Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

«Программирование интернет-серверов»

Отчёт по лабораторной работе №3

ASPA.

Выполнил: Сосновец М.И.

ФИТ 3 курс 4 группа

Преподаватель: Гончар Е.А.

Минск 2025

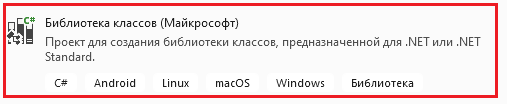
**Задание 1.**

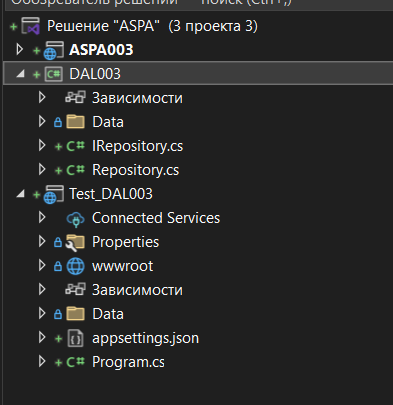
1. Разработайте библиотеку, применив следующий шаблон

Имя решения: **ASPA**

Имя проекта: **DAL003**

Версия .NET: **7.0** или **8.0**



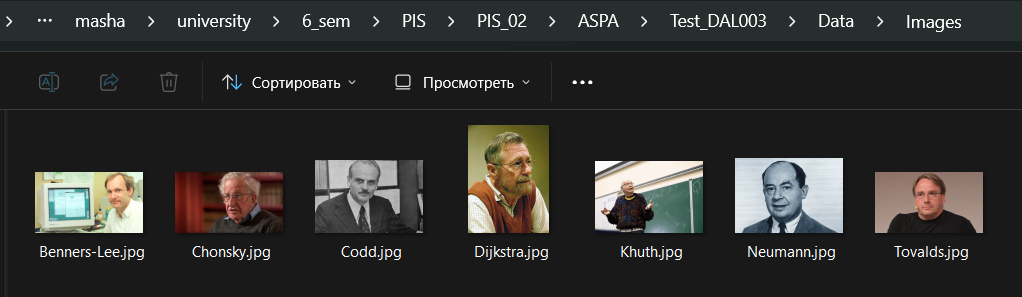


1. Библиотека **DAL003** должна реализовать DAL (Data Access Level) реализовав паттерн **Repository** для доступа к следующим данным в формате JSON.

|  |
| --- |
| using DAL003;  using System.Text.Json;  public class Repository : IRepository  {  private readonly string \_jsonFileName;  private readonly string \_directory;  private List<Celebrity> \_celebrities;  public string BasePath => \_directory;  public Repository(string directory, string jsonFileName)  {  \_directory = directory;  \_jsonFileName = jsonFileName;  \_celebrities = new List<Celebrity>();  LoadData();  }  private void LoadData()  {  var jsonFilePath = Path.Combine(\_directory, \_jsonFileName);  if (File.Exists(jsonFilePath))  {  try  {  var json = File.ReadAllText(jsonFilePath);  \_celebrities = JsonSerializer.Deserialize<List<Celebrity>>(json) ?? new List<Celebrity>();  }  catch (Exception ex)  {  Console.WriteLine($"Error loading data from JSON: {ex.Message}");  \_celebrities = new List<Celebrity>();  }  }  else  {  \_celebrities = new List<Celebrity>();  }  }  public Celebrity[] GetAllCelebrities() => \_celebrities.ToArray();  public Celebrity? getCelebrityById(int id) => \_celebrities.FirstOrDefault(c => c.Id == id);  public Celebrity[] GetCelebritiesBySurname(string surname) =>  \_celebrities.Where(c => c.Surname.Equals(surname, StringComparison.OrdinalIgnoreCase)).ToArray();  public string? getPhotoPathById(int id)  {  var celebrity = getCelebrityById(id);  return celebrity?.PhotoPath;  }  public void Dispose()  {  }  public static Repository Create(string directory, string jsonFileName)  {  return new Repository(directory, jsonFileName);  }  } |



