МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования   
«САНКТ–ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой № 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| д-р физ.-мат. наук, проф. |  |  |  | В.Г.Фарафонов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

БАКАЛАВРСКАЯ РАБОТА

|  |  |
| --- | --- |
| на тему | Разработка файлового менеджера |
|  | |
|  | |

|  |  |
| --- | --- |
| выполнена | Новиковым Данилой Юрьевичем |
| фамилия, имя, отчество студента в творительном падеже | |

|  |  |  |  |
| --- | --- | --- | --- |
| по направлению подготовки | 09.03.03 |  | Прикладная информатика |
|  | код |  | наименование направления |
|  | | | |
| наименование направления | | | |
| направленности | 05 |  | Прикладная информатика |
|  | код |  | наименование направленности |
| в инновационной деятельности | | | |
| наименование направленности | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент группы № | М122 |  |  |  | Д. Ю. Новиков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Руководитель

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. экон. наук, доц. |  |  |  | С. В. Удахина |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования   
«САНКТ–ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

УТВЕРЖДАЮ

Заведующий кафедрой №\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ БАКАЛАВРСКОЙ РАБОТЫ

|  |  |  |  |
| --- | --- | --- | --- |
| студенту группы | М122 |  | Любавскому Даниилу Алексеевичу |
|  | номер |  | фамилия, имя, отчество |

|  |  |
| --- | --- |
| на тему | Разработка 3D калькулятора |
|  | |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| утвержденную приказом ГУАП от |  | № |  |

|  |  |
| --- | --- |
| Цель работы: | Разработать програмное обеспечение для визуализации конструкций и рассчета статической неопределимости |
|  | |
|  | |

|  |  |
| --- | --- |
| Задачи, подлежащие решению: |  |
|  | |
|  | |

|  |  |
| --- | --- |
| Содержание работы (основные разделы): | Исследование предметной области, |
| Разработка приложения, Оценка эффективности | |
|  | |
|  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Срок сдачи работы « |  | » |  | 20 | 25 |

Руководитель

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. экон. наук, доц. |  |  |  | С. В. Удахина |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

Задание принял(а) к исполнению

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| студент группы № | М122 |  |  |  | Д. Ю. Новиков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

РЕФЕРАТ

Разработка программного обеспечения для расчета статически неопределимых стержневых конструкций методом сил с визуализацией на базе DearPyGui и использованием расчетного ядра на C++

Выпускная квалификационная работа 1 страницы, 2 рисунка, 3 таблицы, 4 литературных источника, 5 приложений.

Ключевые слова:

Конструкции, статическая определимость, рассчет, 3D модели, разработка программы.

Цель работы – разработать програмное обеспечение для визуализации конструкций и рассчета статической неопределимости .

Объектом исследования является CAD система.

В процессе разработки использовались программные средства:

В ходе выполнения ВКР была разработана программа для визуализации 3D конструкций и рассчета статической неопределимости.

Область применения: инженирия

Экономическая эффективность: см. ниже

Наглядная работа с конструкциями и помощь при рассчете статической неопределимости.

**СОДЕРЖАНИЕ**

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 5](#_Toc198398965)

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ 6](#_Toc198398966)

[ВВЕДЕНИЕ 7](#_Toc198398967)

[Глава 1. Исследование предметной области 10](#_Toc198398968)

[1.1 Описание объекта исследования 10](#_Toc198398969)

[1.1.1 Far Manager 10](#_Toc198398970)

[1.1.2 Total Commander 11](#_Toc198398971)

[1.1.3 Проводник Windows 13](#_Toc198398972)

[1.2 Описание предмета исследования 15](#_Toc198398973)

[1.2.1. Алгоритм хеширования MD5 16](#_Toc198398974)

[1.2.2 Алгоритм хеширования SHA-1 18](#_Toc198398975)

[1.2.3 Алгоритм хеширования SHA-256 20](#_Toc198398976)

[1.3 Структуры файловых систем компьютера 21](#_Toc198398977)

[1.3.1 NTFS 22](#_Toc198398978)

[1.3.2 ext4 24](#_Toc198398979)

[1.3.3 Apple File System 25](#_Toc198398980)

[1.3.4 Сравнение упомянутых файловых систем 26](#_Toc198398981)

[1.4 Вывод по главе 1 28](#_Toc198398982)

[Глава 2. Разработка приложения 29](#_Toc198398983)

[2.1 Постановка задачи разработки 29](#_Toc198398984)

[2.2 Описание обеспечивающих подсистем разработки 29](#_Toc198398985)

[2.2.1 Выбор платформы для реализации 29](#_Toc198398986)

[2.2.2 Выбор языка разработки 30](#_Toc198398987)

[2.2.3 Выбор среды разработки 32](#_Toc198398988)

[2.3 Описание и реализация функциональной части программы 33](#_Toc198398989)

[2.3.1 Схема разработки учебника с использованием конструктора 33](#_Toc198398990)

[2.3.3 Разработка программы 35](#_Toc198398991)

