

-

СЦЕНАРИЙ ВИДЕО: "КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ 'ТЕЛЕФОННАЯ КНИГА' НА JAVA С НУЛЯ"

ЧАСТЬ 1: ВВЕДЕНИЕ (3 минуты)

0:00-3:00

"Привет, друзья! Сегодня мы создадим с нуля полноценное консольное приложение 'Телефонная книга' на Java."

"Это идеальный проект для начинающих, который позволит вам перейти от основ процедурного кода к объектно-ориентированному программированию (ООП), научит работать с коллекциями, пользовательским вводом и, самое главное, с файловым вводом/выводом (I/O) для сохранения данных!"

"Телефонная книга — это отличный способ прокачать навыки Java, научиться хранить данные в памяти и на диске, структурировать большой проект через классы и создавать персистентные (сохраняющие данные после закрытия) приложения."

Наше приложение будет уметь:

- Принимать выбор пользователя через консольное меню.
- Добавлять новые контакты (Имя, Телефон).
- Просматривать весь список контактов.
- Искать контакты по части имени (без учета регистра).
- Сохранять и загружать данные контактов из текстового файла.
- Обрабатывать некорректный ввод пользователя.

"Всё это мы напишем в трех классах: Contact, PhoneBookManager и Main, чтобы вы увидели, как организовать структуру настоящего приложения!"

ЧАСТЬ 2: ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ (3 минуты)

3:00-6:00

"Мы используем только стандартные возможности Java и ее библиотеки I/O. Всё максимально просто и нативно."

Что применим:

- Классы ООП: Contact, PhoneBookManager, Main для четкой структуры.

- **Scanner** для интерактивного взаимодействия с пользователем и получения ввода.
- **ArrayList<Contact>** (интерфейс List) для хранения контактов в памяти.
- **File, FileWriter, PrintWriter** для сохранения данных в файл (phonebook.txt).
- **FileReader, BufferedReader** для загрузки данных из файла.
- Циклы **while** для основного меню и перебора контактов.
- Оператор **switch** для чистой реализации меню.
- Stream API (Java 8+) для эффективного и современного поиска/фильтрации контактов.
- Блок **try-with-resources** для безопасной работы с файлами и предотвращения утечек.

"Мы создадим компактное, но полнофункциональное приложение, которое демонстрирует все ключевые концепции начального ООП и персистентности на Java."

ЧАСТЬ 3: ЧЕМУ ВЫ НАУЧИТЕСЬ (3 минуты)

6:00-9:00

"После просмотра видео вы получите не только работающую Телефонную книгу, но и глубокое понимание фундаментальных концепций Java."

Вы поймёте:

- Как применять принципы ООП (инкапсуляция) на практике через класс Contact.
- Как управлять состоянием приложения в центральном классе (PhoneBookManager).
- Как работать с коллекциями (List/ArrayList) для хранения объектов.
- Как использовать файловый ввод/вывод для сохранения данных между запусками программы.
- Как использовать Stream API для эффективной обработки коллекций (поиск/фильтрация).
- Как обрабатывать ошибки ввода пользователя и исключения при работе с файлами (try-catch).
- Как строить главный цикл приложения и реализовывать меню через switch.

"Это идеальный стартовый проект для практики структурного проектирования и основ персистентности данных на Java!"

ЧАСТЬ 4: ФУНКЦИОНАЛ ПРИЛОЖЕНИЯ (3 минуты)

9:00-12:00

"Мы создадим полноценную систему управления контактами с удобным интерфейсом и надежным хранением!"

Что будет у нашей программы:

- Интуитивное меню выбора (1. Добавить, 2. Просмотреть, 3. Найти, 4. Выход).
- ООП-Архитектура — разделение ответственности между классами (Модель, Менеджер, Интерфейс).
- Персистентность данных — контакты сохраняются в файл phonebook.txt при выходе и загружаются при запуске.
- Продвинутый поиск — поиск по части имени, без учета регистра (toLowerCase(), contains).
- Система Валидации — проверка на пустые поля при добавлении и на целостность данных при загрузке.
- Безопасный I/O — использование try-with-resources для гарантии закрытия файловых потоков.

