббревиатуры. Применять аббревиатуры можно только после разъяснения их значений. Разъяснение аббревиатуры достаточно дать один раз, при первом ее употреблении, причем сначала записывается полная расшифровка аббревиатуры, а затем в круглых скобках записывается аббревиатура.

Аббревиатуры целесообразно вводить в тех случаях, если они используются многократно. Аббревиатурам, установленным в государственных стандартах, расшифровку можно не давать. Например, ЭВМ, БРУ и т. д. (прописными буквами без точек).

4.2 Таблицы

4.2.1 Каждая таблица должна иметь заголовок. Заголовок и слово «Таблица» (без кавычек в тексте) начинают с прописной буквы. Заголовки не подчеркивают.

4.2.2 Заголовок таблицы должен быть максимально точным, простым, в единственном числе и именительном падеже.

4.2.3 Таблицы нумеруют последовательно арабскими цифрами (за исключением таблиц, приведенных в приложении) в пределах раздела. В левом верхнем углу таблицы по абзацному отступу перед заголовком таблицы помещают надпись «Таблица» с указанием номера таблицы. Номер таблицы должен состоять из номера раздела и порядкового номера таблицы, разделенных точкой, например: «Таблица 1.2» (вторая таблица первого раздела). Далее через тире записывают название таблицы. Например:

Таблица 1.1 – Характеристики ПК (точка в конце не ставится).

Если в тексте работы только одна таблица, то номер ей не присваивают и слово «Таблица» не пишут.

4.2.4 Заголовки граф таблиц должны начинаться с прописных букв, подзаголовки – со строчных, если они составляют одно предложение с заголовком, и с прописных, если они самостоятельные.

4.2.5 Таблицу размещают после первого упоминания о ней в тексте таким образом, чтобы ее можно было читать без поворота курсовой работы или с поворотом по часовой стрелке.

4.2.6 Таблицу с большим количеством строк допускается переносить на другой лист. При переносе таблицы на другой лист слово «Таблица» и ее номер с заголовком указывают один раз над первой частью таблицы; над другими частями пишут слово «Продолжение» и номер таблицы, например:

Продолжение таблицы 1.2.

Если шапка таблицы громоздкая, ее не дублируют, а пронумеровывают графы и повторяют их нумерацию на следующей странице.

4.2.7 Таблицу с большим количеством граф допускается делить на части и помещать одну часть под другой в пределах одной страницы. Если строки или графы таблицы выходят за формат таблицы, то в первом случае в каждой части таблицы повторяется ее заголовок, во втором случае – боковик.

4.2.8 Все цифровые данные должны быть логически однородными и сопоставимыми, основу их группировки должны составлять существенные признаки.

4.2.9 Если цифровые или иные данные в какой-либо строке таблицы не приводят, то в ней ставят прочерк.

4.2.10 Если цифровые данные в графах таблицы выражены в различных единицах физических величин, то их указывают в заголовке каждой графы. Если все параметры, размещенные в таблице, выражены в одной и той же единице физической величины, сокращенное обозначение единицы физической величины помещают над таблицей.

Если все данные в строке приведены для одной физической величины, то единицу физической величины указывают в соответствующей строке боковика таблицы.

4.2.11 Слова «более», «не более», «менее», «не менее», «в пределах» следует помещать рядом с наименованием соответствующего параметра или показателя (после единицы физической величины) в боковике таблицы или в заголовке графы.

4.2.12 Цифры в графах располагают так, чтобы классы чисел во всей графе были точно один под другим. Исключение составляют числа с интервалами величин. Числовые значения в одной графе должны иметь одинаковое количество десятичных знаков, исключение составляют числа с интервалами величин.

4.2.13 Для сокращения текста заголовков и подзаголовков граф отдельные понятия заменяют буквенными обозначениями, если они пояснены в тексте или приведены на иллюстрациях. Например, D – диаметр, H – высота и т. д.

Показатели с одним и тем же буквенным обозначением группируют последовательно, в порядке возрастания индекса.

4.3 Иллюстрации

4.3.1 Рисунки должны быть выполнены при помощи любого стандартного графического редактора или средствами MS Word.