[2.4 Разработка интерфейса приложения 38](#_Toc198398992)

[2.4.1 Требования к интерфейсу 38](#_Toc198398993)

[2.4.2 Интерфейс программы 40](#_Toc198398994)

[Глава 3. Оценка эффективности 41](#_Toc198398995)

[3.1 Тестирование приложения 41](#_Toc198398996)

[3.2 Обоснование экономической эффективности 41](#_Toc198398997)

[ЗАКЛЮЧЕНИЕ 42](#_Toc198398998)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 43](#_Toc198398999)

[ПРИЛОЖЕНИЕ А. Код 45](#_Toc198399000)

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В данной работе применяются следующие термины с соответствующими определениями:

Big-endian – порядок записи байтов в памяти от старшего к младшему;

Birthday-атака (Атака «дней рождения») – используемое в криптоанализе название для метода взлома шифров или поиска коллизий хеш-функций на основе парадокса дней рождения;

Аудит (в файловой системе) – ряд процессов для отслеживания изменений в файловой системе. Помогает предотвратить снижение уровня безопасности, программные сбои и неэффективное использование файловых ресурсов;

Графический интерфейс – интерфейс пользователя, представленный в виде графических элементов экрана;

Журналируемость – способность ведения журнала, в котором хранится список изменений. Данный журнал помогает сохранить целостность файловой системы при сбоях;

Интерфейс пользователя – совокупность средств, при помощи которых пользователь взаимодействует с различными программами и устройствами;

Квотирование – механизм ограничения доступного пользователям пространства в файловой системе;

Кластер – логическая единица хранения данных в таблице размещения файлов, объединяющая группу секторов;

Приложение – программа, предназначенная для решения практической задачи и подразумевающая непосредственное взаимодействие с пользователем;

Файловая система – порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах, а также в другом электронном оборудовании;

Файловый менеджер – компьютерная программа, предоставляющая интерфейс пользователя для работы с файловой системой и файлами;

Хеш-сумма – последовательность символов фиксированной длины, полученная путём преобразования произвольных исходных данных при помощи специального математического алгоритма.

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В данной работе применяются следующие сокращения и обозначения:

ACL – Access Control List;

APFS – Apple File System;

ext4 – Fourth Extended File System;

FAT – File Allocation Table;

FAR – File and Archive manager;

GUI – Graphical User Interface;

IDEF – Integrated Defiition;

NTFS – New Technology File System;

MD5 – Message Digest 5;

SHA-1 – Secure Hash Algorithm 1;

SHA-256 – Secure Hash Algorithm 256;

TC – Total Commander;

АНБ – Агентство Национальной Безопасности;

ГОСТ – Межгосударственный стандарт;

ИТ – Информационные Технологии;

ОС – Операционная система;

ПК – Персональный компьютер;

ПО – Программное обеспечение;

ЯП – Язык программирования.

PDF – Portable Document Format;

# ВВЕДЕНИЕ

**Актуальность темы**  
Современное строительство и машиностроение предъявляют повышенные требования к надежности и экономичности конструкций. Статически неопределимые системы, обладающие значительными преимуществами по сравнению с определимыми (повышенная жесткость, лучшая устойчивость, перераспределение усилий), находят широкое применение в практике проектирования. Однако их расчет сопряжен со значительными математическими трудностями, требующими применения специальных методов анализа и сложных вычислений. В этой связи разработка программных средств, автоматизирующих процесс расчета и визуализации результатов, представляет особую актуальность для образовательного процесса и инженерной практики.

**Цель и задачи ВКР**  
Целью работы является разработка программного обеспечения для расчета и визуализации статически неопределимых конструкций.

Для достижения поставленной цели решаются следующие задачи:

1. Анализ методов расчета статически неопределимых систем
2. Разработка математической модели метода сил
3. Создание алгоритмов численной реализации расчетных методов
4. Разработка пользовательского интерфейса для ввода данных и визуализации результатов
5. Тестирование и верификация разработанного программного обеспечения

**Объект и предмет исследования**  
Объектом исследования являются статически неопределимые стержневые системы (рамы, балки, фермы).  
Предметом исследования выступают алгоритмы расчета и визуализации статически неопределимых конструкций, реализованные в разработанном программном обеспечении.

**Методы исследования**  
В работе применялись:

* Методы строительной механики (метод сил, метод перемещений)
* Численные методы линейной алгебры
* Методы объектно-ориентированного программирования
* Принципы UX/UI-дизайна
* Методы тестирования программного обеспечения

**Структура работы**  
Выпускная квалификационная работа состоит из введения, трех глав, заключения и списка литературы:

1. Теоретические основы расчета статически неопределимых систем
2. Разработка математической модели и алгоритмов
3. Программная реализация и тестирование

Разработанное программное обеспечение позволяет автоматизировать трудоемкие расчеты, наглядно представлять результаты анализа и может быть использовано в учебном процессе при изучении строительной механики, а также в инженерной практике на этапах предварительного проектирования.

