

## Разработка клиент-серверного приложения для конструирования виртуального музея

*А. В. Лушников, Э. М. Вихтенко, Т. В. Жукова*

*Тихоокеанский государственный университет, Хабаровск*

**Аннотация:** В статье описывается методика разработки клиент-серверного приложения, предназначенного для конструирования виртуального музея. Подробно рассмотрено создание серверной части приложения с функциями обработки и выполнения запросов к клиентской части, а также создание базы данных и взаимодействия с ней. Клиентская часть разработана при помощи фреймворка Angular и языка TypeScript, трехмерная реализация основывается на библиотеке three.js, которая является надстройкой технологии WebGL. Серверная часть разработана на платформе ASP.NET Core на языке C#. Схема базы данных основана на подходе Code-First с использованием Entity Framework Core. В качестве системы управления базой данных используется Microsoft SQL Server.

**Ключевые слова:** клиент-серверное приложение, конструктор виртуального тура, виртуальный музей, библиотека three.js, фреймворк, Angular, ASP.NET Core, Entity Framework Core, Code-First, WebGL.

### Введение

В современном мире развитие информационных технологий по обеспечению удаленного доступа к данным в различных сферах жизнедеятельности человека является важным и актуальным. В работе исследуются технологии и инструменты, применяемые для создания виртуальных туров по музеям мира [1].

На сегодняшний день существует несколько программных средств для создания виртуальных музеев, но все они являются проприетарными. Как показывает анализ рынка, большинство существующих решений принадлежат зарубежным компаниям, которые, в свою очередь, закрыли доступ к приобретению данного рода программного обеспечения (ПО) из РФ. В связи с этим значимость реализация отечественных инструментов повышается. Стоит отметить, что важно не только повторить реализацию функций зарубежных аналогов, но и расширить возможности программного средства (ПС), на основе которых создать конкуренцию на мировом рынке. В

---

данном случае одним из преимуществ является использование современных технологий и подходов к разработке ПО, а также возможности полного изменения пользовательского интерфейса и даже логики работы виртуального тура для учета требований конкретных пользователей. В работе показано использование бесплатных инструментов, которые позволяют создавать виртуальные туры, используя технологию WebGL (Web-based Graphics Library) [2].

Виртуальный музей в разработанном программном средстве (ПС) представлен несколькими точками обзора, которые могут меняться при желании. Точкой обзора является камера, расположенная в центре сферы, для поверхности которой используется текстура. Текстурой, в данном случае, является сферическая панорама, созданная при помощи фотоаппарата, специальной аппаратуры и редактора фотографий (рис. 1).



Рис.1. – Панорама возле входа в центральном зале

Пользовательский интерфейс содержит панель на левой половине экрана, в которой располагаются карта с точками обзора и кнопки с выпадающими списками для переключения между сценами, как показано на рис. 2. Панель можно развернуть/свернуть при необходимости. На правой

половине экрана содержится скрываемая панель с информацией об экспонатах.