4.3.2 Иллюстрации должны быть расположены так, чтобы их было удобно рассматривать без поворота курсового проекта или с поворотом по часовой стрелке. Иллюстрации располагают после первой ссылки на них.

4.3.3 Иллюстрации обозначаются словом «Рисунок» и нумеруются последовательно арабскими цифрами в пределах раздела, за исключением иллюстраций, приведенных в приложении.

Номер иллюстрации должен состоять из номера раздела и порядкового номера иллюстрации, разделенных точкой. Например:

Рисунок 1.2 (второй рисунок первого раздела).

4.3.4 Иллюстрации должны иметь наименование. При необходимости иллюстрации снабжают поясняющими данными (подрисуночный текст). Наименование иллюстрации помещают под ней – Times New Roman, 12 pt, выравнивание по центру под рисунком, межстрочный интервал – «Одинарный».

4.4 Формулы

4.4.1 Формулы следует оформлять с помощью встроенного объекта в текстовый редактор MS Word средства, которое можно вызвать на вкладке «Вставка» → «Объект» → MS Equation 3.0 или MathType.

Рекомендуется оставить для формул установки по умолчанию.

4.4.2 Пояснение значений символов и числовых коэффициентов следует приводить непосредственно под формулой в той же последовательности, в какой они даны в формуле. Значение каждого символа и числового коэффициента следует давать с новой строки. Первую строку объяснения начинают со слова «где» без двоеточия, без абзачного отступа.

4.4.3 Уравнения и формулы следует выделять из текста: интервал перед – не менее 6 пт, после – не менее 6 пт. Если уравнение не умещается на одну строку, оно должно быть перенесено после знака равенства (=) или после знаков плюс (+), минус (-), умножения (х) и деления (:).

4.4.4 Небольшие и не имеющие принципиального значения формулы можно размещать по тексту. Те формулы, на которые придется ссылаться в дальнейшем, следует пронумеровать, а те, на которые ссылок не будет, нумеровать не нужно, чтобы не загромождать текст. Следующие друг за другом формулы разделяются между собой запятой или точкой с запятой.

4.4.5 Формулы (если их более одной) нумеруют арабскими цифрами в пределах раздела. Номер формулы состоит из номера раздела и порядкового номера формулы в разделе, разделенных точкой. Номер указывают с правой стороны листа на уровне формулы в круглых скобках, например: (3.1) – первая формула третьего раздела. Если номер не умещается в одной строке с формулой, то его располагают в следующей строке ниже формулы. При переносе формулы ее номер ставится на уровне последней строки. Если особенно важная формула заключена в рамку, то ее номер находится в правом краю против основной строки формулы. Номер формулы-доби располагают на середине основной горизонтальной черты формулы.

4.5 Нумерация

4.5.1 Страницы курсовой работы нумеруют арабскими цифрами. На титульном листе и содержании номер не ставится. Лист задания на курсовую работу не включают в общую нумерацию работы. Номера страниц проставляются снизу по центру, начиная со страницы с введением.

4.5.2 Главы должны иметь порядковую нумерацию в пределах всей курсовой работы и обозначаться арабскими цифрами. Содержание, введение, заключение и список использованных источников не нумеруются.

4.5.3 Содержащиеся в тексте пункта или подпункта перечисления требований, указаний, положений записывают после двоеточий и обозначают арабскими цифрами со скобкой или дефисом. Перед двоеточием должно стоять обобщающее слово.

Перечисления записываются с абзаца. В конце каждого перечисления, если за ним следует еще перечисление, ставят точку с запятой.

4.5.4 Иллюстрации (таблицы, чертежи, схемы, графики), которые расположены на отдельных страницах курсового проекта, включают в общую нумерацию страниц. Таблицу, рисунок или чертеж, размеры которого больше формата А4, учитывают как одну страницу. Листы формата более А4 помещают в конце курсовой работы после заключения в порядке их упоминания в тексте.

4.5.5 Примечание к тексту и таблицам, в которых указывают справочные и поясняющие данные, нумеруют последовательно арабскими цифрами. Если примечаний несколько, то после слова «Примечания» ставят двоеточие, например: «Примечания:». Если имеется одно примечание, то его не нумеруют и после слова «Примечание» ставят точку.