1. JSON-файл ( статическое свойство ***Repository.JSONFileName***) и фотографии, находятся в директории, имя которого задается параметром конструктора ***Repository***. Сам директорий с файлами фотографий и JSON-файлом располагается в текущем директории приложения.
2. Структура директории для хранения фотографий и json-файла приведена ниже



1. **Repository** должен реализовывать следующий интерфейс

|  |
| --- |
| using System;  using System.Collections.Generic;  namespace DAL003  {  public interface IRepository : IDisposable  {  string BasePath { get; }  Celebrity[] GetAllCelebrities();  Celebrity? getCelebrityById(int id);  Celebrity[] GetCelebritiesBySurname(string surname);  string? getPhotoPathById(int id);  }  public record Celebrity(int Id, string Firstname, string Surname, string PhotoPath);  } |

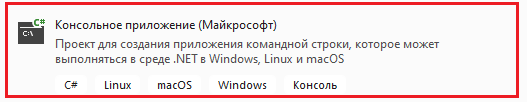
♣**Задание 2.**

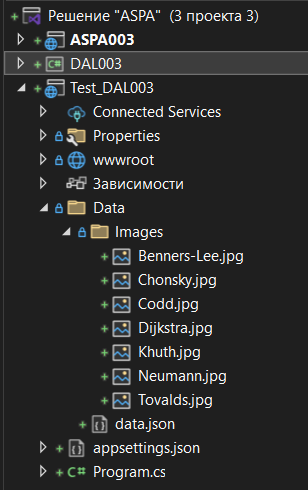
1. Разработайте консольное приложение, применив следующий шаблон