# Глава 1. Исследование предметной области

**1.1Общая характеристика статически неопределимых конструкций**

Статически неопределимыми называются конструкции, в которых количество неизвестных усилий (реакций опор или внутренних сил) превышает число независимых уравнений равновесия статики. Такие системы требуют дополнительных условий для своего расчета, учитывающих деформации элементов конструкции.

#### **Основные особенности:**

1. **Наличие "лишних" связей**
   * В отличие от статически определимых систем, где все усилия можно найти только из уравнений равновесия, в неопределимых конструкциях часть связей является избыточной. Эти связи повышают жесткость и надежность конструкции, но усложняют расчет.
2. **Зависимость от деформаций**
   * Для решения необходимо учитывать не только условия равновесия, но и **совместность деформаций** (перемещений). Это требует анализа физических и геометрических характеристик материала (модуль упругости, моменты инерции сечений и т. д.).
3. **Повышенная надежность**
   * Благодаря избыточным связям такие конструкции обладают:
     + Большей устойчивостью к нагрузкам.
     + Перераспределением усилий при локальных повреждениях.
     + Лучшей сопротивляемостью динамическим воздействиям (вибрациям, ударам).
4. **Типичные примеры**
   * Многопролетные неразрезные балки.
   * Рамы с жесткими узлами.
   * Арки и фермы с дополнительными стержнями.
   * Железобетонные монолитные конструкции (плиты, колонны).

### Особенности расчета статически неопределимых конструкций

Расчет статически неопределимых систем имеет ряд принципиальных особенностей, отличающих его от анализа статически определимых конструкций:

1. **Необходимость учета деформаций**  
   Основная особенность - невозможность определить усилия только из уравнений статики. Требуется дополнительно рассматривать:

* Условия совместности деформаций
* Физические уравнения (закон Гука)
* Геометрические характеристики сечений

1. **Выбор основной системы**  
   Ключевой этап расчета - преобразование исходной системы в статически определимую путем:

* Удаления "лишних" связей
* Замены их неизвестными усилиями (силами или моментами)
* Контроля геометрической неизменяемости основной системы

1. **Составление канонических уравнений**  
   Для каждого удаленного элемента связи составляется уравнение:  
   δ₁₁X₁ + δ₁₂X₂ + ... + Δ₁ₚ = 0  
   где:

* δᵢⱼ - перемещения от единичных усилий
* Δᵢₚ - перемещения от внешней нагрузки
* Xᵢ - искомые неизвестные усилия

1. **Вычисление коэффициентов**  
   Определение коэффициентов требует:

* Построения единичных и грузовых эпюр
* Вычисления интегралов Мора или использования формулы Верещагина
* Учета жесткостных характеристик всех элементов

1. **Проверка решения**  
   Обязательные этапы контроля:

* Статическая проверка (удовлетворение уравнений равновесия)
* Деформационная проверка (совместность перемещений)
* Баланс работы внешних и внутренних сил

1. **Особенности различных методов**

* Метод сил: удобен при небольшой степени статической неопределимости
* Метод перемещений: эффективен для рамных систем с множеством узлов
* Смешанный метод: комбинирует преимущества первых двух

1. **Современные вычислительные аспекты**  
   При программной реализации важно:

* Оптимизировать решение систем уравнений
* Обеспечить точность численных методов
* Реализовать удобный ввод исходных данных
* Визуализировать результаты расчетов

Главная сложность расчета заключается в необходимости одновременного учета статических, геометрических и физических условий, что требует комплексного подхода и тщательного контроля на всех этапах решения.

### 1.2. Математическая модель: метод сил

Метод сил является классическим подходом к расчету статически неопределимых систем, основанным на принципе суперпозиции и концепции "лишних" неизвестных. Суть метода заключается в последовательном выполнении следующих операций:

1. Определение степени статической неопределимости системы (n) как разности между количеством неизвестных реакций и числом независимых уравнений статики.
2. Выбор основной системы путем удаления n лишних связей и замены их соответствующими неизвестными усилиями X₁, X₂,..., Xₙ.
3. Формулировка условий совместности деформаций, обеспечивающих эквивалентность основной и исходной систем.

**Уравнения равновесия**

Для основной системы составляются канонические уравнения метода сил, отражающие условия равенства нулю перемещений по направлениям удаленных связей:

δ₁₁X₁ + δ₁₂X₂ + ... + δ₁ₙXₙ + Δ₁ₚ = 0  
δ₂₁X₁ + δ₂₂X₂ + ... + δ₂ₙXₙ + Δ₂ₚ = 0  
...  
δₙ₁X₁ + δₙ₂X₂ + ... + δₙₙXₙ + Δₙₚ = 0

где:

* δᵢⱼ - перемещение в направлении i-ой связи от единичного усилия Xⱼ=1 (коэффициент при неизвестном);
* Δᵢₚ - перемещение в направлении i-ой связи от внешней нагрузки (свободный член).

