ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ПИ им. Фельдмана

Факультет ИСП

Отчет по лабораторной работе №2

Тема: «Работа с ветками»

Курс: «Профессиональная практика программной инженерии»

Выполнил:

ст. гр. ПИ-18а

Жильцов В.А.

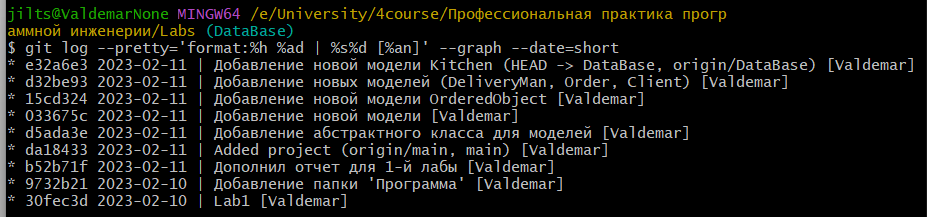
Проверили:

Филипишин Д. А.

Донецк – 2023

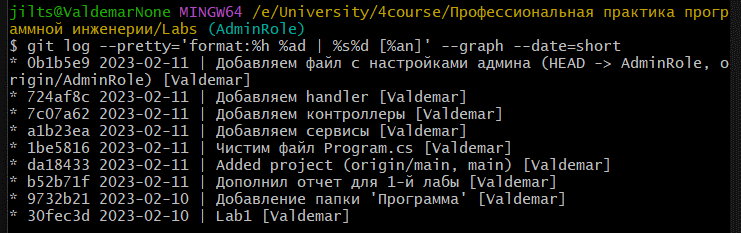
Модуль моделей ORM — ветка db\_models

Результат вывода git log --pretty='format:%h %ad | %s%d [%an]' --graph --date=short



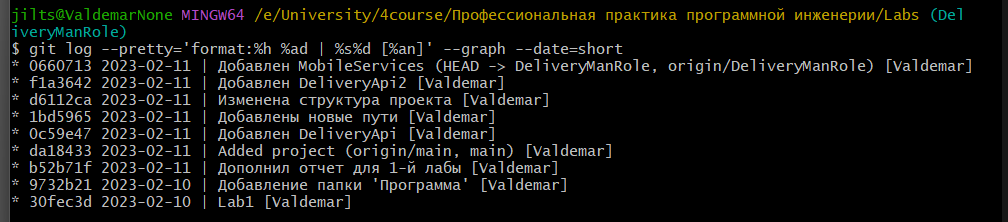
Модуль администратора — AdminRole

Результат вывода git log --pretty='format:%h %ad | %s%d [%an]' --graph --date=short



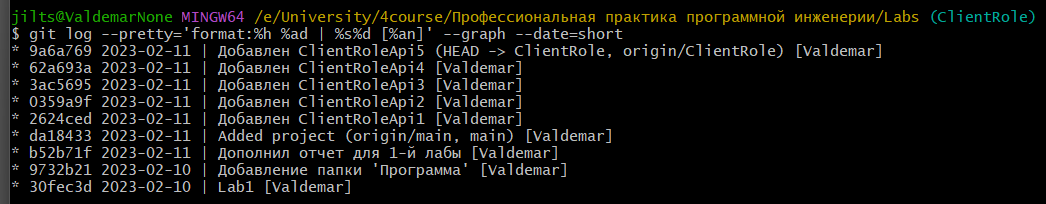
Модуль курьера — DeliveryManRole

Результат вывода git log --pretty='format:%h %ad | %s%d [%an]' --graph --date=short



