**Лабораторная работа №3**

**Построение программного интерфейса для работы с базой данных**

**Задание**

1. Изучить принципы физической организации баз данных
2. Изучить организацию схемы и структуры базы данных, структуры таблиц, типов данных в таблицах
3. Изучить основные типы запросов в базах данных – на создание БД, на создание таблиц, на выборку, на добавление, на удаление и на модификацию данных
4. Изучить интегрированные среды разработки ПО и программные средства управления базами данных
5. Разработать схему базу данных учета кинофильмов в виде UML-диаграммы: таблица *ФИЛЬМЫ(поля: название, страна, режиссер, год\_выхода, кинотеатр);* *КИНОТЕАТР(поля: название, фильм, сеанс, места, билеты, рейтинг(от 0,0 до 100,0 баллов); ОТЗЫВЫ(поля: кинотеатр, фильм, отзыв, балл (от 0,0 до 100,0 баллов)); ХАРАКТЕРИСТИКИ(поля: фильм, описание, жанр, ограничения, главный\_герой, продолжительность, 3D (да/нет), мировой\_рейтинг).* Дополнительно предусмотреть в таблицах индексные и ключевые поля для связи таблиц и нормализации базы данных
6. В локальной СУБД (например, – MS Access) создать базу данных на основе схемы БД (модели - см. п. 5)
7. Разработать программу для интерактивной работы с базой данных на основе запросов основных типов
8. Создать главную форму программы. На форме расположить следующие элементы управления – главное меню программы (*MainMenu*); текстовое поле (*Edit*) для ввода условий запросов на выборку (часть запроса - WHERE); ниже – выпадающий список (*ComboBox*) с названиями таблиц базы данных; ниже - компонент сетки, связанной с данными (*DataGrid*) для отображения результирующего набора записей; ниже - компоненты текстовых полей (*Edit*) для ввода новых значений полей; ниже – две кнопки (*Button*) для перехода к первой или к последней записи результирующего набора записей; справа от компонента (*DataGrid*) - кнопки (*Button*) для ввода команд. Каждая кнопка реализует один тип запроса: 1. Выборка, 2. Удаление, 3. Добавление, 4. Модификация. На кнопках должны быть соответствующие надписи
9. В свойствах компонента (*DataGrid*) предусмотреть возможность выбора строк набора записей
10. Один текстовый компонент (*Edit*) для ввода новых значений полей должен располагаться под одним полем таблицы (*DataGrid*). Количество текстовых компонентов должно динамически меняться и всегда соответствовать количеству полей результирующего набора записей в компоненте (*DataGrid*)
11. Реализовать следующие функции программы:
    1. Реализовать функции главного меню: «Подключиться» (программа динамически подключается к базе данных), «Сохранить» (набор записей, отображаемый на сетке данных, сохраняется в файл), «Выйти» (закрывается программа, предварительно происходит отключение от базы данных)
    2. Из выпадающего списка последовательно выбирать названия таблиц и отображать их содержимое в сетке данных (*DataGrid*)
    3. Для каждой выбранной таблицы реализовывать запросы с помощью кнопок (см. п. 7). При нажатии на кнопку формировать текст SQL-запроса с учетом содержимого полей с условием и полей с новыми значениями
       1. При запросе на выборку задавать условие выборки в текстовом поле (Edit) в верхней части формы
       2. При запросе на добавление и модификацию данных записывать новые значения полей в текстовых полях (Edit) под сеткой данных
       3. При запросе на удаление должна удаляться выделенная запись в сетке данных (*DataGrid*)
       4. Реализовать все типы запросов при трех различных условиях в части WHERE
       5. При реализации запросов результаты сразу же должны отображаться в сетке данных
12. Все запросы формировать динамически программным путем
13. Сделать выводы о компонентах, связанных с данными, о формировании запросов, о навигации в БД

**Отчет должен содержать:**

1. Основные сведения о физической организации баз данных
2. Основные сведения об основных типах запросов
3. Схему базы данных в виде UML-диаграммы
4. Блок-схемы алгоритмов работы методов классов
5. Скриншот экрана приложения с результатом работы программы
6. Текст исходных кодов программы
7. Краткие выводы
8. Перечень использованных источников