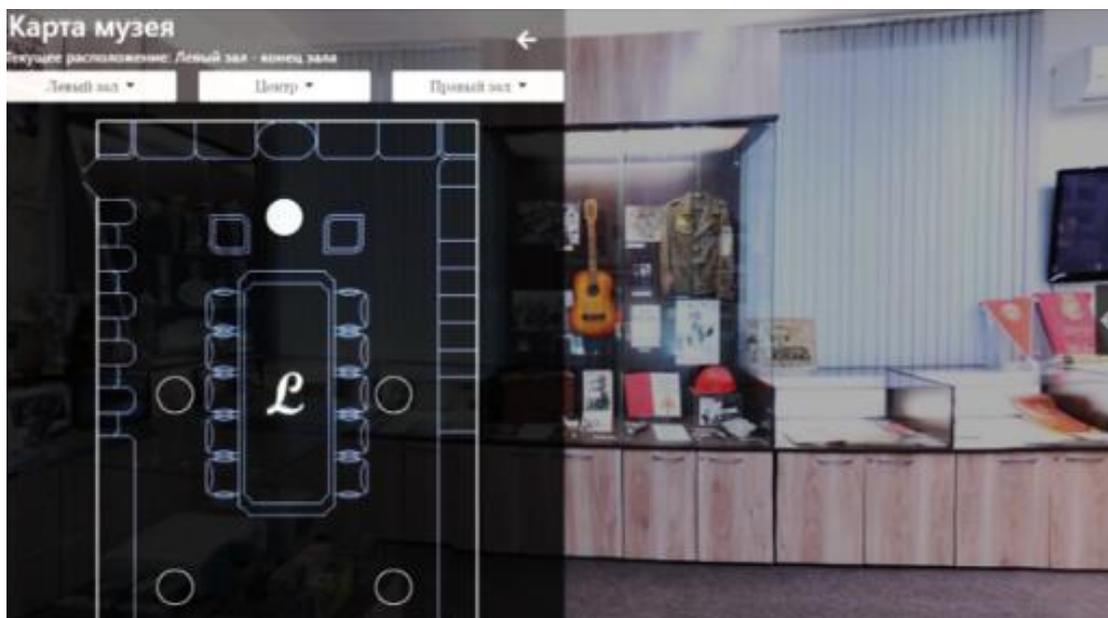


Рис. 2. – Пользовательский интерфейс ПС

Реализованный проект является базой для создания виртуального тура по музею Тихоокеанского государственного университета (ТОГУ), а также он может использоваться для конструирования любых других виртуальных туров.

Клиентская часть разработана при помощи фреймворка Angular [3] и языка TypeScript, трехмерная реализация основывается на библиотеке three.js [4], которая является надстройкой технологии WebGL. Эта часть проекта реализует основную логику для отображения трёхмерных сцен и взаимодействия с ними [5]. Серверная часть разработана на платформе ASP.NET Core [6] на языке C# [7]. Схема базы данных (БД) основана на подходе Code First с использованием Entity Framework Core [8]. В качестве системы управления БД используется Microsoft SQL Server. Все используемые технологии являются проектами с открытым исходным кодом.

## Описание серверной части проекта

С целью реализации конструктора виртуальных туров решены следующие задачи:

- разработана схема хранения информации;
- обеспечено выполнение основных CRUD-операций (create, read, update, delete) с данными;
- обеспечено развертывание ПС на нескольких платформах (кроссплатформенность);
- описаны методы выполняемых операций.

Схема БД для виртуального тура состоит из нескольких базовых элементов: этаж, комната, точка обзора, экспонат. На рис. 3 представлена диаграмма ERD (Entity Relationship Diagrams) [9] в нотации Crow's Foot.