Имя решения: **ASPA**

Имя проекта: **Test**\_**DAL003**

Версия .NET: **7.0** или **8.0**

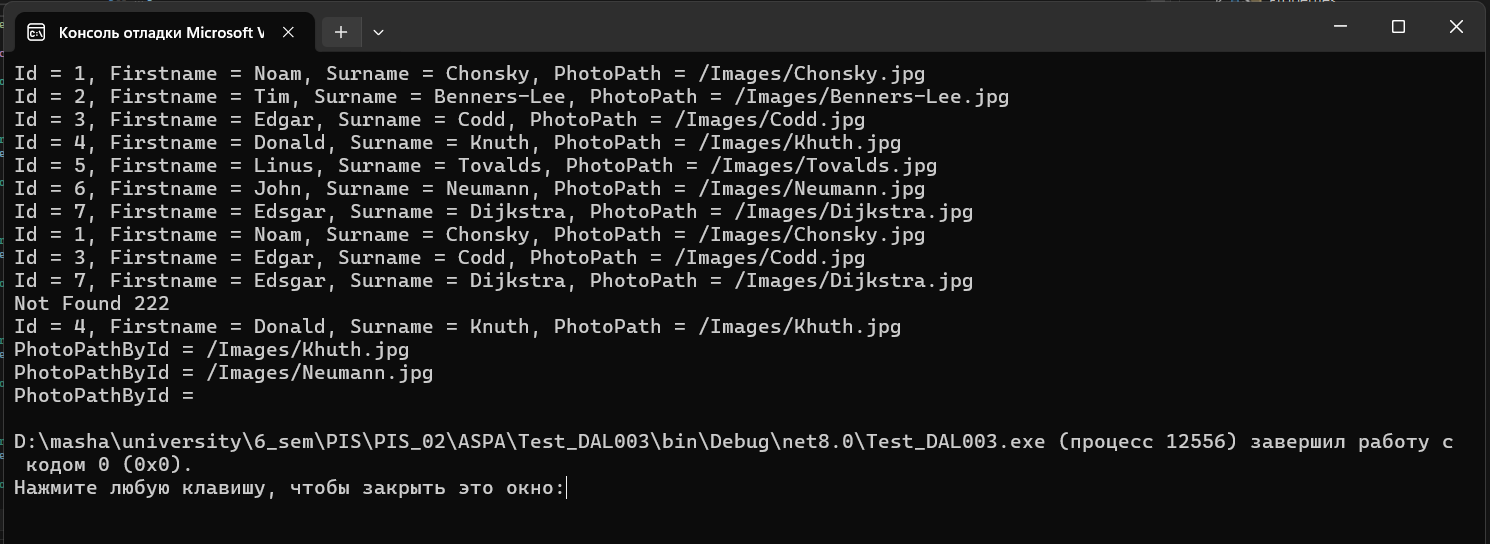
****

****

1. Приложение **Test**\_**DAL003** предназначено для тестирования **DAL003.**
2. Ниже приведён код и результат теста.

|  |
| --- |
| using Microsoft.AspNetCore.Builder;  using Microsoft.Extensions.DependencyInjection;  using Microsoft.Extensions.Hosting;  using DAL003;  using System.Text;  using Microsoft.Extensions.FileProviders;  namespace Test\_DAL003  {  public class Program  {  private static void Main(string[] args)  {  using (IRepository repository = Repository.Create("Data", "data.json"))  {  foreach (Celebrity celebrity in repository.GetAllCelebrities())  {  Console.WriteLine($"Id = {celebrity.Id}, Firstname = {celebrity.Firstname}, " +  $"Surname = {celebrity.Surname}, PhotoPath = {celebrity.PhotoPath}");  }  Celebrity celebrity1 = repository.getCelebrityById(1);  if (celebrity1 != null)  {  Console.WriteLine($"Id = {celebrity1.Id}, Firstname = {celebrity1.Firstname}, " +  $"Surname = {celebrity1.Surname}, PhotoPath = {celebrity1.PhotoPath}");  }  Celebrity celebrity3 = repository.getCelebrityById(3);  if (celebrity3 != null)  {  Console.WriteLine($"Id = {celebrity3.Id}, Firstname = {celebrity3.Firstname}, " +  $"Surname = {celebrity3.Surname}, PhotoPath = {celebrity3.PhotoPath}");  }  Celebrity celebrity7 = repository.getCelebrityById(7);  if (celebrity7 != null)  {  Console.WriteLine($"Id = {celebrity7.Id}, Firstname = {celebrity7.Firstname}, " +  $"Surname = {celebrity7.Surname}, PhotoPath = {celebrity7.PhotoPath}");  }  Celebrity celebrity222 = repository.getCelebrityById(222);  if (celebrity222 != null)  {  Console.WriteLine($"Id = {celebrity222.Id}, Firstname = {celebrity222.Firstname}, " +  $"Surname = {celebrity222.Surname}, PhotoPath = {celebrity222.PhotoPath}");  }  else  {  Console.WriteLine("Not Found 222");  }  foreach (Celebrity celebrity in repository.GetCelebritiesBySurname("Chomsky"))  {  Console.WriteLine($"Id = {celebrity.Id}, Firstname = {celebrity.Firstname}, " +  $"Surname = {celebrity.Surname}, PhotoPath = {celebrity.PhotoPath}");  }  foreach (Celebrity celebrity in repository.GetCelebritiesBySurname("Knuth"))  {  Console.WriteLine($"Id = {celebrity.Id}, Firstname = {celebrity.Firstname}, " +  $"Surname = {celebrity.Surname}, PhotoPath = {celebrity.PhotoPath}");  }  foreach (Celebrity celebrity in repository.GetCelebritiesBySurname("XXXX"))  {  Console.WriteLine($"Id = {celebrity.Id}, Firstname = {celebrity.Firstname}, " +  $"Surname = {celebrity.Surname}, PhotoPath = {celebrity.PhotoPath}");  }  Console.WriteLine($"PhotoPathById = {repository.getPhotoPathById(4)}");  Console.WriteLine($"PhotoPathById = {repository.getPhotoPathById(6)}");  Console.WriteLine($"PhotoPathById = {repository.getPhotoPathById(222)}");  }  }  }  } |

1. Выполните тест **Test**\_**DAL003** и убедитесь в его успешном выполнении.