"Мы создадим систему, которая не только работает, но и использует лучшие практики современного Java-кода!"

ЧАСТЬ 5: НАПИСАНИЕ КОДА (15 минут)

12:00-27:00

"Всё максимально понятно и структурировано. Мы не просто пишем код, мы разбираем его логику и алгоритмы."

Блок 1: Модель Данных (Contact.java)

- Создаём класс Contact и приватные поля name и phoneNumber.
- Пишем Конструктор для инициализации объекта.
- Реализуем Геттеры для безопасного доступа к полям (Инкапсуляция).

- Переопределяем метод `toString()` для красивого консольного вывода.
- Создаём метод `toFileDialog()` для подготовки строки к сохранению (разделение ;).

Блок 2: Менеджер и Бизнес-логика (PhoneBookManager.java)

- Создаём класс `PhoneBookManager` с приватной коллекцией `List<Contact>`.
- Пишем методы `addContact`, `viewAllContacts`.
- Реализуем сложный метод `findContact` с использованием Stream API (`filter` и `collect`).
- Вызываем `loadContactsFromFile()` в конструкторе, чтобы приложение было персистентным с самого старта.

Блок 3: Файловый Ввод/Вывод (I/O)

- Реализуем `saveContactsToFile()`: Используем `PrintWriter` внутри `try-with-resources`.
- Реализуем `loadContactsFromFile()`: Используем `BufferedReader` внутри `try-with-resources`.
- Добавляем логику парсинга строки (`split(FIELD_SEPARATOR)`) и проверки целостности данных при загрузке.

Блок 4: Главный класс и Интерфейс (Main.java)

- Создаём класс `Main` и метод `main`.
- Инициализируем `Scanner` и `PhoneBookManager`.
- Реализуем метод `displayMenu()`.
- Организуем главный цикл `while(running)`.
- Используем `switch` для обработки выбора пользователя.
- Добавляем валидацию ввода (`hasNextInt()`, `scanner.nextLine()` для очистки буфера).
- В методах `addNewContact` и `searchContact` реализуем получение данных от пользователя и вызываем методы менеджера.

"Каждый блок мы будем писать и сразу же тестировать, чтобы вы видели результат каждого этапа!"

ЧАСТЬ 6: ЗАВЕРШЕНИЕ (3 минуты)

27:00-30:00

"Подводим итоги: Мы научились создавать интерактивные консольные приложения на Java с использованием ООП, персистентности данных и современных инструментов, таких как Stream API."

- На практике разобрали работу с Scanner, List и File I/O (PrintWriter, BufferedReader).
- Реализовали полноценную, функциональную телефонную книгу с системой сохранения/загрузки.
- Создали надежную архитектуру с четким разделением ответственности между классами.

"Этот проект отличная основа для дальнейшего развития. Вы можете добавить функционал удаления/редактирования контактов, сортировку по имени или даже перевести это в графический интерфейс!"

"Если тебе понравился проект, поставь лайк и подпишись, впереди будет ещё больше практических проектов на Java!"

"С вами был (твоё имя), до встречи в следующем видео, где мы создадим новый полезный инструмент на Java!"

КЛЮЧЕВЫЕ АКЦЕНТЫ СЦЕНАРИЯ:

- Демонстрация базовых концепций:
 - ООП: Классы, Инкапсуляция (приватные поля, геттеры).
 - Коллекции: ArrayList для хранения объектов.
 - Персистентность (I/O): Сохранение/загрузка данных в/из файла.
 - Современный Java: try-with-resources и Stream API для поиска.
 - Управление потоком: Циклы (while) и условия (switch).
- Практическая польза:
 - Реальная работающая программа с сохранением данных.
 - Код, который можно сразу использовать и расширять.
 - Фундамент для более сложных проектов с работой с файлами.

- **Образовательная ценность:**

- Понимание работы с пользовательским вводом и меню.
- Навыки построения логики поиска и фильтрации.
- Принципы организации главного цикла приложения и I/O.
- Практика обработки ошибок и валидации данных.