**Формирование системы уравнений**

Коэффициенты системы определяются через интегралы Мора:  
δᵢⱼ = ∫(MᵢMⱼ/EI)dx + ∫(NᵢNⱼ/EA)dx + ∫(QᵢQⱼ/GF)dx  
Δᵢₚ = ∫(MᵢMₚ/EI)dx + ∫(NᵢNₚ/EA)dx + ∫(QᵢQₚ/GF)dx

где M, N, Q - соответственно изгибающие моменты, продольные и поперечные силы от единичных воздействий и внешней нагрузки.

Для практических расчетов часто применяют:

1. Способ Верещагина - перемножение эпюр
2. Численные методы интегрирования
3. Матричные алгоритмы (при компьютерной реализации)

Полученная система линейных алгебраических уравнений решается относительно неизвестных Xᵢ, после чего определяются окончательные значения внутренних усилий в элементах конструкции.

## 1.3 Описание объекта исследования

### 1.3.1 SAP2000

**SAP2000 (Structural Analysis Program 2000)** – программный комплекс для расчета и проектирования строительных конструкций, разработанный американской компанией **CSI (Computers and Structures, Inc.)**. Первая версия вышла в **1995 году**, и с тех пор программа претерпела множество изменений, став одним из самых популярных инструментов в области инженерной механики и строительного проектирования.

Основное предназначение SAP2000 – **статический и динамический анализ строительных конструкций**: от простых балок и рам до сложных пространственных сооружений, мостов, высотных зданий и промышленных объектов. Программа работает в **графическом интерфейсе**, где пользователь может создавать модель конструкции, задавать нагрузки, граничные условия и материалы, после чего выполнять анализ и получать результаты в визуальной форме.

SAP2000 использует **метод конечных элементов (МКЭ)**, поддерживает широкий спектр расчетов: линейные и нелинейные, модальные, спектральные, сейсмические, тепловые и другие. Предусмотрена возможность **автоматической генерации расчетных схем**, а также **интеграции с AutoCAD, Revit и Excel**, что делает работу инженера более гибкой и производительной.

Отдельное внимание разработчики уделили **визуализации и пользовательскому опыту**: присутствует трехмерное моделирование, анимация результатов, редактируемые графики, отчеты и таблицы. Интерфейс доступен на нескольких языках, включая русский, а документация охватывает как базовые примеры, так и сложные инженерные задачи.

SAP2000 может применяться как в учебных целях (имеется студенческая лицензия), так и в **проектных институтах и международных инженерных компаниях**, где требуется высокая точность и соответствие мировым стандартам (AISC, Eurocode, СНиП и др.).

Несмотря на свою мощность и универсальность, SAP2000 сохраняет **интуитивный интерфейс**, позволяющий быстро начать работу, а также **возможность автоматизации задач через API** (на языках VBScript, Python и C#).

Таким образом, SAP2000 остаётся востребованным и активно развивающимся инструментом, незаменимым в арсенале инженера-конструктора, проектировщика или исследователя в области строительной механики.