4.6 Приложения

4.6.1 Приложения оформляют как продолжение курсового проекта на последующих его страницах, располагая их в порядке появления ссылок в тексте.

4.6.2 Каждое приложение должно начинаться с нового листа с указанием по центру слова «Приложение», напечатанного прописными буквами, и иметь содержательный заголовок – Times New Roman, 16 pt, отступ первая 0, выравнивание по центру.

4.6.3 Если в курсовой работе более одного приложения, их именуют последовательно прописными буквами алфавита русского языка, например: Приложение А, Приложение Б и т. д.

4.6.4 Рисунки, таблицы и формулы, помещаемые в приложении, нумеруют арабскими цифрами в пределах каждого приложения, например: «Рисунок А.1» (первый рисунок приложения А); «Таблица А.2» (вторая таблица приложения А).

5 Организация выполнения курсового проекта

Продолжительность выполнения курсового проекта – 14 недель. Каждый студент выполняет индивидуальное задание, оговоренное с преподавателем.

Курсовой проект включает выбор (в зависимости приложения) типа сетевого протокола, модели ввода-вывода, типа приложения (однопоточное/многопоточное). Реализацию программы можно проводить с использованием языков: HTML, JavaScript, Java, C# или C++.

Графическая часть курсового проекта должна иметь прямое отношение к теме проекта. Схемы и другие графические материалы должны быть подобраны так, чтобы в совокупности с докладом раскрывали сущность проделанной студентом работы.

Графическая часть, включающая схемы, таблицы, рисунки, плакаты и т. д., представляется на листах формата А3 и может быть выполнена с использованием чертежных принадлежностей или путем распечатки на плоттере.

Каждый студент обязан посетить не менее пяти консультаций по вопросам выполнения курсового проекта, предъявляя на предварительный просмотр преподавателю отдельные результаты курсового проектирования или проект пояснительной записки. По результатам курсового проекта студентом в соответствии с требованиями оформляется пояснительная записка и сдается преподавателю на проверку. Если все недочеты, выявленные при консультациях, устранены, студент защищает курсовой проект, по результатам защиты выставляется итоговая оценка. Если пояснительная записка не удовлетворяет поставленным требованиям, то она возвращается студенту на доработку.

Каждый этап может представляться на проверку преподавателю по частям во время консультаций. Консультации проводятся не чаще чем 2 раза в неделю.

6 Защита курсового проекта

Защита курсового проекта является особой формой проверки глубины осмысления изученного материала, эффективным способом выработки и демонстрации навыков научных выступлений, умения кратко, аргументировано и четко излагать представленный в курсовом проекте материал. Для более наглядного представления полученных результатов студентам предлагается подготовить презентацию работы с использованием мультимедийных технических средств при защите проекта.

Защита курсового проекта принимается комиссией в составе двух-трех человек. В состав комиссии входит непосредственный руководитель курсового проекта.

Формой аттестации по курсовому проекту является дифференцированный зачет и определяется по десятибалльной шкале оценки знаний. Оценка за курсовой проект выставляется по результатам защиты в ведомость и зачетную книжку студента (неудовлетворительная оценка – только в ведомость) и заверяется подписями членов комиссии.

Список литературы

- 1 Общие требования к текстовым документам : ГОСТ 2.105–95 ЕСКД. – Мн., 1995. – 37 с.
- 2 **Гуриков, С. Р.** Информатика : учебник / С. Р. Гуриков. – М. : ФОРУМ ; ИНФРА-М, 2018. – 463 с.
- 3 **Шилдт, Г.** Java 8. Полное руководство : пер. с англ. / Г. Шилдт. – М. : Вильямс, 2018. – 1375 с.
- 4 **Флэнаган, Д.** JavaScript: карманный справочник : пер. с англ. / Д. Флэнаган. – 3-е изд. – М. : Вильямс, 2015. – 314 с.
- 5 **Курняван, Б.** Программирование WEB-приложений на языке Java : пер. с англ. / Б. Курняван. – М. : Лори, 2014. – 880 с.
- 6 **Дадян, Э. Г.** Методы, модели, средства хранения и обработки данных : учебник / Э. Г. Дадян, Ю. А. Зеленков. – М. : Вузовский учебник ; ИНФРА-М, 2017. – 168 с.
- 7 **Арлоу, Д.** UML и унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Д. Арлоу, А. Нейштадт. – М. : Символ, 2016. – 624 с.

