

Министерство цифрового развития, связи и массовых коммуникаций РФ  
Уральский технический институт связи и информатики (филиал) ФГБОУ ВО  
"Сибирский государственный университет телекоммуникаций и  
информатики" в г. Екатеринбурге (УрТИСИ СибГУТИ)



Уральский технический  
институт связи  
и информатики

### ОТЧЕТ

По дисциплине «Сетевое программирование»  
Практическое занятие №5  
«Разработка блок-схемы работы приложения»

Выполнила:

студентка группы ПЕ-216

Морос Е.Е.

Проверил:

преподаватель

Бурумбаев Д.И.

## 1 Цель работы:

- 1.1 Научиться составлять блок-схемы работы приложений;
- 1.2 Закрепить знания по теме «Разработка блок-схемы работы приложения».

## 2 Перечень оборудования:

- 2.1 Персональный компьютер;
- 2.2 Microsoft Office;
- 2.3 Графический редактор.

## 3.Ход работы:

3.1. Разработка блок-схемы приложения и ERD (Entity-Relationship Diagram, диаграммы "сущность-связь") тесно связаны, но отвечают за разные аспекты проектирования. Вот основные моменты их взаимосвязи:

### 1. Блок-схема приложения

Показывает логику работы приложения, последовательность действий, обработку данных, ключевые процессы и переходы между ними.

Чаще используется для визуализации бизнес-логики или логики программного модуля (например, как пользователь проходит регистрацию или делает заказ).

### 2. ERD (диаграмма "сущность-связь")

Показывает структуру данных — какие сущности есть в предметной области (например, Пользователь, Заказ, Продукт) и как они связаны между собой. Используется для проектирования базы данных приложения.

Как они связаны?

### 1. Обеспечивают целостное проектирование

Блок-схема описывает, как данные будут протекать через ваше приложение (например, пользователь регистрируется → создаётся запись в таблице User).

ERD описывает структуру этих данных (например, какие поля есть у сущности User, её связи с другими сущностями).

### 2. Взаимное влияние

Проектируя блок-схемы, вы выявляете, какие объекты и действия важны — это помогает определить нужные сущности и связи в ERD.

При построении ERD могут появиться новые детали (например, требуются дополнительные сущности или связи), которые могут скорректировать или уточнить логику ваших блок-схем.

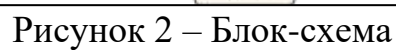
### 3. От бизнес-процессов к данным

Сначала разрабатывается блок-схема (понимание, какие процессы нужны). Затем на основании выявленных процессов определяются необходимые сущности и отношения между ними — и строится ERD.

### 4. Реализация

Разработчик, видя блок-схему, понимает, куда и как обращаться к данным. Видя ERD, понимает, откуда брать эти данные и как их хранить.

## Products – Warehouse Locations: 1:1



#### 4. Контрольные вопросы:

##### 4.1. Что такое блок-схема и для чего она используется в разработке приложений?

Блок-схема — это графическое представление алгоритма или процесса с помощью стандартных символов (блоков), соединённых линиями. Она используется для наглядного описания логики работы приложения, упрощённого понимания и анализа сложных процессов, планирования архитектуры, и выявления логических ошибок ещё до написания кода.

##### 4.2. Какие основные элементы используются при создании блок-схемы, и что они представляют?

- Основные элементы блок-схемы включают:
- Начало/Конец: (овалы) — начало или завершение процесса.
- Процесс/Действие: (прямоугольники) — выполнение какого-либо действия или этапа.
- Условие/Решение: (ромбы) — проверка условия и выбор направления выполнения.
- Ввод/Вывод: (параллелограммы) — операции ввода и вывода информации.
- Поток управления: (стрелки) — направление движения потока процесса.

##### 4.3. Какие стандартные символы используются для обозначения различных операций, принятия решений и потоков данных в блок-схеме?

- Овал — начало и конец (Start/End).
- Прямоугольник — операция или действие (Process/Action).
- Ромб — проверка условия/выбор (Decision).
- Параллелограмм — ввод или вывод данных (Input/Output).
- Стрелки — направление потока управления.

##### 4.4. Каковы основные этапы создания блок-схемы для работы приложения?

- Определение задачи и постановка цели.
- Разделение процесса на логические шаги.
- Выбор и запись последовательности операций и условий.
- Изображение каждого шага в виде соответствующего символа.
- Соединение блоков стрелками по логике процесса.
- Проверка схемы на полноту и отсутствие логических ошибок.
- Переработка и оптимизация по необходимости.

##### 4.5. Как можно использовать блок-схему для документации и обсуждения логики работы приложения с другими участниками проекта?

Блок-схема служит общим наглядным языком для всех участников (разработчиков, аналитиков, тестировщиков). Схема помогает легче объяснять, обсуждать и анализировать алгоритмы, выносить изменения и фиксировать логику работы приложения в документации проекта.

4.6. Какие преимущества предоставляет использование блок-схем при разработке приложений?

- Простота восприятия и обсуждения даже без глубоких технических знаний.
- Раннее выявление логических ошибок.
- Облегчение коммуникации внутри команды.
- Быстрое обучение новых участников проекта.
- Упрощение сопровождения и тестирования приложений.

4.7. Каковы основные правила построения понятной и эффективной блок-схемы для работы приложения?

- Ясность и однозначность блоков и направлений стрелок.
- Использование стандартных обозначений.
- Логичная последовательность шагов.
- Минимизация перекрёстных линий.
- Компактность и структурированность (разделение на модули при необходимости).
- Хорошая читаемость (текст в блоках короткий и понятный).

4.8. Какие инструменты могут использоваться для создания блок-схем, и как выбрать подходящий инструмент для конкретного проекта?

Популярные инструменты:

- Онлайн-сервисы: Lucidchart, draw.io (diagrams.net), Creately.
- Офисные пакеты: Microsoft Visio, Google Drawings, LibreOffice Draw.
- Специализированные ПО: yEd, SmartDraw, OmniGraffle (Mac).

Выбор зависит от:

- Требуемых функций (совместная работа, экспорт/импорт форматов),
- Сложности проекта,
- Бюджета (есть бесплатные и платные опции),
- Предпочтений и опыта команды.

4.9. Как блок-схема может быть использована для выявления и исправления ошибок в логике работы приложения до начала разработки кода?

С помощью её визуализации можно легко заметить нелогичные переходы, заикливания, пропущенные условия или неохваченные сценарии. Исправление ошибок на этапе блок-схемы дешевле и быстрее, чем после написания кода.

4.10. Каким образом блок-схема может помочь понять последовательность выполнения операций и потенциальные точки улучшения в работе приложения?

Блок-схема отображает пошаговую логику. Это помогает:

- Выявить узкие места и дубликаты операций,
- Оптимизировать последовательность действий,
- Найти излишние/избыточные действия,
- Определить, где возможны параллелизация и автоматизация процессов.