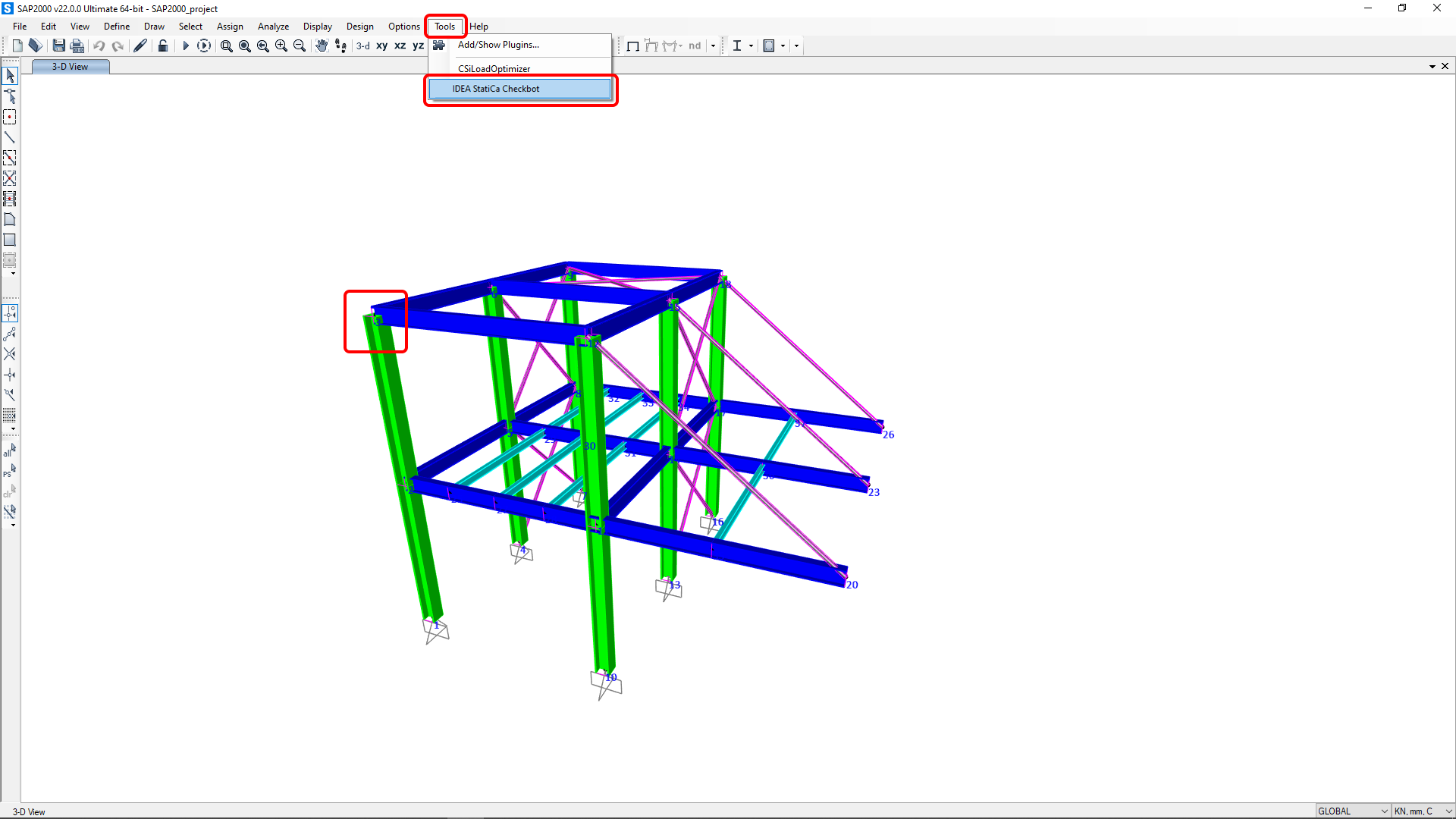


Рисунок 1 - Внешний вид Far Manager

### 1.3.2 ****LIRA-SAPR****

**LIRA-SAPR (ЛИРА-САПР)** – профессиональный программный комплекс для **расчета и проектирования строительных конструкций**, разработанный российской компанией **ЛИРА Софт**. Первая версия появилась в **1990-х годах**, и с тех пор программа стала одним из ведущих инструментов в области строительной механики, активно применяемым на территории России, СНГ и за их пределами.

Программа предназначена для выполнения **инженерных расчетов по методу конечных элементов (МКЭ)** и охватывает весь спектр задач, возникающих при проектировании зданий и сооружений – от простых схем до сложных многоуровневых конструкций, включая сейсмостойкие объекты, мосты, промышленные установки и высотные здания.

LIRA-SAPR поддерживает **линейный и нелинейный анализ**, статические и динамические расчёты, расчёт устойчивости, теплопередачи, прогрессирующего обрушения, а также **автоматическую проверку и подбор армирования** по нормативам (СП, СНиП, Eurocode и др.). Одним из ключевых преимуществ комплекса является **встроенный механизм генерации расчетной схемы на основе BIM-модели**, а также **интеграция с Revit, AutoCAD, IFC и другими CAD-системами**.

Интерфейс LIRA-SAPR сочетает **мощные аналитические инструменты и наглядную 3D-визуализацию**, позволяя пользователю быстро ориентироваться в структуре модели, редактировать параметры и отслеживать результаты в виде таблиц, графиков и цветных карт усилий и перемещений. Специальный модуль **"АРМ конструктора"** облегчает работу с результатами и оформлением проектной документации.

Программа реализует **модульный подход**, где пользователь может подключать только те компоненты, которые необходимы для конкретной задачи: от расчетов металлических и железобетонных конструкций до анализа грунтового основания и совместной работы системы "здание-основание".

**Преимуществами LIRA-SAPR** являются:

* Полная локализация на русском языке и поддержка нормативов СНиП/СП;
* Подробная документация и техническая поддержка;
* Возможность работы с крупными моделями и распределённые расчёты;
* Развитые инструменты автоматизации (скрипты, API, плагин для Grasshopper);
* Применяется в учебных заведениях и аккредитован в экспертизе проектной документации.

Несмотря на обилие функций, начальный порог освоения программы может быть выше, чем у некоторых аналогов, из-за высокой степени детализации настроек. Тем не менее, LIRA-SAPR остаётся одним из самых мощных и гибких решений для **профессионального проектирования и анализа строительных конструкций в русскоязычной инженерной среде**.