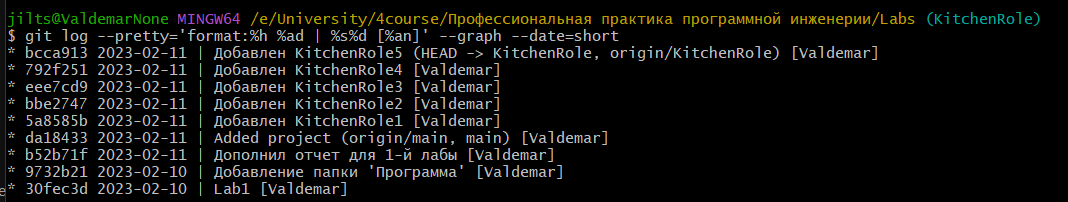
Модуль клиента — ClientRole

Результат вывода git log --pretty='format:%h %ad | %s%d [%an]' --graph --date=short

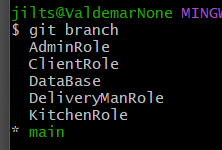


Модуль кухни — KitchenRole

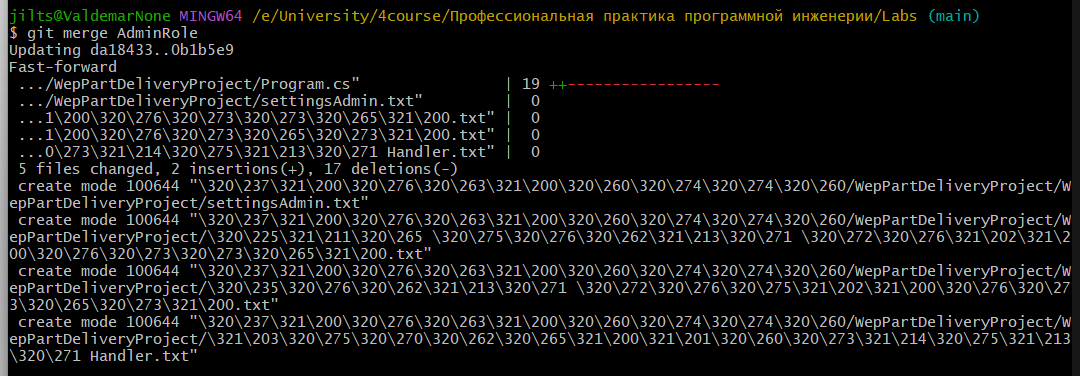
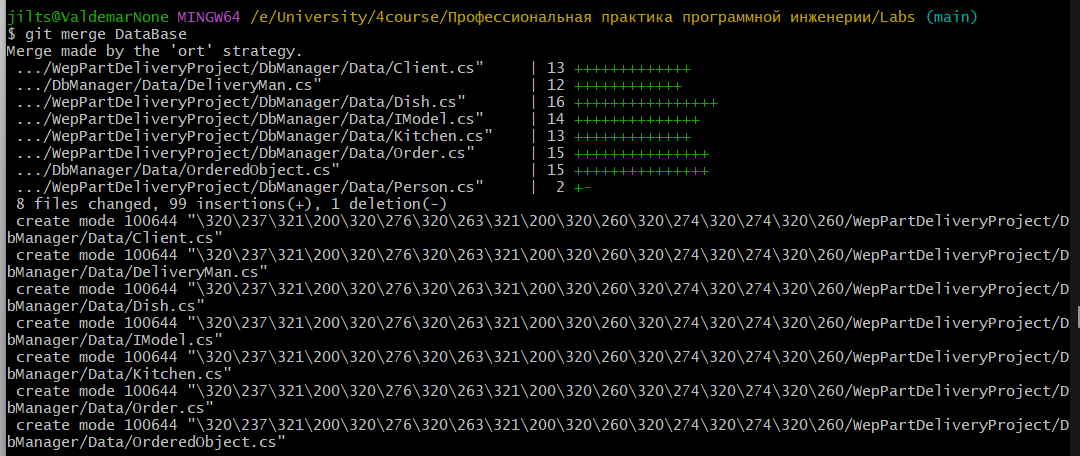
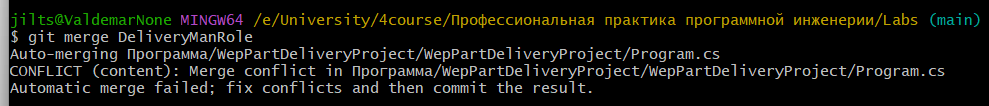
Результат вывода git log --pretty='format:%h %ad | %s%d [%an]' --graph --date=short