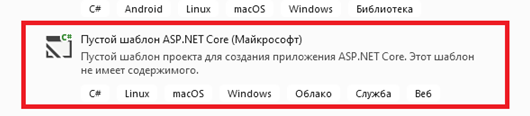
**Задание 3.**

1. Разработайте ASPA, применив следующий шаблон.

Имя решения: **ASPA**

Имя проекта: **ASPA003**

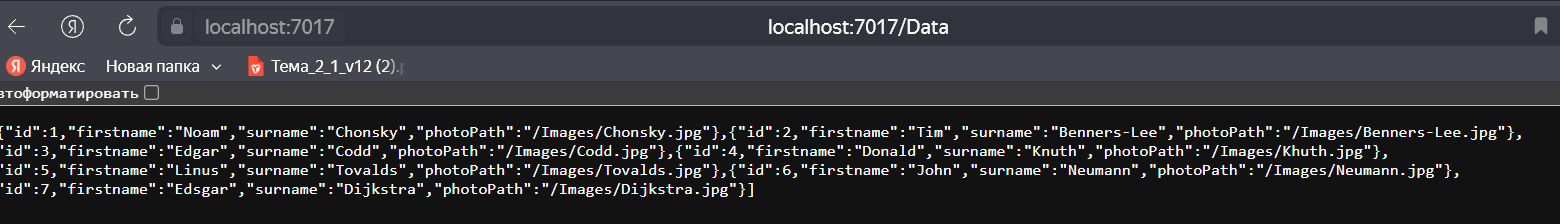
Версия .NET: **7.0** или **8.0**

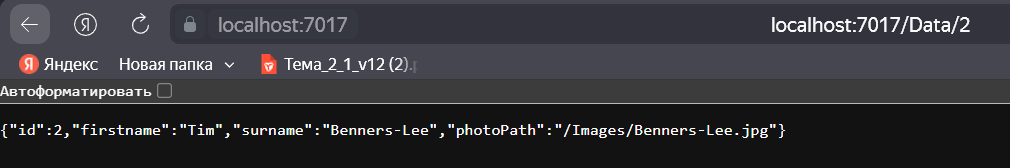


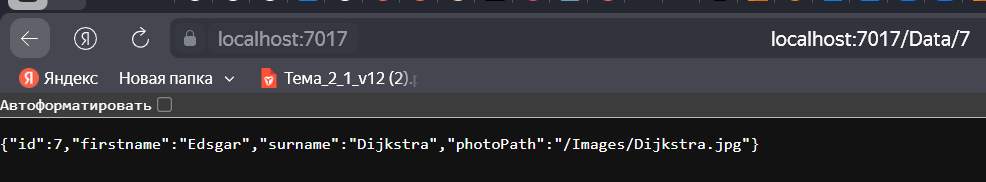
1. Подключите библиотеку **DAL003** к приложению **ASPA003**
2. Фрагмент кода **ASPA003** приведён ниже, исследуйте его и включите в своё приложение.

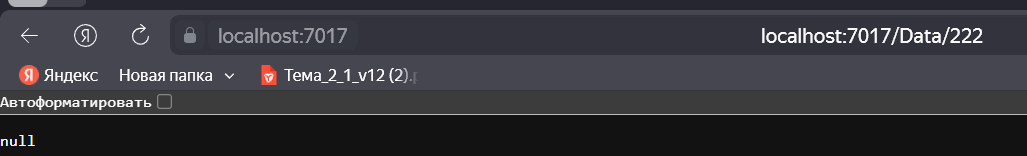
|  |
| --- |
| using DAL003;  using Microsoft.AspNetCore.Builder;  using Microsoft.Extensions.DependencyInjection;  var builder = WebApplication.CreateBuilder(args);  var app = builder.Build();  using (IRepository repository = Repository.Create("Data", "Data.json"))  {  app.MapGet("/Data", () => repository.GetAllCelebrities());  app.MapGet("/Data/{id:int}", (int id) => repository.getCelebrityById(id));  app.MapGet("/Data/BySurname/{surname}", (string surname) => repository.GetCelebritiesBySurname(surname));  app.MapGet("/Data/PhotoPathById/{id:int}", (int id) => repository.getPhotoPathById(id));  }  app.Run(); |