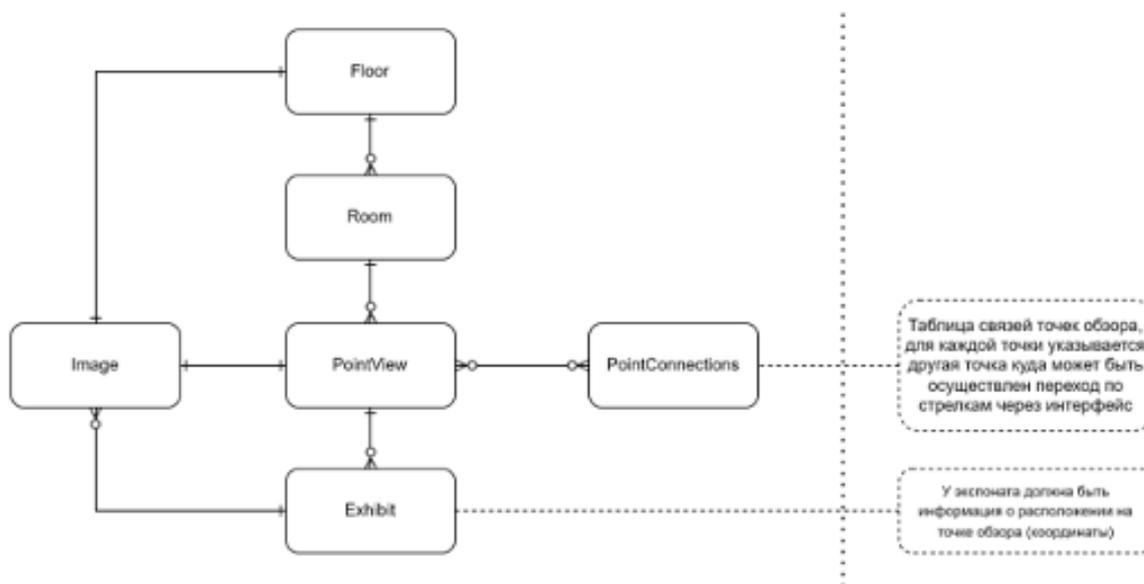


Рис. 3. – Схема связи сущностей БД для виртуального тура

Три сущности Floor (этаж), PointView (точка обзора) и Exhibit (экспонат) содержат сущность Image (изображение). Для этажа – это изображение схематичной карты, для точки обзора – текстура (сферическая панорама), для экспоната – его фотография.

Дополнительно в модель введена сущность PointConnections (связь точек обзора), которая связывает две различные точки обзора, из которых

есть прямой переход. Таким способом организован прямой переход через интерфейс на соседние точки обзора в виде стрелок, а не через карту этажа.

Для реализации методов работы с данными в БД используется логика http-запросов клиентской части к серверу [10], а именно выполняется следующий алгоритм:

1. Создать этаж (Floor);
2. Добавить для этажа карту (Image);
3. Добавить на этаж помещения (Room);
4. Для каждого помещения добавить точки обзора (PointView)
5. Для каждой точки обзора загрузить панораму (Image);
6. При наличии добавить на точку обзора экспонаты (Exhibit);
7. Для каждого экспоната загрузить необходимые изображения (Image) и добавить текст описания.

Помимо этого для каждой сущности реализована возможность ее редактирования, удаления и получения.

В связи с требованием создания БД посредством скриптов, использовался специальный подход разработки ПО – Code-First, при котором в первую очередь создается код, а затем на его основе генерируется база данных и модели данных. Код в стиле Code-First для модели PointView имеет следующий вид:

```
public class PointView : IBaseModel
{
    public Guid Id { get; set; } = Guid.NewGuid();
    public required string Name { get; set; }
    [ForeignKey("Room")]
    public Guid RoomId { get; set; }
    public virtual required Room Room { get; set; }
    [ForeignKey("Image")]
```

```
public Guid? TextureImageId { get; set; }  
public virtual Image? Image { get; set; }  
public virtual ICollection<Exhibit> Exhibits { get; set; }  
[ForeignKey("Parent")]  
public Guid? ParentId { get; set; }  
public virtual PointView? Parent { get; set; }  
[InverseProperty("Parent")]  
public virtual ICollection<PointView> ChildViews { get; set; }  
}
```

На рис. 4 отображена схема БД при просмотре в Microsoft SQL Server Management Studio. На схеме отображены ключи и столбцы таблицы PointView и видно, что они соответствуют модели, представленной в коде приложения.

