**Лаба4.1**

В данном проекте реализована программа на языке C++, предназначенная для хранения, фильтрации и сортировки информации о сотрудниках. Программа использует объектно-ориентированный подход, а также активно применяет стандартную библиотеку шаблонов (STL), что делает её гибкой и расширяемой.

Основной класс BD инкапсулирует данные сотрудников, считанные из текстового файла. В файле ожидается, что каждая строка содержит имя с годом рождения, профессию и структурное подразделение, разделённые запятыми. Загруженные данные преобразуются в объекты структуры Employee, после чего могут быть отсортированы или отфильтрованы по различным критериям.

Для работы с фильтрами и сортировками активно применяются функциональные объекты (std::function) и лямбда-выражения. Это позволяет легко изменять логику фильтрации и сортировки без изменения основной логики программы. Например, используются фильтры для выбора только инженеров или исключения председателей, а также сортировка по году рождения.

Дополнительным плюсом реализации является модульность: методы класса BD разделяют функциональность на чтение файла, фильтрацию, сортировку и отображение, что упрощает поддержку и расширение кода. Функции getSorted, getWithExceptions и display являются хорошими примерами разделения ответственности.

В функции main демонстрируются два запроса: один выводит всех инженеров, второй — всех сотрудников, не являющихся председателями, отсортированных по году рождения. Такой подход облегчает адаптацию программы под другие запросы и критерии.

**Лаба4.2**

В данной работе реализована программа на языке C++, предназначенная для обработки и отображения библиографических ссылок из текстового файла. Основная цель программы — продемонстрировать навыки работы с файлами, строками, лямбда-выражениями и контейнерами стандартной библиотеки STL.  
Программа построена в объектно-ориентированной парадигме. Основным элементом является класс BD, отвечающий за хранение ссылок, их загрузку из файла и отображение в отсортированном виде. Внутри класса определены два вектора: references\_ru и references\_en, разделяющие ссылки на русскоязычные и англоязычные соответственно. Разделение происходит при загрузке файла по первым символам строк, что демонстрирует простую, но эффективную фильтрацию данных.  
Для сортировки используется универсальная функция getSorted, принимающая лямбда-функцию, реализующую логику сравнения. В данной реализации применяется функция sortAscending, сортирующая строки в алфавитном порядке. Также реализован метод display, выводящий отсортированные ссылки либо с нумерацией, либо без.  
В функции main создается объект класса BD, загружающий данные из файла laba4.2.txt. Далее демонстрируется вывод библиографических ссылок сначала с нумерацией, затем без неё. Благодаря использованию функционального программирования через std::function и лямбда-выражения, программа легко масштабируется и может быть адаптирована для других задач обработки строк.  
Таким образом, выполненная работа наглядно демонстрирует применение современных возможностей языка C++ в решении практических задач по обработке текстовой информации. Программа написана модульно, гибко и обладает хорошей читаемостью и расширяемостью.