Приложение А (рекомендуемое)

Примеры индивидуальных заданий

1 Разработка сетевого приложения для автоматизации работы хозрасчетной поликлиники.

Задание

Решаемые задачи: расписание врачей, процедурных кабинетов, учет предоставленных услуг как бесплатных, так и при медицинской страховке, при превышении лимита услуги могут предоставляться за плату.

Запросы:

- список пациентов по участкам;
- поиск данных о пациенте по номеру карты;
- количество пациентов, принятых заданным врачом в текущий день.

Отчеты:

- расписание врачей;
- расписание процедурных кабинетов.

2 Разработка сетевого приложения для автоматизации работы издательства.

Задание

Издательство – предприятие, занимающееся выпуском разнообразной печатной продукции. Издательство заключает договор с заказчиком (клиентом) на выполнение заказа. Заказчиком может выступать частное лицо или организация. Частное лицо может быть автором издания (или одним из авторов, если их несколько) или представителем автора. Организация для контактов с издательством также имеет своего представителя – контактную персону.

Заказ может быть книгой, брошюрой, рекламным проспектом, буклетом, бюллетенем для голосования или каким-либо другим видом издательской продукции. Подготовленные издательством материалы заказчика печатаются в типографиях, где издательство размещает свои заказы. Информацию о работе издательства можно сгруппировать следующим образом:

- сведения о заказчиках (частное лицо или организация, личные данные контактной персоны, адрес, телефон, факс);
- сведения о заказах (номер заказа, заказчик, вид печатной продукции, издание, типография, дата приема заказа, отметка о выполнении, дата выполнения заказа);
- сведения об изданиях (код издания, автор и название, объем в печатных листах, тираж, номер заказа);
- сведения об авторах (личные данные: ФИО, домашний адрес, телефон; дополнительные сведения);
- сведения о типографиях (название, адрес, телефон).

3 Разработка сетевого приложения для автоматизации работы учебного отдела университета.

Задание

В учебной части высшего учебного заведения ведется учет хода учебного процесса. Учету подлежат студенты и студенческие группы, преподаватели, предметы, а также оценки, получаемые студентами при сдаче экзаменов. Студент может сдавать экзамены по одному и тому же предмету несколько раз (например, если предмет читается несколько семестров или студент пересдает экзамен).

На основании имеющейся информации необходимо иметь возможность: просматривать списки студентов, сдавших сессию с тройками, без троек и на отлично; просматривать списки групп и ведомости групп по отдельным дисциплинам; формировать отчет об успеваемости студентов университета по отдельным дисциплинам и рейтинг студентов по их успеваемости.

4 Разработка сетевого приложения для автоматизации работы кондитерской фабрики.

Задание

Кондитерская фабрика производит шоколадные конфеты в коробках. Согласно технологическим условиям, для производства единицы каждого вида продукции необходимо определенное количество компонентов. Компоненты фабрика закупает у ряда поставщиков. Фабрика принимает заказы на свою продукцию, в соответствии с которыми и планирует производство и закупку компонентов.

Для повышения производительности труда планово-финансового отдела, отдела поставок и сбыта необходимо автоматизировать формирование следующих документов по данным за указанный период времени:

- суммы заказов;
- необходимые объемы компонентов;
- заказы на компоненты по поставщикам.

5 Разработка сетевого приложения для автоматизации работы гостиницы.

Задание

Решаемые задачи: учет и планирование загрузки помещений (свободные, занятые, забронированные номера: сколько их на текущую дату). Описание номеров. Предусмотреть возможность бронирования номеров не более чем за 30 дней. В плане должно быть: сотрудничество с туристической фирмой по размещению групп туристов с полупансионом (завтрак – обед, обед – ужин, завтрак – ужин). Расчет общей стоимости услуг гостиницы. Частные клиенты тоже могут использовать полупансион.

