

Лабораторная работа №20

Загрузка и сохранение файлов в БД

1 Цель работы

1.1 Научиться реализовывать настройку интерфейса пользователя для загрузки и сохранения файлов в БД через приложение.

2 Литература

2.1 Полное руководство по языку программирования C# 14 и платформе .NET 10. metanit.com – Текст : электронный // metanit.com, 2025. – URL: <https://metanit.com/sharp/tutorial> - гл.22

3 Подготовка к работе

3.1 Повторить теоретический материал (см.п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Подготовка БД и структуры проекта

5.1.1 Добавить в БД таблицу Кадры (идентификатор, ид фильма, название файла (200 символов)), связанную с таблицей Фильмы.

5.1.2 Для работы создать новое решение, в которое добавить:

- библиотеку классов (для контекста, моделей данных таблиц Фильмы и Кадры),
- оконное приложение WPF,
- веб-приложение Razor Pages.

5.2 Загрузка файла на веб-сервер

5.2.1 В веб-приложении сгенерировать страницы CRUD для созданных моделей.

5.2.2 Добавить в веб-приложение в wwwroot папку images.

5.2.3 На странице добавления кадра реализовать отображение в выпадающем списке названия фильма.

Для того, чтобы через форму можно было загружать файлы на сервер, добавить в нее атрибут enctype:

```
<form method="post" enctype="multipart/form-data">
```

Для отображения окна выбора изображения добавить следующий input:

```
<div class="form-group">
  <input type="file" name="photo" accept="image/*" />
</div>
```

5.2.4 Для сохранения файла добавить в метод Post код для копирования данных файла в поток.

Пример:

```
var file = HttpContext.Request.Form.Files.FirstOrDefault();
if (file?.Length > 0)
{
    var path = Path.Combine(Directory.GetCurrentDirectory(),
        "wwwroot", "images", file.FileName);
    using var stream = new FileStream(path, FileMode.Create);
    await file.CopyToAsync(stream);
    Product.Photo = file.FileName;
}
```

5.2.5 Добавить проверку на размер загружаемого файла (не более 2 МБ).

5.3 Загрузка файла в папку оконного приложения

5.3.1 В оконном приложении настроить окно для выбора названия фильма из выпадающего списка и добавить на окно кнопку «Добавить кадр».

5.3.2 При нажатии на кнопку «Добавить кадр» должно открываться окно выбора файла (с фильтром на изображения).

5.3.3 Реализовать копирование выбранного файла в папку images (рядом с исполняемым файлом). Если ее не существует, создавать автоматически.

5.3.4 Добавить проверку на размер загружаемого файла (не более 2 МБ).

5.4 Загрузка файла в БД через веб-приложение

5.4.1 Для сохранения файла в БД добавить на страницу редактирования фильма настройки для загрузки файла через форму.

5.4.2 Добавить в метод Post код для копирования данных файла в поток.

Пример:

```
if (file?.Length > 0)
{
    using var memoryStream = new MemoryStream();
    await file.CopyToAsync(memoryStream);
    Product.Photo = memoryStream.ToArray();
}
```

5.4.3 Добавить проверку на размер загружаемого файла (не более 3 МБ).

5.5 Загрузка файла в БД через оконное приложение

5.5.1 Для сохранения файла в БД добавить в окно кнопку «Выбрать постер».

5.5.2 При нажатии на кнопку «Выбрать постер» должно открываться окно выбора файла (с фильтром на изображения).

5.5.3 Реализовать копирование выбранного файла в поле таблицы БД.

Пример:

```
string filePath = "image.jpg"; // Заменить на путь к файлу
byte[] fileBytes = File.ReadAllBytes(filePath);
```

5.5.4 Добавить проверку на размер загружаемого файла (не более 3 МБ).

6 Порядок выполнения работы

6.1 Выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8Контрольные вопросы

- 8.1 Чем отличается хранение `byte[]` от `string`?
- 8.2 Как получить `IFormFile` в веб-приложении?
- 8.3 Как передать файл в оконном приложении?
- 8.4 Что такое `MemoryStream` и зачем он используется?
- 8.5 Как сохранить файл в папку, а путь к нему в БД?