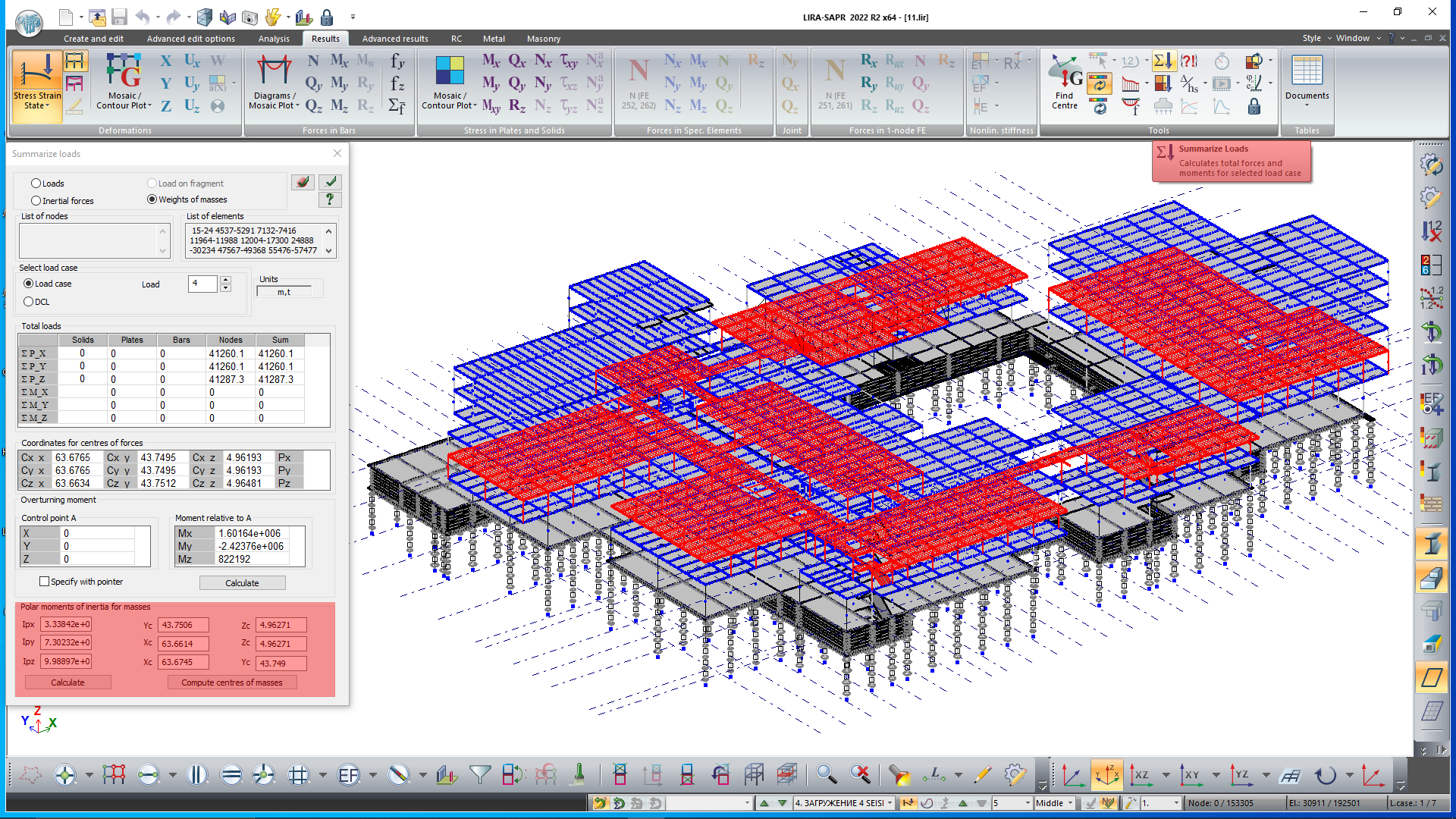


Рисунок 2 - Внешний вид Total Commander (на ОС Windows 7)

### 1.3.3 ****Frame3DD****

**Frame3DD** — это **кроссплатформенный программный пакет для анализа стержневых конструкций**, предназначенный для проведения **плоского и пространственного расчета рам, ферм и 3D-конструкций** методом конечных элементов. Проект был изначально разработан **автором Симоном Уэлсом (Simon Winther Wold)** и распространяется как **открытое программное обеспечение (open source)** под лицензией GPL. Благодаря своей простоте и гибкости, Frame3DD пользуется популярностью среди инженеров, студентов и исследователей, которым требуется легковесный, но точный расчетный инструмент.

Frame3DD работает через **командную строку**, что делает его особенно удобным для **автоматизации** и интеграции в расчетные пайплайны. Входные данные задаются в виде текстового файла с описанием узлов, элементов, нагрузок и граничных условий. После выполнения расчета программа создает выходные текстовые файлы с усилиями, перемещениями, реакциями и другими характеристиками.

Несмотря на отсутствие графического интерфейса, Frame3DD поддерживает экспорт данных в формат, совместимый с **gnuplot, MATLAB, Excel** и другими визуализаторами, а также может быть легко встроен в Python-скрипты для массовой генерации и анализа конструкций.

Программа поддерживает:

* **Гравитационные и сосредоточенные нагрузки**, моменты, распределенные нагрузки;
* **Геометрическую нелинейность** (в том числе эффект П–Δ);
* **Связи (constraints), жесткие диафрагмы, начальные деформации, температурные эффекты**;
* Расчеты на **устойчивость и модальные (собственные) частоты колебаний**;
* Возможность анализа **многоузловых элементов с различными сечениями и свойствами**.