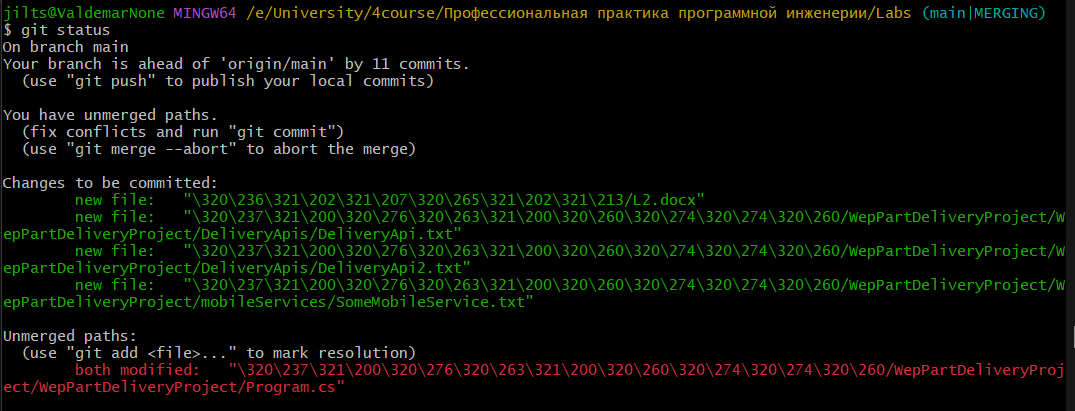
Список веток:



Сливаем ветки:

При слиянии ветки DeliveryManRole получили конфликт в файле Program.cs.



Код из файла.

using DbManager;

using DbManager.Neo4j.Implementations;

using DbManager.Neo4j.Interfaces;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.Options;

using Neo4j.Driver;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

var services = builder.Services;

var Configuration = builder.Configuration;

// Register application setting using IOption provider mechanism

services.Configure<ApplicationSettings>(Configuration.GetSection("ApplicationSettings"));

services.AddTransient<SomeClass>(services.BuildServiceProvider());

services.AddTransient<SomeOtherClass>(services.BuildServiceProvider());

// Fetch settings object from configuration

var settings = new ApplicationSettings();

Configuration.GetSection("ApplicationSettings").Bind(settings);

// This is to register your Neo4j Driver Object as a singleton

services.AddSingleton(GraphDatabase.Driver(settings.Neo4jConnection, AuthTokens.Basic(settings.Neo4jUser, settings.Neo4jPassword)));

// This is your Data Access Wrapper over Neo4j session, that is a helper class for executing parameterized Neo4j Cypher queries in Transactions

services.AddScoped<INeo4jDataAccess, Neo4jDataAccess>();

// This is the registration for your domain repository class

//services.AddTransient<IPersonRepository, PersonRepository>();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

<<<<<<< HEAD

=======

app.Map("/time", appBuilder =>

{

var time = DateTime.Now.ToShortTimeString();

// логгируем данные - выводим на консоль приложения

appBuilder.Use(async (context, next) =>

{

Console.WriteLine($"Time: {time}");

await next(); // вызываем следующий middleware

});

appBuilder.Run(async context => await context.Response.WriteAsync($"Time: {time}"));

});

app.Map("/delivary", appBuilder =>

{

var time = DateTime.Now.ToShortTimeString();

// логгируем данные - выводим на консоль приложения

appBuilder.Use(async (context, next) =>

{

Console.WriteLine($"Time: {time}");

await next(); // вызываем следующий middleware

});