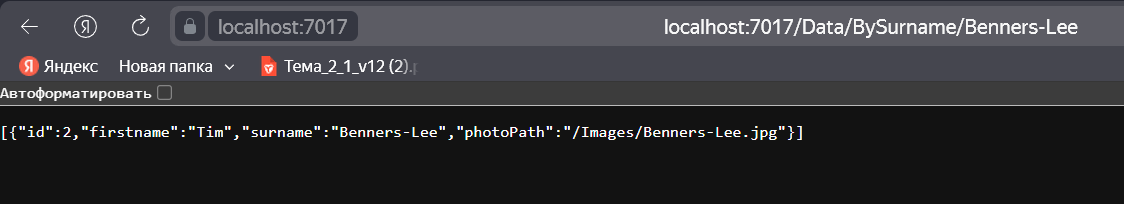
1. Проверьте следующие результаты работы.

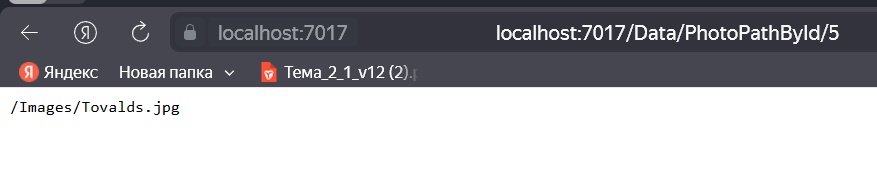










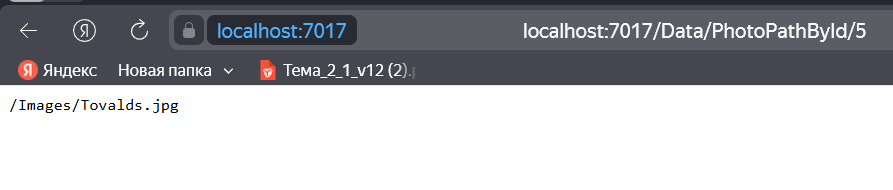


♣**Задание 4.**

1. Внесите изменения в настройки **ASPA003 (UseStaticFiles, StaticFileOptions)**,таким образом, чтобы файлы фотографий были доступны как статические ресурсы.