**Преимущества Frame3DD:**

* Полностью бесплатный и с открытым исходным кодом;
* Быстрый, легкий и не требующий установки (работает из консоли);
* Отлично подходит для **автоматизации** инженерных расчетов и исследований;
* Поддерживает **нелинейный расчет** и базовые динамические анализы;
* Простая и понятная структура входных файлов.

**Недостатки:**

* Отсутствие встроенной графики: пользователю требуется сторонняя программа для визуализации;
* Нет поддержки сложных материалов или бетонной арматуры;
* Требуется подготовка текстовых файлов вручную или через скрипты;
* Интерфейс и документация полностью на английском.

Несмотря на свою минималистичность, **Frame3DD является надежным и быстрым инструментом**, который может быть использован как в академических целях, так и для быстрого предварительного расчета реальных инженерных задач. Особенно популярен среди **инженеров-энтузиастов**, разрабатывающих собственные фреймворки анализа или интеграции с CAD/BIM.

## 1.4 Вывод по главе 1

В первой главе были рассмотрены теоретические основы и существующие программные средства, применяемые для расчета статически неопределимых конструкций. Дано определение таким конструкциям, приведены их особенности и сложности расчета, а также подробно рассмотрен **метод сил** как один из наиболее распространенных аналитических методов в задачах статики.

Метод сил позволяет решать задачи расчета, опираясь на уравнения равновесия и совместности деформаций. Приведено математическое описание метода, включая формирование основной системы уравнений и подходы к ее решению. Также рассмотрены примеры его применения, что формирует необходимую базу для программной реализации.

Дополнительно выполнен обзор популярных программных комплексов (Лира-САПР, SAP2000, Frame3DD), выявлены их ключевые характеристики, достоинства и ограничения. Показано, что, несмотря на широкие возможности таких систем, они могут быть избыточны и сложны для учебных целей и начального ознакомления с расчетными методиками.

Таким образом, можно сделать вывод, что существует потребность в разработке специализированного, интуитивно понятного программного инструмента, который позволил бы выполнять базовые расчеты по методу сил с возможностью визуализации и упрощенного взаимодействия с пользователем. Это и стало основой для постановки задачи разработки, представленной в следующей главе.

# Глава 2. Разработка приложения

### ****2.1. Постановка задачи разработки****

В рамках данной выпускной квалификационной работы поставлена задача создания программного обеспечения, предназначенного для выполнения расчетов статически неопределимых стержневых конструкций методом сил с последующей визуализацией результатов. Основной целевой аудиторией разрабатываемого программного продукта являются студенты и преподаватели технических вузов, изучающие курс «Сопротивление материалов».

Существующие профессиональные системы автоматизированного проектирования (САПР), такие как Лира-САПР или SAP2000, обладают широкой функциональностью, но зачастую избыточны и сложны для первичного изучения расчетных методов. Предлагаемое решение призвано стать учебным инструментом, обеспечивающим более прозрачное понимание логики расчета и наглядную визуализацию внутренних усилий, деформаций и реакций опор.

В отличие от крупных САПР-систем, разрабатываемое приложение фокусируется не на проектировании сложных сооружений, а на демонстрации принципов расчета статически неопределимых систем. Оно должно быть простым в использовании, кроссплатформенным, интуитивно понятным и легко расширяемым.

**Цель разработки:** создать легковесное и доступное программное обеспечение, выполняющее следующие функции:

* Построение модели стержневой конструкции с возможностью задания геометрии, граничных условий и внешних нагрузок;
* Проведение численного расчета внутренней системы усилий с использованием метода сил;
* Визуализация расчетной схемы, нагрузок и результатов в графическом интерфейсе;
* Обеспечение конфигурируемости интерфейса и расширяемости функциональности;
* Обеспечение высокой производительности расчетов за счет использования библиотеки, написанной на языке программирования C++.

Таким образом, поставленная задача сводится к реализации архитектурно модульного приложения с разделением графического интерфейса и расчетного ядра, что обеспечивает удобство разработки, тестирования и дальнейшего масштабирования проекта.

### ****2.2. Выбор технологий и средств реализации****

Разработка программного обеспечения велась с учетом требований к простоте использования, кроссплатформенности, высокой производительности расчетов и удобной визуализации. В связи с этим были выбраны современные и широко применяемые инструменты, обеспечивающие необходимую функциональность при минимальной сложности развертывания.

#### **Язык Python для графического интерфейса**

В качестве основного языка для реализации пользовательского интерфейса выбран язык программирования Python версии 3.13. Данный выбор обусловлен следующими причинами:

* высокая читаемость и простота синтаксиса;
* богатая экосистема библиотек для визуализации, обработки данных и взаимодействия с другими языками;
* активное сообщество и наличие подробной документации;
* кроссплатформенность (Windows, Linux, macOS).