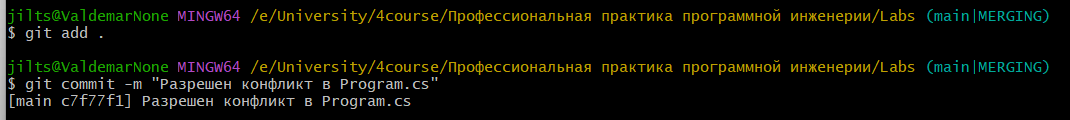
appBuilder.Run(async context => await context.Response.WriteAsync($"Time: {time}"));

});

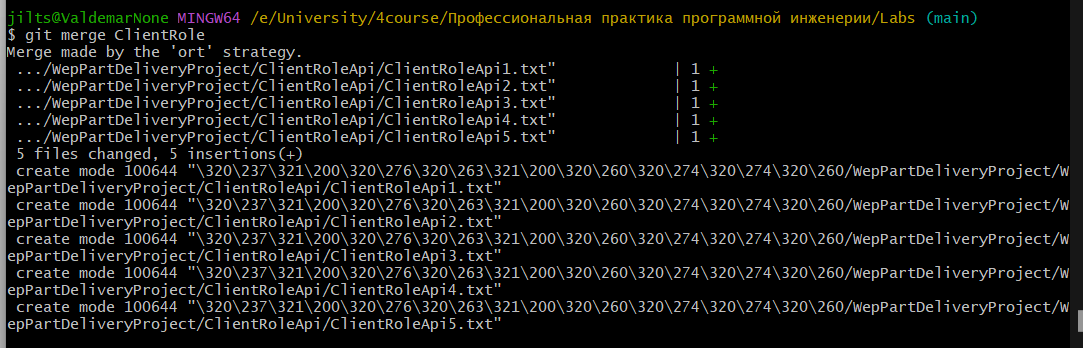
>>>>>>> DeliveryManRole

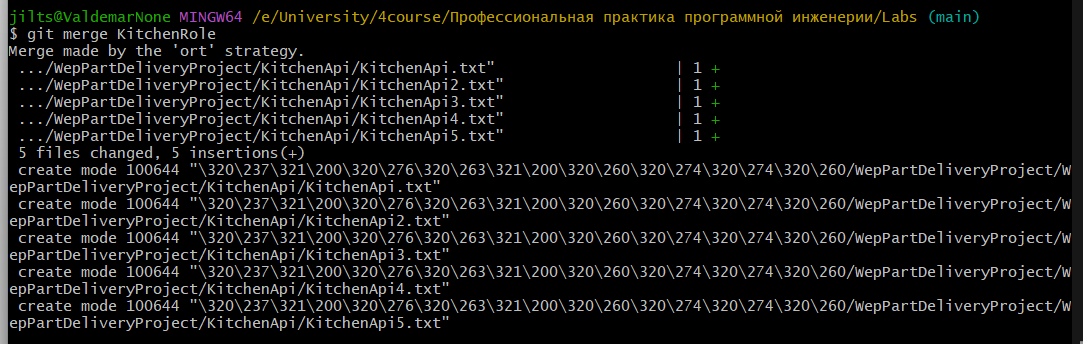
app.Run();

Просто удаляем разделители конфликта в файле для его разрешения.