6 Разработка сетевого приложения для автоматизации работы оптового склада.

Задание

Склад осуществляет продажу товаров оптом. Любая фирма, занимающаяся продажей товаров в розницу, закупает необходимые ей товары на складе, который служит посредником между производителями и продавцами.

На склад товар поступает от некоторой фирмы-поставщика, в свою очередь склад продает товар фирме-покупателю, заключая с ней сделку о продаже товара.

Деятельность оптового склада характеризуется следующей информацией, которую можно объединить в группы следующим образом:

- поставщики (код поставщика, название фирмы-поставщика, полный адрес, телефон);
- покупатели (код покупателя, название фирмы-покупателя, полный адрес, телефон);
- товар на складе (код товара, поставщик, название товара, единицы измерения, количество, цена покупки за единицу товара, цена продажи за единицу товара);
- сделки о продаже (код товара, поставщик, покупатель, количество проданного товара).

На основании описанных данных необходимо вести учет поставщиков, покупателей, продаж, движения товара на складе.

7 Разработка сетевого приложения для автоматизации работы туристического агентства.

Задание

В качестве предметной области (ПО) рассматривается туристическое агентство, предоставляющее услуги организации отдыха и курортного лечения за рубежом. База данных должна хранить данные о фирме, предоставляющей услуги по туризму, данные о клиентах, о предоставляемых маршрутах, о системе скидок на услуги в зависимости от социального положения клиентов.

База данных должна обеспечивать ведение учета маршрутов, по которым агентство организует туры, облегчать поиск необходимых данных, таких как справочник маршрутов, список клиентов по турам и др., оценка рентабельности маршрутов.

Клиенты заключают с фирмой договор, в котором оговариваются условия отдыха. В прайс-листе указываются стоимости маршрутов (стран, городов). Некоторым категориям клиентов (студентам, пенсионерам, детям и т. д.) предоставляются скидки.

8 Разработка сетевого приложения для автоматизации работы юридической фирмы.

Задание

Фирма производит операции по купле-продаже и аренде, оформлению документов и консультации юристов.

Система учета работы и поощрения сотрудников предусматривает оценку деятельности сотрудников и, в зависимости от ее успешности, устанавливает объем поощрений.

Основная работа выполняется агентами, которые являются посредниками между продавцами и покупателями.

Деятельностью фирмы руководит директор, который оформляет все необходимые документы, бухгалтер ведет бухгалтерский учет в фирме, а юристы дают консультации по вопросам, касающимся деятельности фирмы.

Приложение Б (рекомендуемое)

Примеры используемых блоков кода

При работе с базой данных необходимо совершить подключение к базе данных посредством функционала класса DriverManager из библиотеки JDK.

Код подключения к базе данных, находящейся на локальном сервере, представлен в листинге Б.1.

```
public Connection getConnection()
{
    String url =
"jdbc:mysql://localhost/db?serverTimezone=Europe/Moscow$useSSL=false";
    String username = "root";
    String password = "password";
    Connection con = null;
    try {
        con = DriverManager.getConnection(url,username,password);
        return con;
    }
    catch (SQLException ex) {
        Logger.getLogger(main_form.ciass.getName()).log(Level.SEVERE,null,ex);
        return null;
    } }
}
```

Листинг Б.1 – Код подключения к базе данных

Примеры реализации соединения для передачи информации между клиентской и серверной частями посредством сокетов в Java представлены в листингах Б.2 и Б.3.

```
// A Java program for a ClientSide
import java.net.*;
import java.io.*;
public class ClientProgram {
    // инициализация сокета и входного / выходного потоков
    private Socket socket = null;
    private DataInputStream input = null;
    private DataOutputStream out = null;
    // конструктор с ip адресом и портом
    public Client(String address, int port)
    {
        // установка соединения
```

```

try {
    socket = new Socket(address, port);
    System.out.println("Connected");
    // прием ввода с терминала
    input = new DataInputStream(System.in);
    // отправка вывода в сокет
    out = new DataOutputStream(socket.getOutputStream());
}
catch(UnknownHostException u)
{
    System.out.println(u);
}
catch(IOException i)
{
    System.out.println(i); }
// строка для чтения сообщения из ввода
String line = "";
// продолжается чтение, пока не будет введено "Over"
while (!line.equals("Over"))
{
    try {
        line = input.readLine();
        out.writeUTF(line);
    }
    catch(IOException i)
    {
        System.out.println(i);
    } }
// закрытие соединения
try {
    input.close();
    out.close();
    socket.close();
}
catch(IOException i)
{
    System.out.println(i);
} }
public static void main(String args[]) {
    Client client = new Client("127.0.0.1", 5000);
} }