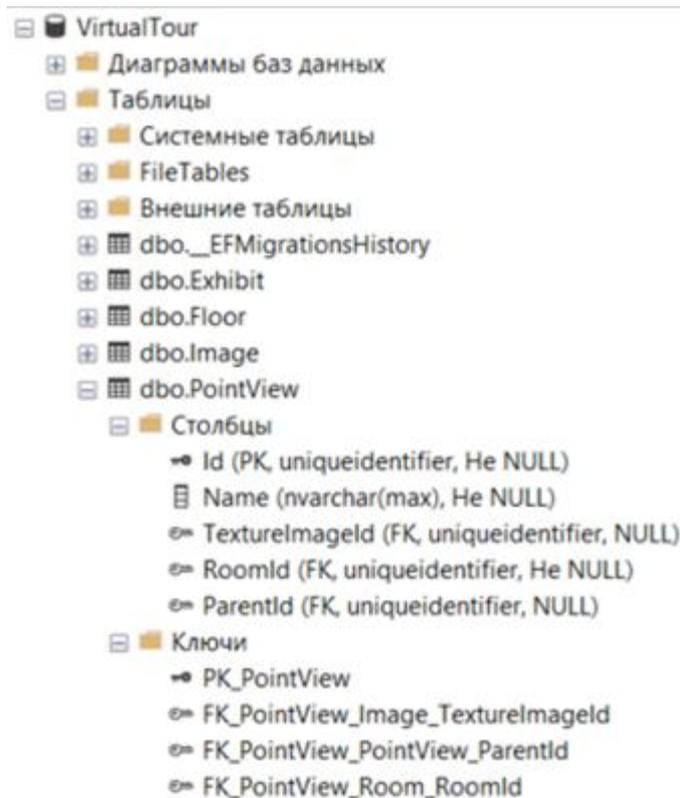


Рис. 4. – База данных VirtualTour

Для работы с каждой сущностью описывается отдельный контроллер, в котором происходит инициализация контекста для работы с БД.

### Клиент-серверное взаимодействие

В режиме конструирования виртуального тура пользователь сначала выбирает операцию создания этажа, для выполнения которой требуется заполнить форму (рис. 5), которая выводится в клиентской части.

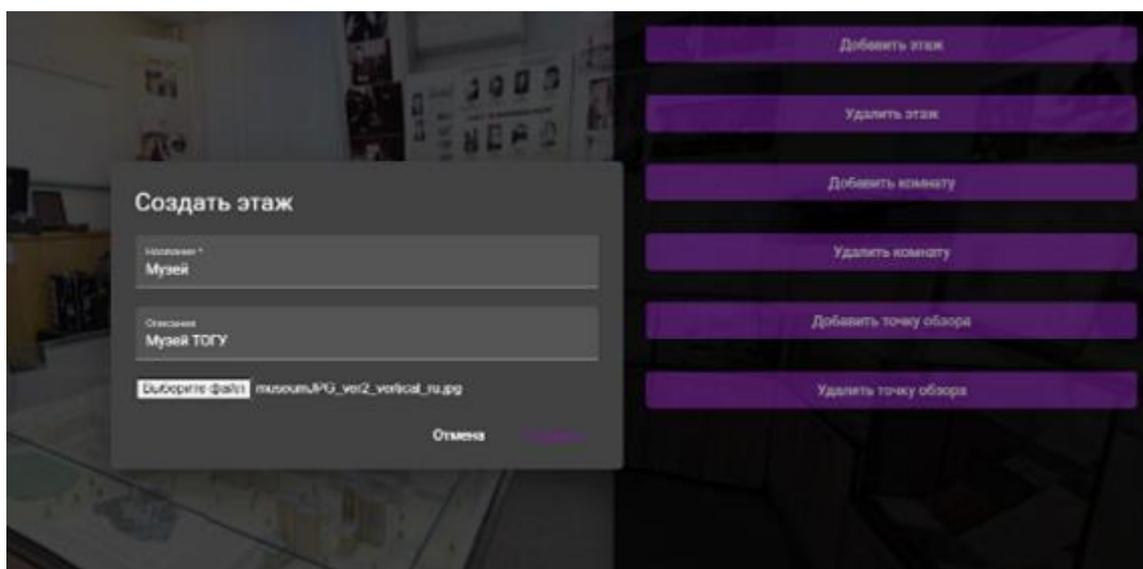


Рис. 5. – Форма создания этажа

После заполнения и подтверждения формы пользователем происходит отправка модели на серверную часть системы, сформированной согласно данным в форме (название, описание и изображение карты навигации). Затем на сервере происходит проверка модели, и, если она удовлетворяет условиям, происходит сохранение в БД. После этого сервер возвращает результат выполнения операции клиенту. После завершения операции пользователь может повторить этот процесс для создания нового этажа. На рис. 6 представлена диаграмма последовательности действий в нотации UML (Unified Modeling Language), описывающая клиент-серверное взаимодействие на примере создания этажа.

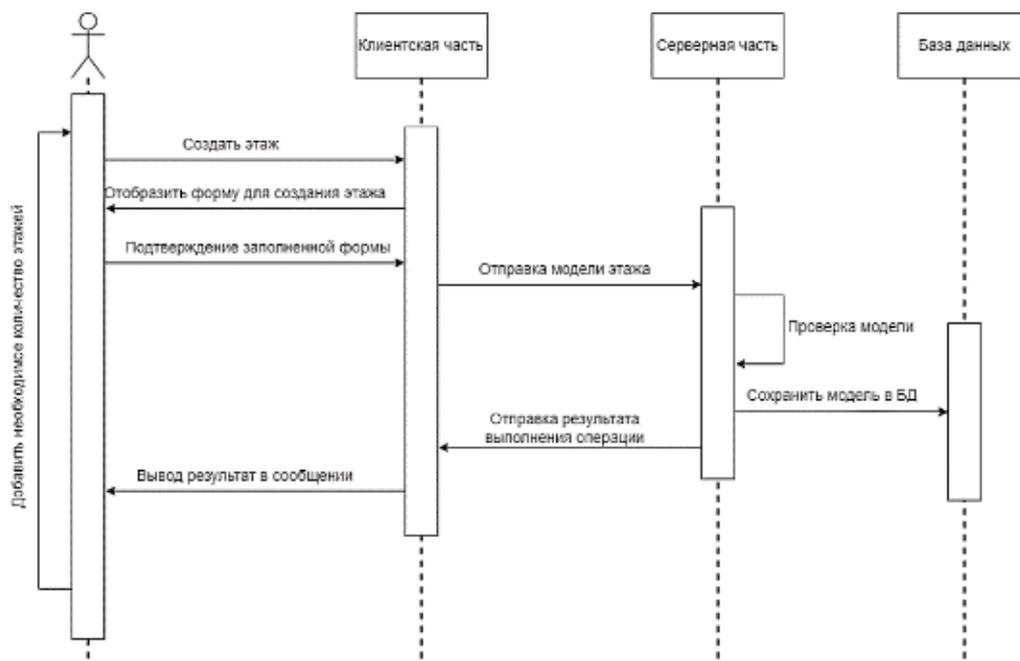


Рис. 6. – Диаграмма последовательности действий при создании этажа

Для виртуального тура изображения играют ключевую роль. Например, карта этажа обеспечивает более понятную навигацию по туру, для создания трехмерной сцены требуются текстуры, а также изображения экспонатов. Учитывая, что виртуальный тур будет динамически создаваться, а некоторые изображения могут меняться или обновляться довольно часто, принято решение хранить изображения в базе данных в виде байт-строки. В серверной части описан метод получения, обработки и сохранения в БД изображения, которое было отправлено в форме клиентской части. Так как в проекте есть три сущности, которые ссылаются на изображения, то в каждой прописан метод, который создает изображение и привязывает его к сущности. Таким образом обеспечивается конструирование виртуальных туров, позволяющее хранить данные в базе и получать их по требованию.

### Тестирование разработанного комплекса

Тестирование ПС проводилось на основе имеющихся панорам и карт расположения точек обзора по ранее реализованному проекту «Виртуальный музей ТОГУ».

На рис. 7 показана панель навигации, где отображаются добавленные в виртуальный музей этажи, комнаты, экспонаты и точки обзора.

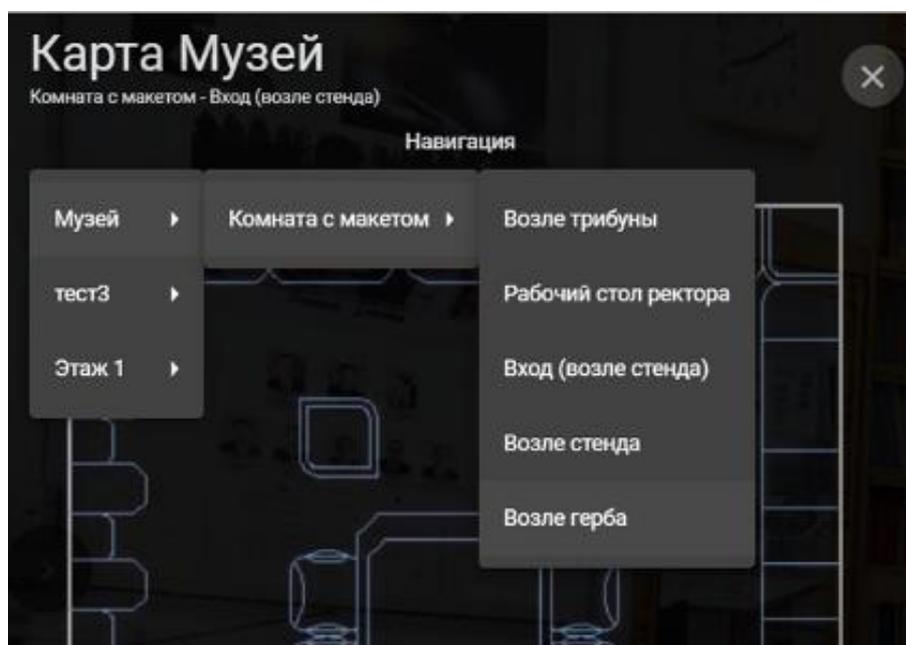


Рис. 7. – Панель навигации после создания точек обзора

Для проверки корректности навигации выполнялись последовательно переходы во все точки обзора, из которых можно рассмотреть экспонат. Например, обзор экспоната «Макет» можно осуществить с противоположных сторон из точки «Возле трибуны» и «Рабочий стол ректора». На рис. 8 и 9 продемонстрирован результат.



Рис. 8. – Точка обзора «Рабочий стол ректора»



Рис. 9. – Точка обзора «Возле трибуны»

### Заключение

В работе разработан программно-информационный комплекс для виртуальных музеев. К достоинствам разработки относятся следующие ее характеристики: программный продукт позволяет полностью разворачивать клиент-серверное приложение на собственных серверах; проект использует современные технологии для реализации логики обработки клиентских запросов и создания трёхмерных представлений визуальных компонент для доступа через сеть Интернет; проект использует библиотеки и другое программное обеспечение с открытым исходным кодом.

На данном этапе разработки проект позволяет создавать виртуальный тур, где точки обзора соединены друг с другом, но переход организован через схематическую карту, а не через интерактивные точки. По мнению авторов дальнейшие исследования в рассматриваемой сфере разработки российского ПО для конструирования виртуальных туров являются перспективными. Подобного рода разработки позволяют обеспечивать сохранение культурного и исторического наследия в электронном виде с использованием технологии трехмерного представления данных.

## Благодарность за финансовую поддержку работы

*Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации, дополнительное соглашение № 75-02-2024-1432 от 28 февраля 2024 г.*

## Литература

1. Технические рекомендации по созданию виртуальных музеев. Версия 1.0 // Минкультуры России. URL: [culture.gov.ru/documents/po-sozdaniyu-virtualnykh-muzeev-250714](http://culture.gov.ru/documents/po-sozdaniyu-virtualnykh-muzeev-250714).
  2. WebGL: 2D and 3D graphics for the web. URL: [developer.mozilla.org/en-US/docs/Web/API/WebGL\\_API](http://developer.mozilla.org/en-US/docs/Web/API/WebGL_API) (дата обращения: 10.05.2024).
  3. API List // Angular URL: [angular.io/api](http://angular.io/api) (дата обращения: 10.05.2024).
  4. Creating a scene // three.js URL: [threejs.org/docs/#manual/en/introduction/Creating-a-scene](http://threejs.org/docs/#manual/en/introduction/Creating-a-scene) (дата обращения: 10.05.2024).
  5. Лушников А.В. Миграция веб-проекта с нативного Javascript на фреймворк Angular // Far East Math - 2022: Материалы национальной научной конференции, Хабаровск, 22–26 ноября 2022 года. Хабаровск: Тихоокеанский государственный университет, 2022. С. 137-141.
  6. Lock, A., 2023. ASP.NET Core in Action, Third Edition. Manning. 984 p.
  7. Skeet, J., 2019. C# in Depth: Fourth Edition. Manning. 528 p.
  8. Smith, J.P., 2021. Entity Framework Core in Action, Second Edition. Manning. 624 p.
  9. Олейников Д.Г., Анисимова Г.Б. Разработка информационной системы для лингвистического центра // Инженерный вестник Дона. 2022. №4. URL: [ivdon.ru/ru/magazine/archive/n4y2022/7594](http://ivdon.ru/ru/magazine/archive/n4y2022/7594).
  10. Бондаренко Т.В., Федотов Е.А., Бондаренко А.В. Разработка HTTP сервера //Инженерный вестник Дона. 2018. №2. URL: [ivdon.ru/ru/magazine/archive/n2y2018/5038](http://ivdon.ru/ru/magazine/archive/n2y2018/5038).
-



## References

1. Tehnicheskie rekomendacii po sozdaniyu virtual'nyh muzeev. Versija 1.0 [Technical recommendations for the creation of virtual museums. Version 1.0]. Minkul'tury Rossii. URL: [culture.gov.ru/documents/po-sozdaniyu-virtualnykh-muzeev-250714](http://culture.gov.ru/documents/po-sozdaniyu-virtualnykh-muzeev-250714).
2. WebGL: 2D and 3D graphics for the web. URL: [developer.mozilla.org/en-US/docs/Web/API/WebGL\\_API](http://developer.mozilla.org/en-US/docs/Web/API/WebGL_API) (date assessed: 10.05.2024).
3. Angular. URL: [angular.io/api](http://angular.io/api) (date assessed: 10.05.2024).
4. three.js. URL: [threejs.org/docs/#manual/en/introduction/Creating-a-scene](http://threejs.org/docs/#manual/en/introduction/Creating-a-scene) (date assessed: 10.05.2024).
5. Lushnikov A.V. Far East Math - 2022: Materialy nacional'noj nauchnoj konferencii, Habarovsk, 22–26 nojabrja 2022 goda. Habarovsk: Tihookeanskij gosudarstvennyj universitet, 2022. Pp. 137-141.
6. Lock, A., 2023. ASP.NET Core in Action, Third Edition. Manning. 984 p.
7. Skeet, J., 2019. C# in Depth: Fourth Edition. Manning. 528 p.
8. Smith, J.P., 2021. Entity Framework Core in Action, Second Edition. Manning. 624 p.
9. Olejnikov D.G., Anisimova G.B. Inzhenernyj vestnik Dona. 2022. №4. URL: [ivdon.ru/ru/magazine/archive/n4y2022/7594](http://ivdon.ru/ru/magazine/archive/n4y2022/7594).
10. Bondarenko T.V., Fedotov E.A., Bondarenko A.V. Inzhenernyj vestnik Dona. 2018. №2. URL: [ivdon.ru/ru/magazine/archive/n2y2018/5038](http://ivdon.ru/ru/magazine/archive/n2y2018/5038).

**Дата поступления: 24.04.2024**

**Дата публикации: 25.06.2024**