

Московский Международный Колледж Цифровых Технологий  
«Академия ТОП»

**ПРОЕКТ**

**по предмету Технология доступа к базам данных ADO.NET:  
«Animal Shelter Management System»**

Выполнили студенты группы 9/3-РПО-23/1

Еремеев Иван Иванович

Панов Илья Сергеевич

Санкт-Петербург, 2024

# ВВЕДЕНИЕ

Animal Shelter Management System — это консольное приложение, предназначенное для автоматизации и повышения эффективности работы приюта для животных.

Цель проекта – спроектировать БД для приюта животных и создать CLI для управления БД.

Задачи проекта:

- Спроектировать и реализовать нормализованную схему базы данных
- Разработать слой доступа к данным
- Создать удобный консольный пользовательский интерфейс
- Обеспечить функционал для отчетности и аналитики
- Подготовить документацию и инструкции по развертыванию

# ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ

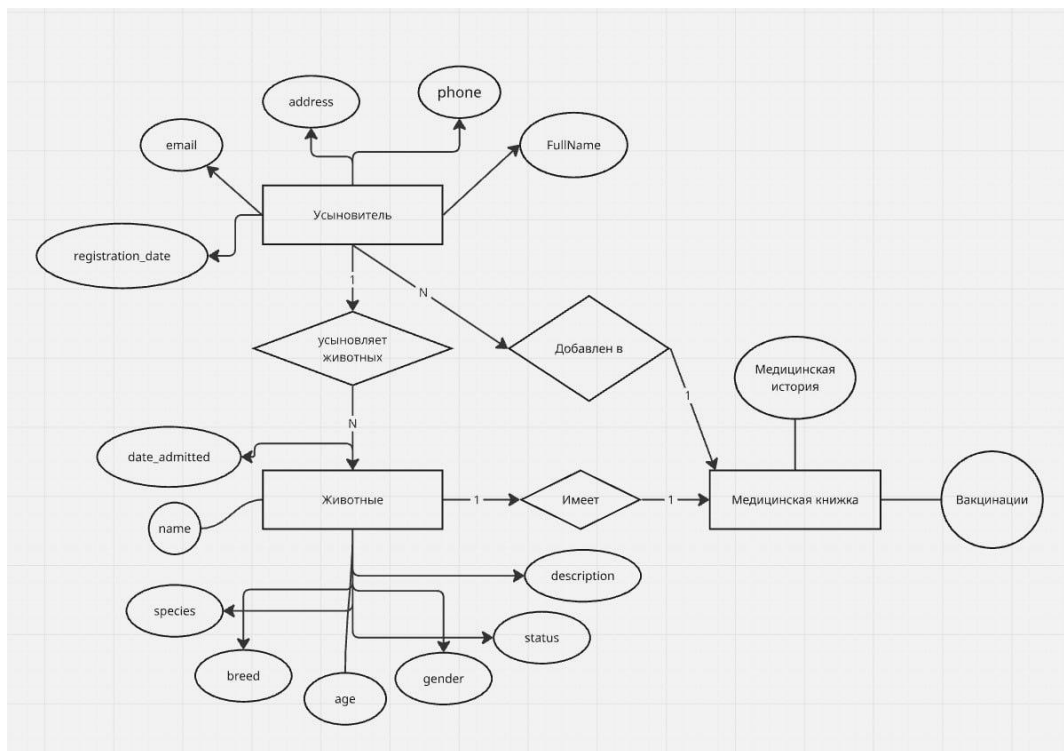


Рисунок 1. Первая ER-Диаграмма

Данная ER-диаграмма отображает структуру информационной системы для учета приюта или центра усыновления животных. В системе выделены три основные сущности: Усыновитель, Животные и Медицинская книжка. Усыновитель содержит атрибуты (полное имя, email, телефон, адрес, дата регистрации) и связан с многими животными, каждое из которых характеризуется своим набором атрибутов (имя, дата поступления, вид, порода, возраст, пол, статус, описание). Один усыновитель может усыновить несколько животных, но каждое животное в момент времени может быть усыновлено лишь одним усыновителем. Животные имеют одну медицинскую книжку, где зафиксированы медицинская история и вакцинации. Связи четко отражают бизнес-логику: усыновитель "усыновляет" животных, животное "имеет" медицинскую книжку, медицинская книжка "добавляется в" систему.

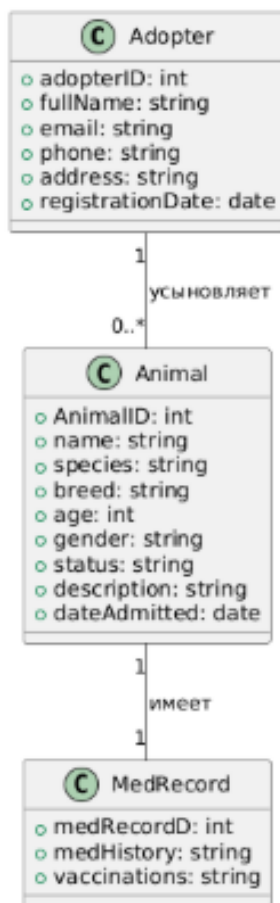


Рисунок 2. Пример как может выглядеть не нормализованная таблица

На этом этапе наши таблицы не нормализованы. Чтобы привести их к третьей нормальной форме (3NF), нужно разбить все неатомарные значения на ячейки (1NF), убрать частичные зависимости (2NF), убрать транзитивные зависимости (3NF). После проведения нормализации БД выглядит так (Рисунок 3):

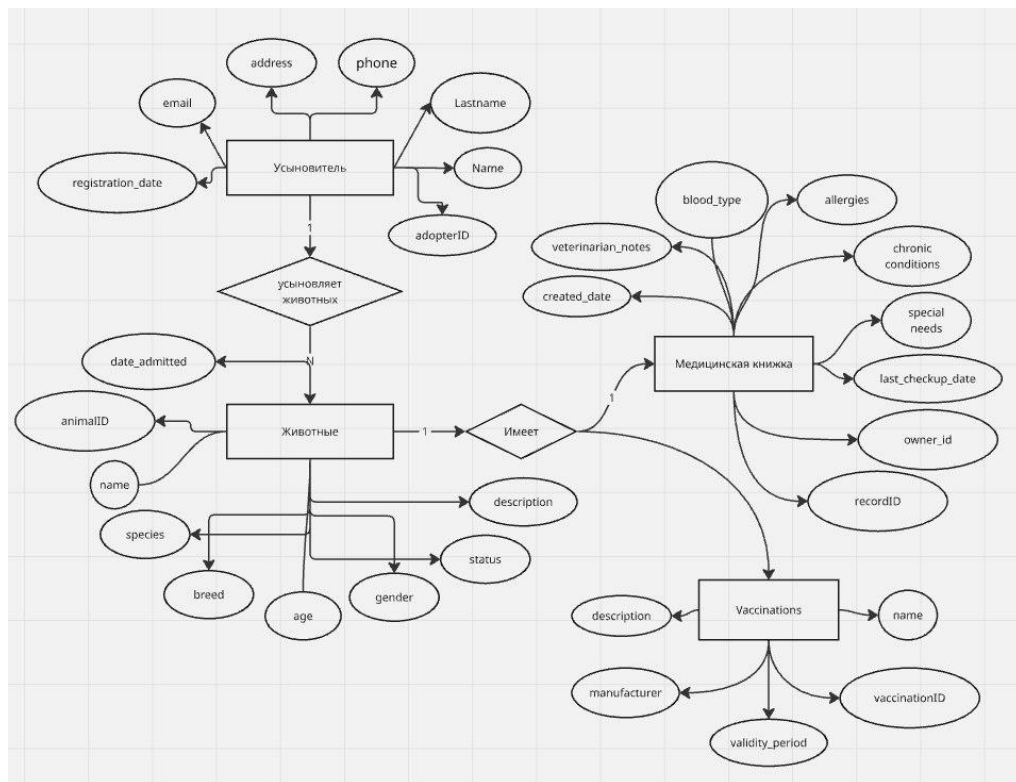


Рисунок 3. Нормализованная БД

Итоговый анализ нормальных форм:

1NF

- full\_name разбит на first\_name и last\_name
- Все поля атомарны

2NF

- Все таблицы имеют простые первичные ключи
- Нет частичных зависимостей

3NF

- Добавлен owner\_id в medical\_records для устранения транзитивных зависимостей
- Нету транзитивных зависимостей

В процессе написания таблиц была добавлена смежная таблица «animal\_vaccinations» и статусы были вынесены в отдельные таблицы «animal\_statuses», «adoption\_statuses». Результат показан ниже (Рисунок 4):

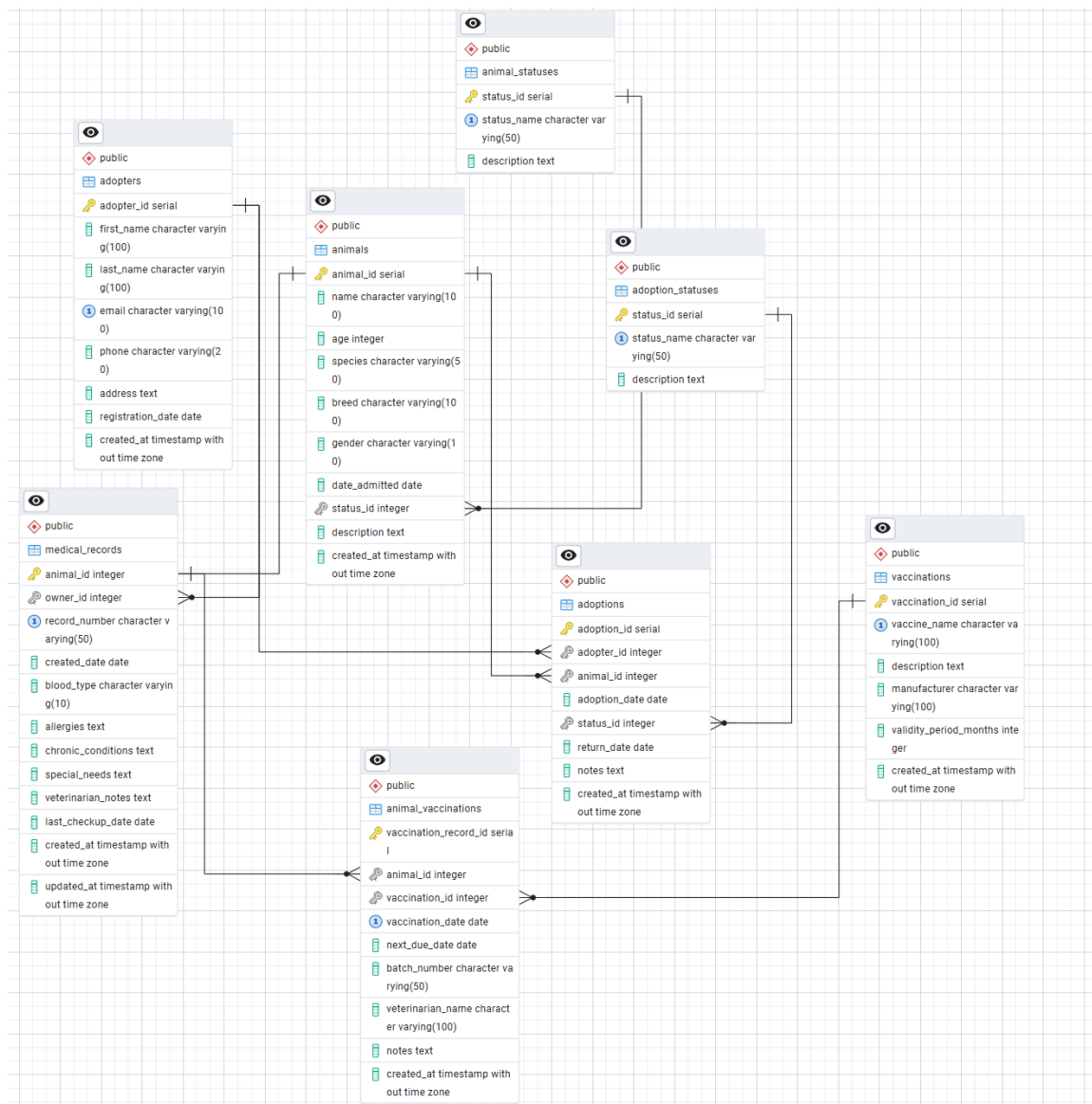
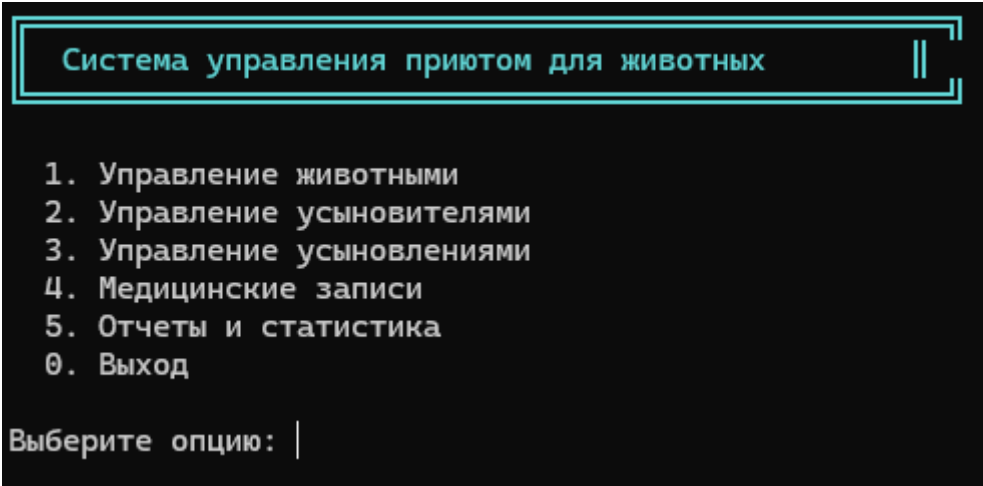


Рисунок 4. Скриншот из СУБД

# ИНТЕРФЕЙС ПРОГРАММЫ

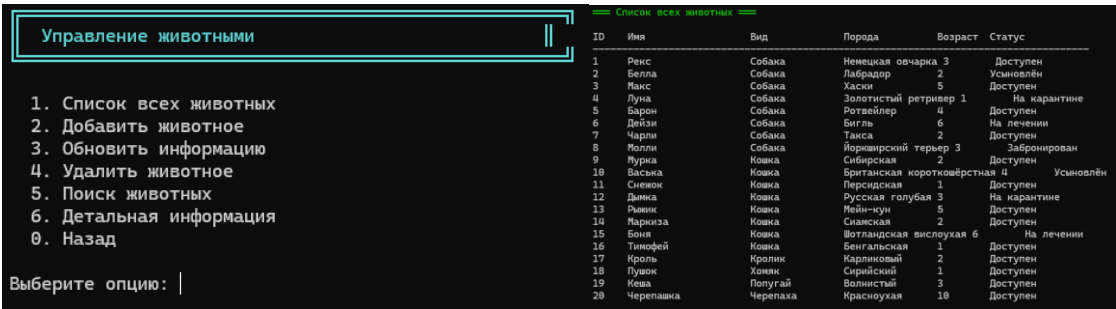
Для созданной БД было разработано CLI для взаимодействия с БД  
(Изображение 1):



Изображение 1. Консольный интерфейс

- 1- Переход в меню управления животными
- 2- Переход в меню управления усыновителей
- 3- Переход в меню управления усыновлений
- 4- Переход в меню управления медицинские записи животных
- 5- Переход в меню отчета и статистики приюта
- 0- Выход из программа

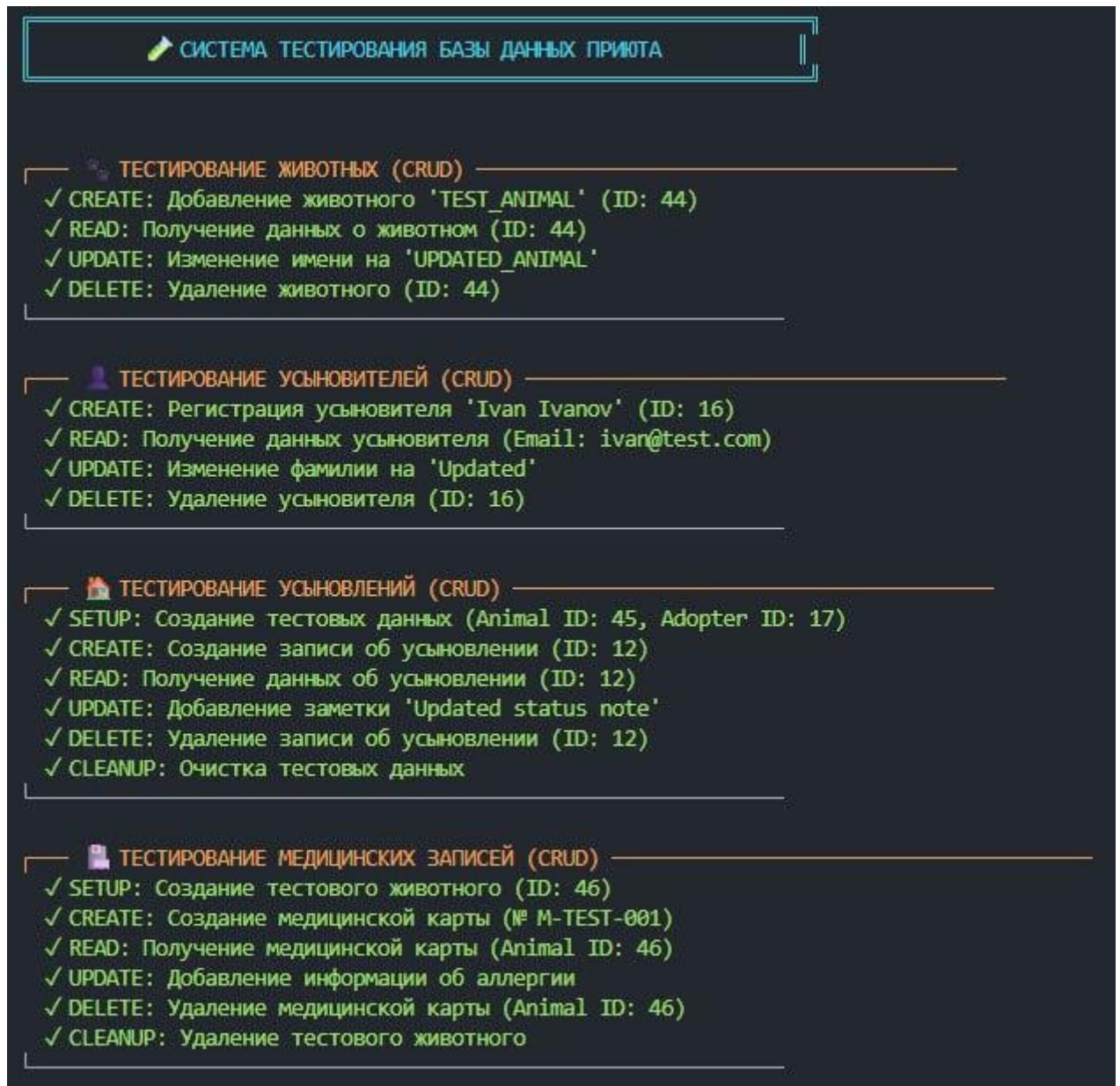
Пример того как выглядет меню управления (Изображение 2 и 3):



Изображение 2. Управления животными. Изображение 3. Список животных  
Санкт-Петербург, 2024

# ТЕСТИРОВАНИЕ

В проекте есть тестирование функций. В целом есть 20 тест-кейсов, которые все проходят успешно. Их можно увидеть ниже (Изображение 4):



Изображение 4. Пройденные тесты

Для подробного описания каждого теста, можно посмотреть их здесь:

[Test\\_Cases\\_Animal\\_Shelter.xlsx](#)



## ВЫВОДЫ

Проект «Animal Shelter Management System» успешно реализовал поставленные цели по созданию нормализованной базы данных и удобного консольного интерфейса для управления приютом животных. Была разработана оптимальная схема БД, соответствующая принципам нормализации (3NF), что обеспечивает целостность данных, минимизирует избыточность и позволяет эффективно работать с информацией о животных, усыновителях, медицинских записях и процессах усыновления. Внедрение CLI с интуитивной навигацией предоставляет пользователям полный контроль над операциями приюта, включая добавление, редактирование, поиск и формирование отчетов.

Ключевым результатом стала надежная система, готовая к практическому использованию. Её модульная архитектура, поддержка ADO.NET и тщательное тестирование (20 успешных тест-кейсов) демонстрируют стабильность и соответствие требованиям предметной области. Проект может служить основой для дальнейшего расширения функционала, например, интеграции веб-интерфейса или дополнительных аналитических инструментов.

# ПРИЛОЖЕНИЕ КОД

Program.cs

```
using System;

using System.Linq; // Добавьте эту строку
using AnimalShelterCLI.Data;
using AnimalShelterCLI.Models; // Добавьте эту строку
using AnimalShelterCLI.UI;

namespace AnimalShelterCLI
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.OutputEncoding = System.Text.Encoding.UTF8;

            try
            {
                using var context = new ShelterContext();

                // Проверка подключения к БД
                if (context.Database.CanConnect())
                {
                    Console.ForegroundColor = ConsoleColor.Green;

                    Console.WriteLine("✓ Подключение к базе данных установлено");
                    Console.ResetColor();
                }
                else
                {
                    Console.ForegroundColor = ConsoleColor.Red;

                    Console.WriteLine("X Ошибка подключения к базе данных");
                    Console.ResetColor();

                    return;
                }
            }
        }
    }
}
```

```

// Инициализация статусов животных
if (!context.AnimalStatuses.Any())
{
    context.AnimalStatuses.AddRange(
        new AnimalStatus { StatusId = 1, StatusName = "Available" }
    );
}

// Инициализация статусов усыновления
if (!context.AdoptionStatuses.Any())
{
    context.AdoptionStatuses.AddRange(
        new AdoptionStatus { StatusId = 1, StatusName = "Pending" }
    );
}

context.SaveChanges();

var handlers = new MenuHandlers(context);

var mainMenu = new Menu("Система управления приютом для животных");

mainMenu.AddSubMenu("Управление животными", () =>
{
    var animalMenu = new Menu("Управление животными");
    animalMenu.AddItem("Список всех животных", handlers.ListAllAnimals);
    animalMenu.AddItem("Добавить животное", handlers.AddAnimal);
    animalMenu.AddItem("Обновить информацию", handlers.UpdateAnimal);
    animalMenu.AddItem("Удалить животное", handlers.DeleteAnimal);
    animalMenu.AddItem("Поиск животных", handlers.SearchAnimals);
    animalMenu.AddItem("Детальная информация", handlers.ViewAnimalDetails);
    return animalMenu;
});

mainMenu.AddSubMenu("Управление усыновителями", () =>
{
    var adopterMenu = new Menu("Управление усыновителями");

```

```

    adopterMenu.AddItem("Список усыновителей", handlers.ListAllAdopters);
    adopterMenu.AddItem("Регистрация нового усыновителя", handlers.AddAdopter);
    adopterMenu.AddItem("Обновить данные", handlers.UpdateAdopter);
    adopterMenu.AddItem("Детальная информация", handlers.ViewAdopterDetails);
    return adopterMenu;
});

mainMenu.AddSubMenu("Управление усыновлениями", () =>
{
    var adoptionMenu = new Menu("Управление усыновлениями");
    adoptionMenu.AddItem("Список усыновлений", handlers.ListAllAdoptions);
    adoptionMenu.AddItem("Создать запись об усыновлении", handlers.CreateAdoption);
    adoptionMenu.AddItem("Обновить статус", handlers.UpdateAdoption);
    return adoptionMenu;
});

mainMenu.AddSubMenu("Медицинские записи", () =>
{
    var medicalMenu = new Menu("Медицинские записи");
    medicalMenu.AddItem("Просмотр медицинской карты", handlers.ViewMedicalRecord);
    medicalMenu.AddItem("Создать медицинскую карту", handlers.CreateMedicalRecord);
    medicalMenu.AddItem("Добавить запись о вакцинации", handlers.AddVaccinationRecord);
    medicalMenu.AddItem("Просроченные вакцинации", handlers.ViewOverdueVaccinations);
    medicalMenu.AddItem("Справочник вакцин", handlers.ListAllVaccinations);
    medicalMenu.AddItem("Добавить вакцину", handlers.AddVaccination);
    return medicalMenu;
});

mainMenu.AddSubMenu("Отчеты и статистика", () =>
{
    var reportMenu = new Menu("Отчеты и статистика");
    reportMenu.AddItem("Отчет по животным", handlers.GenerateAnimalReport);
    reportMenu.AddItem("Отчет по усыновлениям", handlers.GenerateAdoptionReport);
    return reportMenu;
});

mainMenu.AddItem("Запустить тесты", () => new TestRunner(context).RunAllTests());

```

```
        mainMenu.Display();  
        Console.WriteLine("\nДо свидания!");  
    }  
    catch (Exception ex)  
    {  
        Console.ForegroundColor = ConsoleColor.Red;  
        Console.WriteLine($"Критическая ошибка: {ex.Message}");  
        Console.WriteLine($"Детали: {ex.StackTrace}");  
        Console.ResetColor();  
    }  
}  
}
```