```

Листинг Б.2 – Пример реализации соединения на стороне клиента

```

// A Java program for a Serverside
import java.net.*;
import java.io.*;
public class ServerSide {
//инициализация сокета и входного потока
private Socket socket = null;
private ServerSocket server = null;
private DataInputStream in = null;
// конструктор с портом
public Server(int port) {
// запуск сервера и ожидание соединения
try {
server = new ServerSocket(port);
System.out.println("Server started");
System.out.println("Waiting for a client ...");
socket = server.accept();
System.out.println("Client accepted");
// принимает входные данные из клиентского сокета
in = new DataInputStream(
new BufferedInputStream(socket.getInputStream()));
String line = "";
// чтение сообщения из клиента пока не отправлено "Over"
while (!line.equals("Over")) {
try {
line = in.readUTF();
System.out.println(line);
}
catch(IOException i)
{
System.out.println(i);
} }
System.out.println("Closing connection");
// закрытие соединения
socket.close();
in.close();
}
catch(IOException i){
System.out.println(i);
} }
public static void main(String args[]){
Server server = new Server(5000);
} }

```

Листинг Б.3 – Пример реализации соединения на стороне сервера

**Приложение В
(рекомендуемое)**

Пример оформления титульного листа

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

Тема курсового проекта

Курсовой проект
по дисциплине «Программирование сетевых приложений»

Специальность 1-40 05 01 Информационные системы и технологии
(по направлениям)

Направление специальности 1-40 05 01-1 Информационные системы
и технологии (в проектировании и производстве)

Проверил
должность преподавателя
_____ ФИО преподавателя
«__» _____ 20__ г.

Выполнил
студент группы
_____ ФИО студента
«__» _____ 20__ г.

Приложение Г (рекомендуемое)

Примеры графической части курсового проекта

На рисунках Г.1–Г.3 представлены примеры диаграмм, составляющие графическую часть курсового проекта.