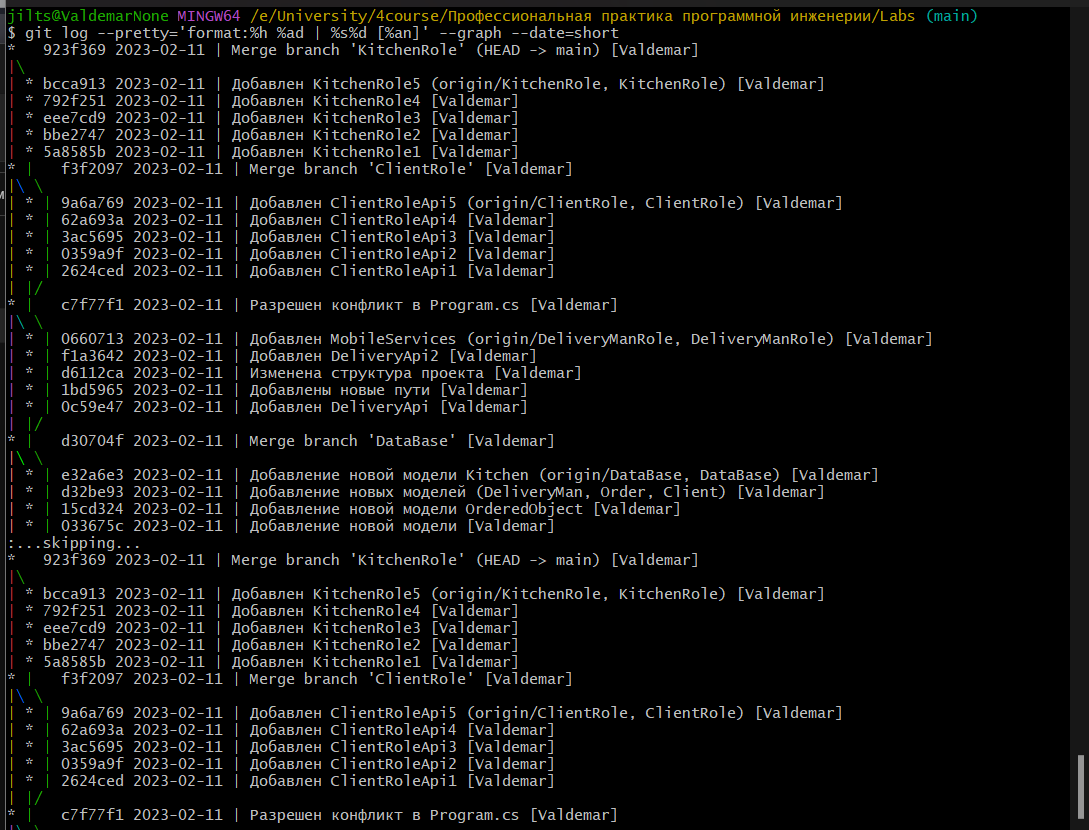


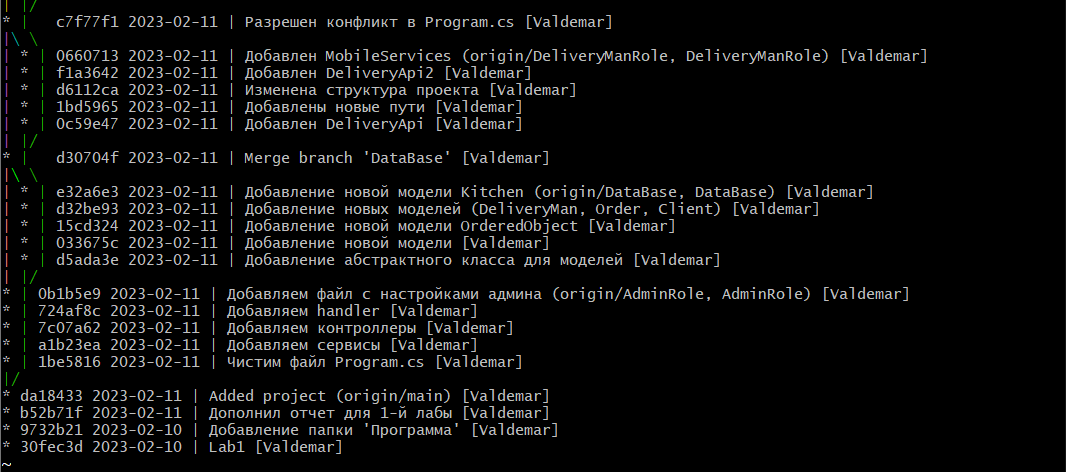
Сливаем оставшиеся ветки:





Результат





Удаление веток:

