

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина «Объектно-ориентированное программирование»

«К защите допустить»
Руководитель курсовой работы
_____ А.Г. Бокун
_____._____.2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
на тему:
**«ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ СИСТЕМА ВИЗУАЛИЗАЦИЙ
ПЛАНЕТАРНЫХ МОДЕЛЕЙ(НА ПРИМЕРЕ 3D-ГЛОБУСА ЗЕМЛИ)»**

БГУИР КП 6-05 0612 02 36 ПЗ

Выполнил студент группы 453503
ГОРБЕНКО Екатерина Сергеевна

_____ (подпись студента)

Курсовая работа представлена на
проверку _____._____.2026

_____ (подпись студента)

Минск 2026

ВВЕДЕНИЕ

В современном мире трехмерная компьютерная графика перестала быть просто инструментом для создания развлекательного контента. Она превратилась в важнейший элемент научного познания, образования и профессиональной деятельности. Особое место в этом ряду занимают системы планетарной визуализации — цифровые глобусы, позволяющие исследовать поверхность Земли и других небесных тел. Такие системы применяются для анализа рельефа в геологии и геодезии, планирования инфраструктурных проектов, метеорологических наблюдений, а также в образовательном процессе для изучения географии и астрономии .

Актуальность данной работы обусловлена необходимостью создания доступных и гибких инструментов для работы с геопространственными данными. Существующие коммерческие решения, такие как Google Earth или Цифровой глобус от «СканЭкс» , часто являются закрытыми системами. Их функциональность ограничена лицензионной политикой, а интеграция с собственными модулями или специфическими форматами данных затруднена или невозможна. Кроме того, они не всегда позволяют наглядно продемонстрировать студентам внутреннюю архитектуру подобных систем, что важно в образовательных целях.

Разработка собственной системы визуализации с использованием объектно-ориентированного подхода позволяет решить эти проблемы. Применение объектно-ориентированного программирования (ООП) является ключевым фактором при создании сложных графических приложений. Как отмечается в литературе, механизмы инкапсуляции, наследования и полиморфизма позволяют выстроить гибкую, расширяемую архитектуру, инвариантную по отношению к конкретным задачам, будь то моделирование механических узлов или построение планетарных моделей. Это дает возможность легко добавлять поддержку новых небесных тел, типов поверхностей или слоев данных без переписывания всего ядра программы.

Объектом исследования данной курсовой работы являются методы и алгоритмы трехмерной компьютерной графики.

Предметом исследования выступает применение объектно-ориентированной методологии для построения интерактивных систем визуализации планетарных моделей.

Целью работы является проектирование и разработка программной системы для визуализации трехмерной модели Земли с использованием объектно-ориентированного подхода.

Для достижения поставленной цели необходимо решить следующие задачи:

1 Провести анализ существующих систем-аналогов для визуализации планетарных моделей.

2 На основе анализа сформулировать функциональные требования к разрабатываемой системе.

3 Изучить теоретические основы объектно-ориентированного проектирования применительно к задачам компьютерной графики.

4 Спроектировать объектную модель приложения, выделить основные классы и взаимосвязи между ними.

5 Выбрать инструментальные средства и реализовать ключевые модули системы.

6 Провести тестирование разработанного программного обеспечения для проверки корректности его работы.

1 АНАЛИЗ АНАЛОГОВ

Проектирование объектно-ориентированной системы визуализации планетарных моделей требует предварительного исследования существующего ландшафта программных решений. В современной индустрии программного обеспечения задачи трехмерного рендеринга геопространственных данных решаются различными методами — от создания закрытых коммерческих экосистем до разработки открытых программных каркасов (frameworks). Проведение сравнительного анализа позволяет не только выявить функциональные преимущества конкурентов, но и определить наиболее эффективные архитектурные паттерны, которые могут быть адаптированы в рамках данной курсовой работы.

Наиболее массовым и технологически совершенным продуктом в рассматриваемой области является система Google Earth Pro. Данное решение базируется на инновационной для своего времени тайловой архитектуре, позволяющей динамически загружать фрагменты текстур и геометрии в зависимости от положения виртуальной камеры. С точки зрения конечного пользователя, Google Earth предоставляет беспрецедентный уровень детализации, интегрируя спутниковые снимки, данные аэрофотосъемки и трехмерные модели городских ландшафтов. Однако при детальном рассмотрении данного продукта с позиции разработчика выявляется ряд критических ограничений. Google Earth является закрытым проприетарным программным обеспечением, что полностью исключает возможность анализа его внутренней объектной модели. Использование предоставляемого API жестко ограничено лицензионной политикой корпорации и требует постоянного высокоскоростного соединения с удаленными серверами, что делает систему непригодной для использования в изолированных образовательных или специализированных инженерных комплексах..

В качестве альтернативы закрытым системам выступает проект NASA WorldWind, представляющий собой программный интерфейс для разработки геоинформационных систем с открытым исходным кодом. В отличие от готовых приложений, WorldWind спроектирован как расширяемый SDK, что делает его крайне интересным для изучения в контексте дисциплины объектно-ориентированного программирования. Архитектура системы построена на четком разделении ответственности между компонентами: математическая модель планеты инкапсулирована в отдельные классы эллипсоидов, а визуальные данные представлены в виде иерархии слоев. Такая структура позволяет разработчику внедрять собственные методы триангуляции и шейдерные эффекты через механизмы наследования. Несмотря на высокую научную точность и поддержку стандартов WGS84, текущее состояние проекта характеризуется смещением вектора разработки в сторону веб-технологий и языка Java, что создает определенные трудности при необходимости нативной интеграции в современные приложения на платформе .NET без использования ресурсоемких мостов.

Развитие веб-ориентированных графических стандартов привело к появлению библиотеки CesiumJS, которая на сегодняшний день является стандартом де-факто для браузерной визуализации глобусов. CesiumJS демонстрирует высокую эффективность в управлении большими массивами данных за счет применения спецификации 3D Tiles. Архитектурно система полагается на асинхронную модель обработки ресурсов, что критически важно для предотвращения блокировок пользовательского интерфейса при загрузке тяжелых текстур. Однако специфика среды исполнения (веб-браузер) накладывает ограничения на низкоуровневое управление памятью и прямое взаимодействие с графическим конвейером через такие API, как Vulkan или DirectX. Для целей курсового проектирования опыт CesiumJS полезен в части изучения алгоритмов автоматического выбора уровня детализации (LOD), но прямой перенос его логики в десктопную среду требует глубокой переработки системы типов и управления жизненным циклом объектов.

Отдельную категорию составляют астрономические симуляторы, такие как Stellarium и Celestia. В отличие от вышеупомянутых геоинформационных систем, данные продукты фокусируются на астрофизической точности позиционирования небесных тел и реализме атмосферных эффектов. В Celestia реализована развитая объектно-ориентированная модель, где каждое небесное тело является экземпляром класса, реализующего общий интерфейс физического взаимодействия и рендеринга. Это позволяет системе бесшовно обрабатывать как малые объекты (спутники, кометы), так и целые планетные системы. Сильной стороной этих решений является реализация алгоритмов атмосферного рассеяния и динамического освещения, однако они обладают низкой детализацией при визуализации поверхности планет на малых высотах, так как не используют сложные механизмы кэширования пространственных данных.

Проведенный сравнительный анализ позволяет сделать вывод, что на текущем рынке программного обеспечения наблюдается дефицит специализированных объектно-ориентированных систем, которые сочетали бы в себе легковесность, независимость от облачных сервисов и гибкость в расширении объектной модели для образовательных целей. Большинство существующих решений либо слишком перегружены функционалом, либо являются «черными ящиками», недоступными для модификации. Проектируемая в рамках данной работы система призвана занять нишу компактных и высокопроизводительных модулей визуализации. Основное внимание будет уделено реализации принципов слабой связности между модулем математических вычислений, отвечающим за аппроксимацию сферы, и графическим ядром. Такой подход позволит создать архитектуру, способную адаптироваться к различным типам планет и наборам визуальных данных, что полностью соответствует целям объектно-ориентированного проектирования и задачам данной курсовой работы. На основе выявленных преимуществ и недостатков аналогов в следующем разделе будут сформированы детальные требования к собственной программной системе.