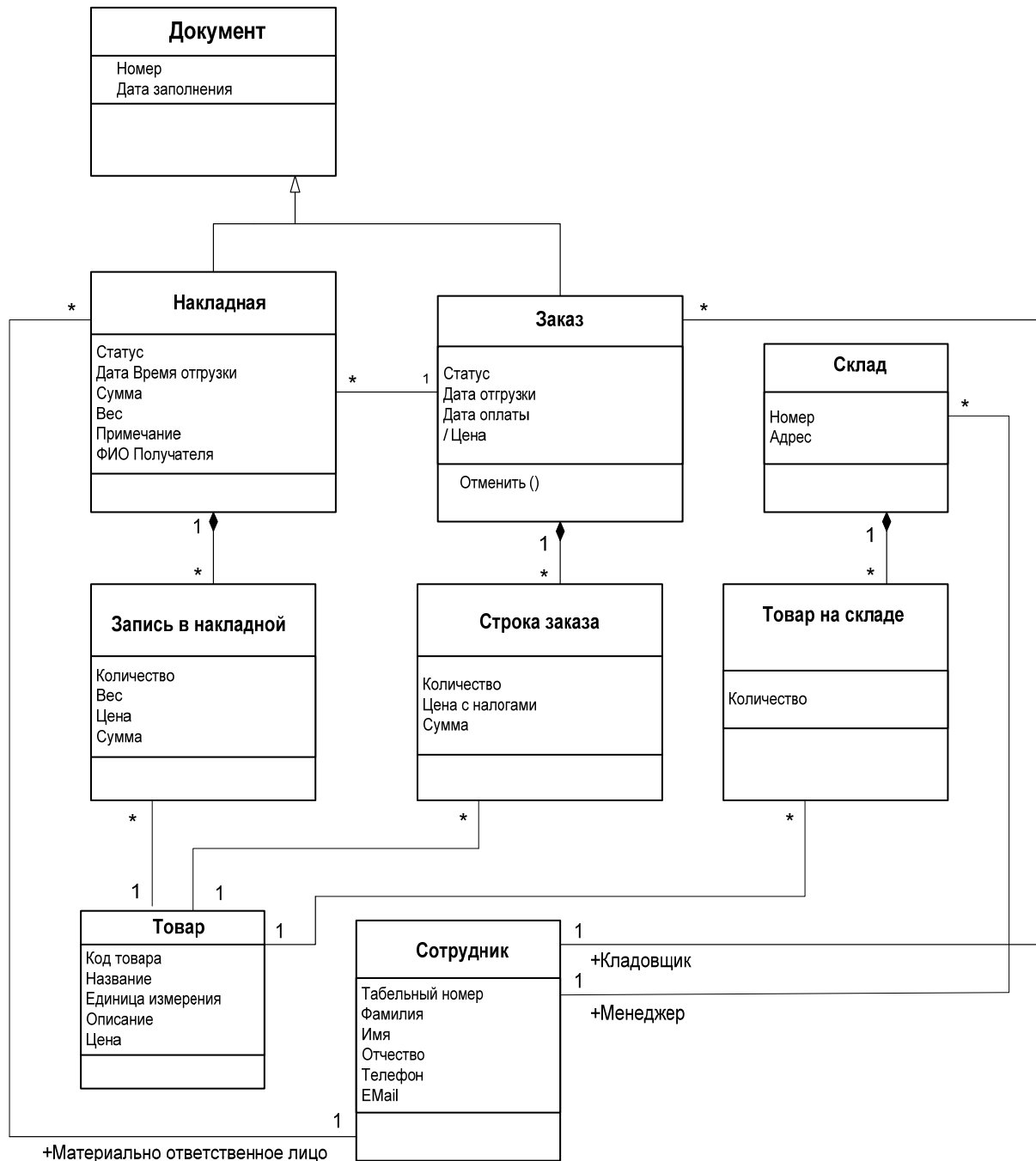


Рисунок Г.1 – Диаграмма классов

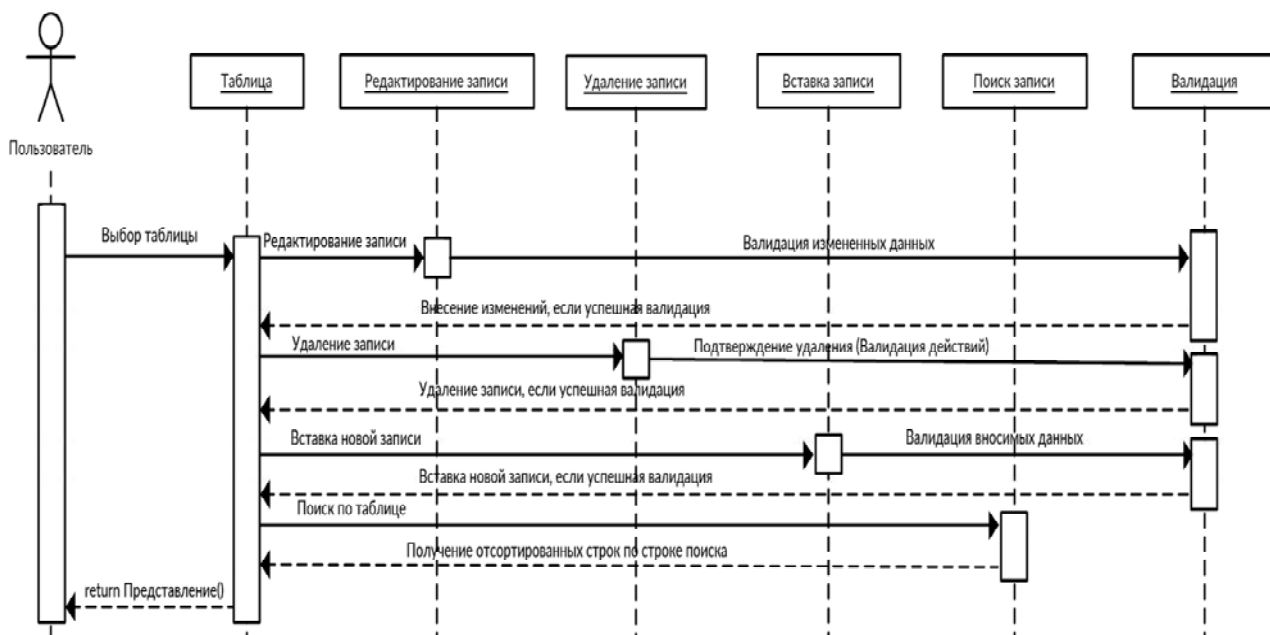