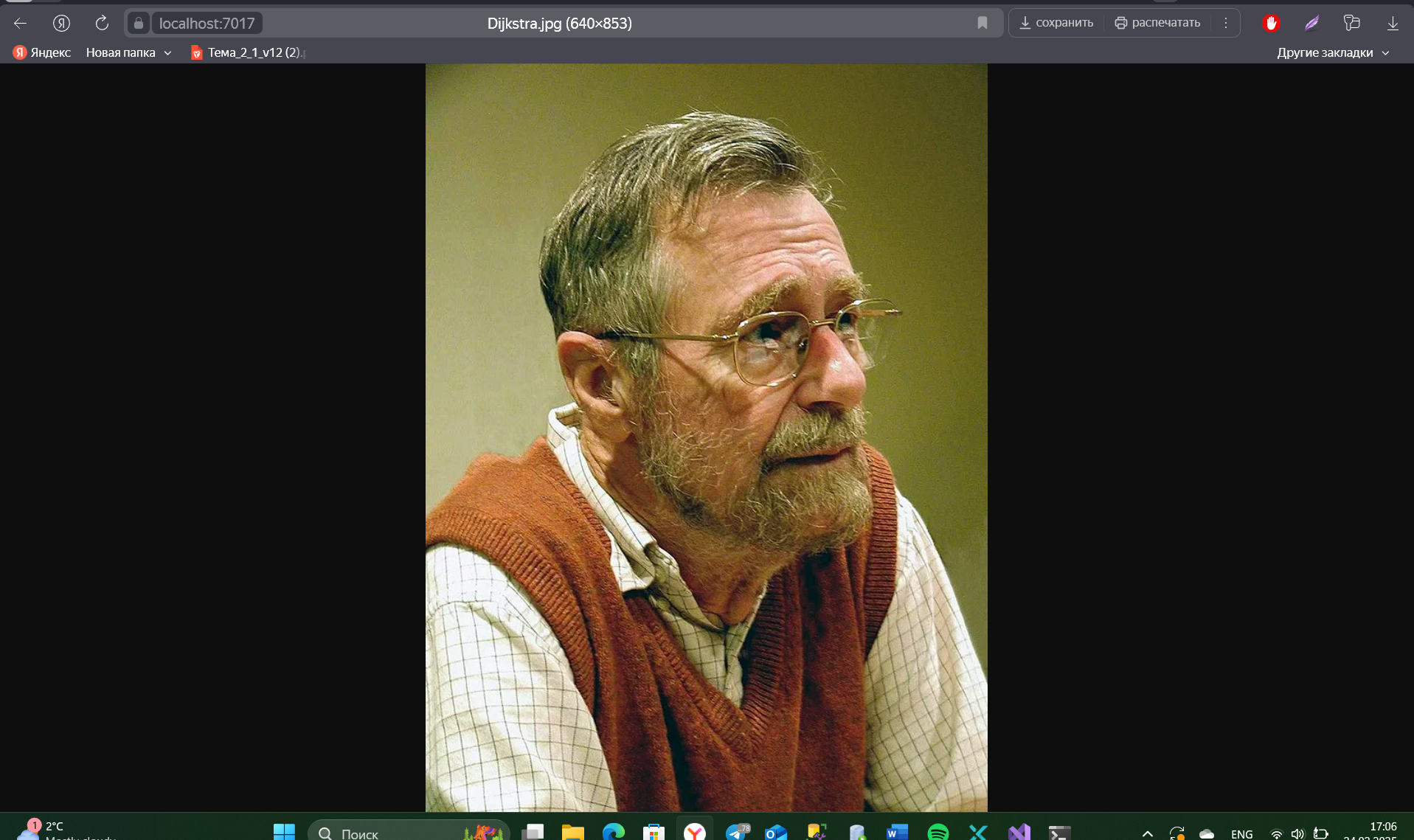
|  |
| --- |
| using DAL003;  using Microsoft.AspNetCore.Builder;  using Microsoft.Extensions.DependencyInjection;  using Microsoft.Extensions.FileProviders;  using System.IO;  var builder = WebApplication.CreateBuilder(args);  var app = builder.Build();  app.UseStaticFiles(new StaticFileOptions  {  FileProvider = new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(), "Data", "Images")),  RequestPath = "/photos"  });  using (IRepository repository = Repository.Create("Data", "Data.json"))  {  app.MapGet("/Data", () => repository.GetAllCelebrities());  app.MapGet("/Data/{id:int}", (int id) => repository.getCelebrityById(id));  app.MapGet("/Data/BySurname/{surname}", (string surname) => repository.GetCelebritiesBySurname(surname));  app.MapGet("/Data/PhotoPathById/{id:int}", (int id) => repository.getPhotoPathById(id));  }  app.Run(); |