Рисунок Г.2 – Диаграмма последовательности

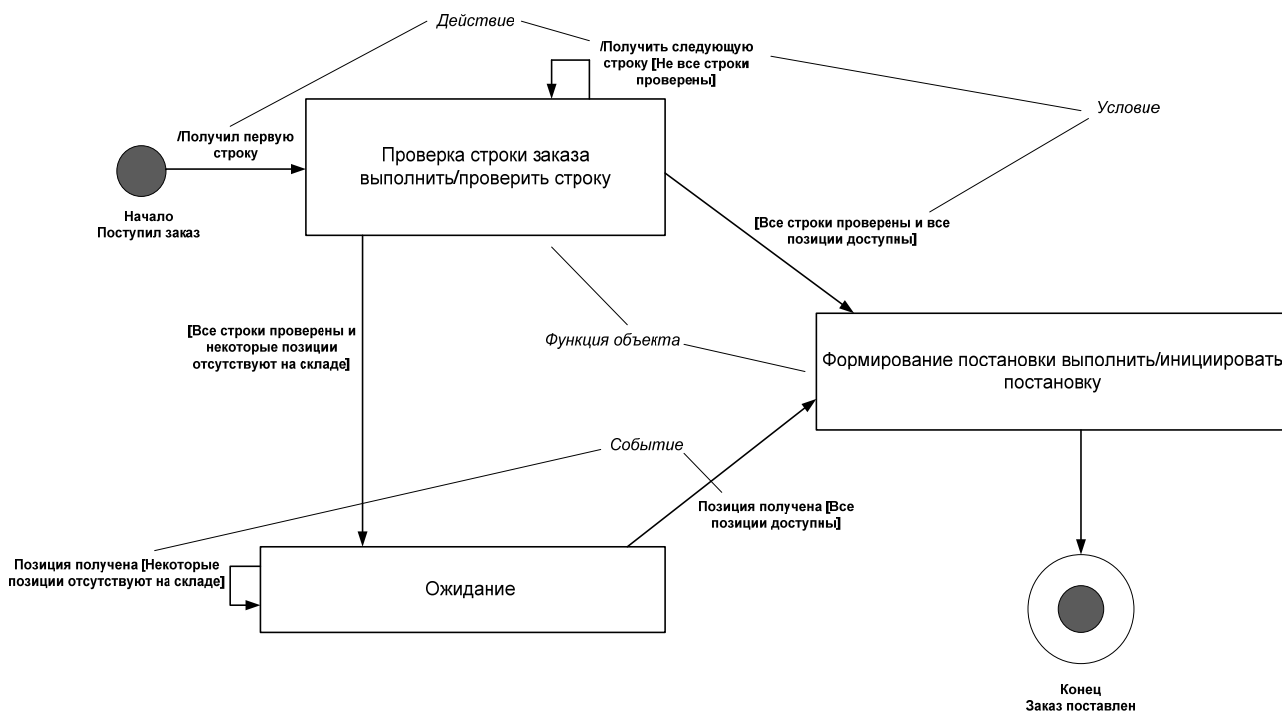


Рисунок Г.3 – Диаграмма состояний