1. Проверьте следующие результаты работы







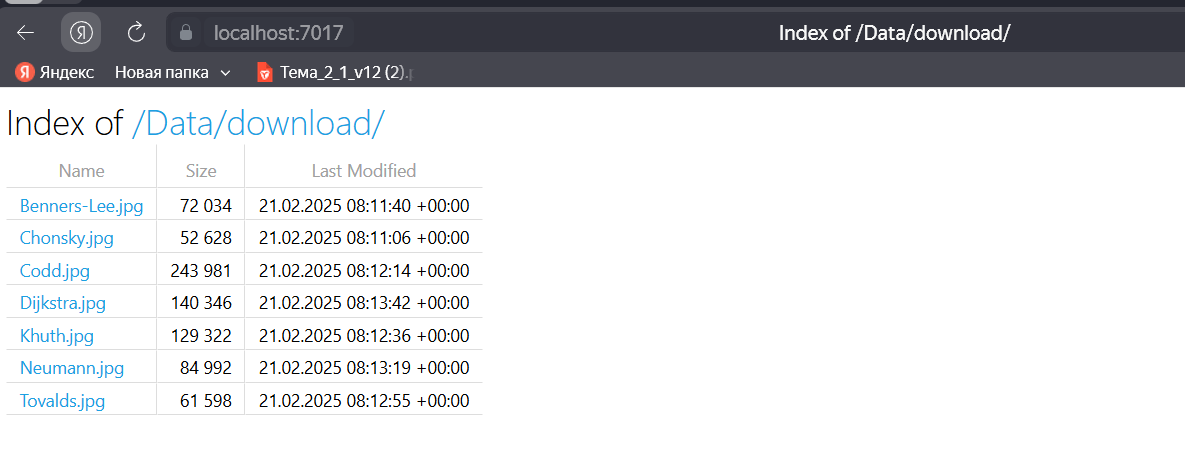


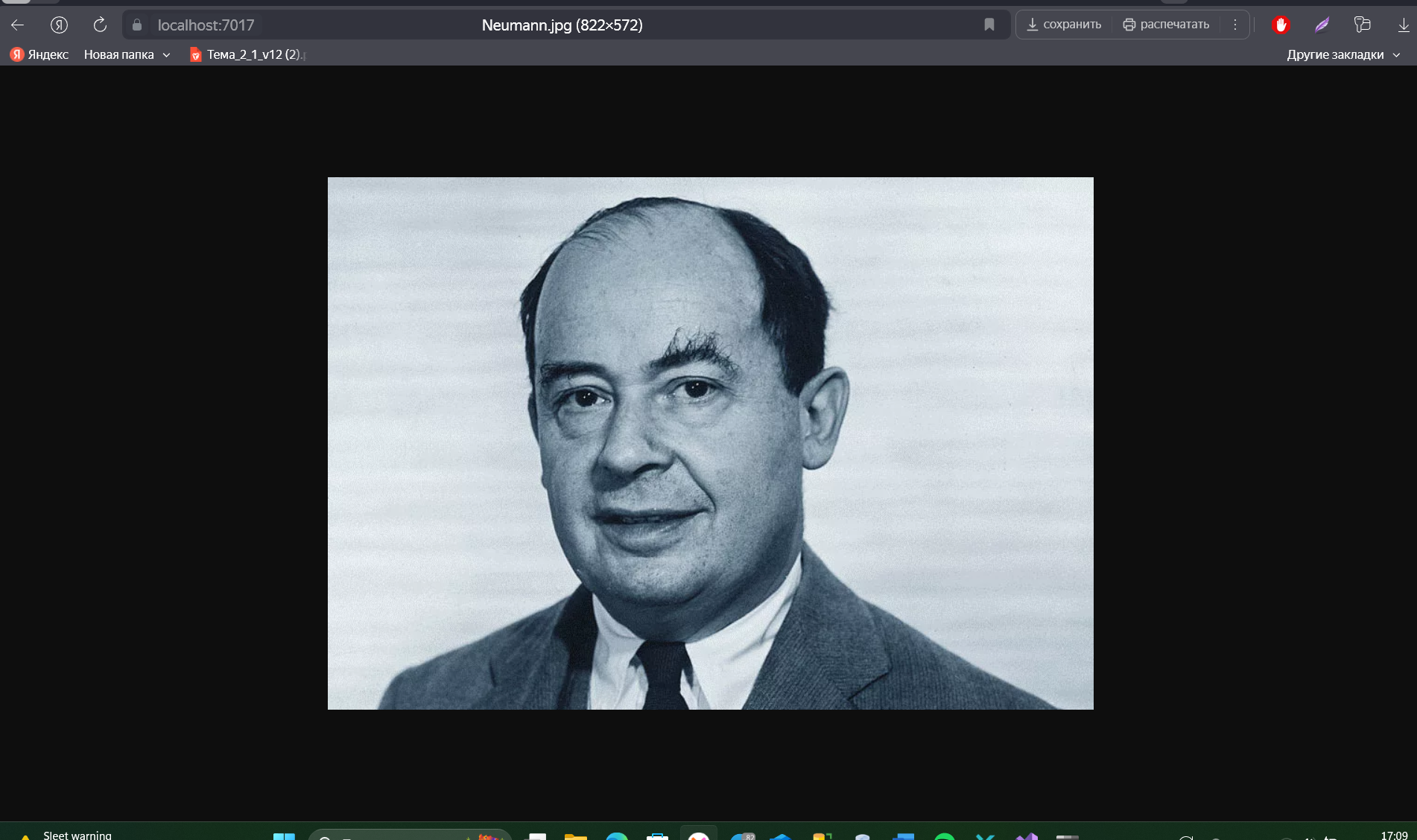
♣**Задание 5.**

1. Внесите изменения в настройки **ASPA003 (UseDirectoryBrowser, DirectoryBrowserOptions)**,таким образом, чтобы файлы фотографий были доступны для отображения.

|  |
| --- |
| using DAL003;  using Microsoft.AspNetCore.Builder;  using Microsoft.Extensions.DependencyInjection;  using Microsoft.Extensions.FileProviders;  using System.IO;  var builder = WebApplication.CreateBuilder(args);  var app = builder.Build();  app.UseStaticFiles(new StaticFileOptions  {  FileProvider = new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(), "Data", "Images")),  RequestPath = "/photos"  });  app.UseStaticFiles(new StaticFileOptions  {  FileProvider = new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(), "Data", "download")),  RequestPath = "/Data/download"  });  app.UseDirectoryBrowser(new DirectoryBrowserOptions  {  FileProvider = new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(), "Data", "download")),  RequestPath = "/Data/download"  });  app.MapGet("/", () => "Hello");  app.Run(); |

1. Проверьте следующие результаты работы





1. Внесите изменения в настройки **ASPA003 (UseStaticFiles, StaticFileOptions, OnPrepareResponse)**,таким образом, чтобы файлы фотографий были доступны скачивания.

|  |
| --- |
| using DAL003;  using Microsoft.AspNetCore.Builder;  using Microsoft.Extensions.DependencyInjection;  using Microsoft.Extensions.FileProviders;  using Microsoft.AspNetCore.Http;  using System.IO;  var builder = WebApplication.CreateBuilder(args);  var app = builder.Build();  app.UseStaticFiles(new StaticFileOptions  {  FileProvider = new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(), "Data", "Images")),  RequestPath = "/photos",  OnPrepareResponse = context =>  {  context.Context.Response.Headers["Content-Disposition"] = "attachment; filename=" + Path.GetFileName(context.File.Name);  }  });  app.UseStaticFiles(new StaticFileOptions  {  FileProvider = new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(), "Data", "download")),  RequestPath = "/Data/download",  OnPrepareResponse = context =>  {  context.Context.Response.Headers["Content-Disposition"] = "attachment; filename=" + Path.GetFileName(context.File.Name);  }  });  app.UseDirectoryBrowser(new DirectoryBrowserOptions  {  FileProvider = new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(), "Data", "download")),  RequestPath = "/Data/download"  });  app.MapGet("/", () => "Hello");  app.Run(); |

1. Проверьте следующие результаты работы