Разработка велась в среде Visual Studio Code под управлением ОС Linux. При этом программное обеспечение сохраняет возможность запуска и на платформе Windows при наличии соответствующих библиотек и зависимостей.

#### **Библиотека DearPyGui для визуализации**

Для построения графического интерфейса и визуализации конструкций использована библиотека **DearPyGui**. Это современная и активно развиваемая библиотека на основе ImGui, ориентированная на создание визуально насыщенных и интерактивных интерфейсов. Среди ее преимуществ:

* высокая производительность (на уровне C++) благодаря использованию графического API;
* простота интеграции в проекты на Python;
* поддержка кастомизации интерфейса (темы, стили, поведение);
* наличие встроенных инструментов для построения графических элементов: линий, стрелок, подписей, графиков.

В рамках проекта DearPyGui применяется для отображения расчетной схемы, опор, внешних нагрузок и результатов расчета.

#### **Язык C++ для расчетного ядра**

Для выполнения вычислительно затратных операций, связанных с решением систем линейных уравнений, формированием матриц жесткости и применением метода сил, используется модуль, реализованный на языке **C++ (стандарт C++17)**. Компиляция производилась компилятором **GCC 14.2.1**.

Выбор C++ продиктован необходимостью обеспечения высокой производительности при расчете, особенно на больших объемах данных. Взаимодействие между Python и C++ реализовано посредством технологии **ctypes**, которая позволяет вызывать функции из скомпилированной динамической библиотеки (.so или .dll) напрямую из Python-кода.

Такой подход позволяет отделить расчетную часть от интерфейсной логики, повысить масштабируемость и упростить отладку.

#### **Git — система контроля версий**

Во время разработки активно использовалась распределённая система контроля версий **Git**, обеспечивающая:

* ведение истории изменений;
* возможность возврата к стабильным версиям;
* удобство работы в команде (в перспективе);

Использование Git особенно полезно при многомодульной структуре проекта и экспериментировании с различными реализациями расчетных алгоритмов.

#### **Формат хранения данных: JSON**

Для хранения параметров модели (геометрия конструкции, опоры, нагрузки) используется формат **JSON (JavaScript Object Notation)**. Данный формат обладает следующими достоинствами:

* читаемость как для машины, так и для человека;
* поддержка в большинстве языков программирования;
* легкость сериализации и десериализации данных;
* гибкость структуры (поддержка вложенных объектов и массивов).

Файл модели может быть легко передан между различными версиями приложения или импортирован в другие системы.

Структура файла модели будет подробно рассмотрена в подразделе 2.3 при описании архитектуры приложения.

### ****2.3. Архитектура приложения****

Разрабатываемое приложение имеет модульную архитектуру, обеспечивающую разделение логики визуализации и вычислений, а также расширяемость и читаемость кода. Все компоненты взаимодействуют друг с другом по четко определённым интерфейсам, что упрощает поддержку и возможную доработку программы в будущем.

#### **Структура проекта**

Проект условно разделен на следующие основные компоненты:

* **Графический интерфейс пользователя (GUI)** — реализован на Python с использованием библиотеки DearPyGui. Отвечает за отображение расчетной схемы, сбор данных от пользователя и визуализацию результатов.
* **Расчетное ядро** — отдельная библиотека, написанная на C++. Содержит реализацию алгоритмов по методу сил, операции с матрицами жесткости, решение СЛАУ и обработку граничных условий.
* **Модуль взаимодействия (интерфейс связи)** — реализован на Python с использованием библиотеки ctypes. Отвечает за формирование структуры входных данных, передачу их в C++-библиотеку и получение результатов.
* **Файлы конфигурации и модели** — хранятся в формате JSON. Содержат описание конструкции: узлы, элементы, опоры, нагрузки, настройки визуализации и пр.

Пример структуры проекта:

#### **Взаимодействие модулей: Python и C++**

Взаимодействие между Python и C++ происходит следующим образом:

1. Python-часть формирует структуру входных данных, основываясь на пользовательском вводе (координаты узлов, типы опор, приложенные силы и т.д.).
2. Эти данные сериализуются в структуры C (например, массивы или структуры struct) с помощью ctypes и передаются в экспортируемые функции из .so/.dll библиотеки.
3. C++-библиотека выполняет расчет и возвращает результаты (перемещения, реакции, усилия в элементах).
4. Python принимает результаты и отображает их в интерфейсе.

Такой подход минимизирует потери производительности и обеспечивает удобную интеграцию низкоуровневого кода в высокоуровневую оболочку.

#### **Схема обмена данными между интерфейсом и расчетным ядром**

Формат входных и выходных данных стандартизирован и основан на простой структуре:

* **Входные данные:**
  + Массив координат узлов
  + Связи между узлами (элементы)
  + Типы опор (закрепления)
  + Внешние силы и моменты
* **Выходные данные:**
  + Перемещения узлов
  + Реакции опор
  + Продольные силы, изгибающие моменты, поперечные силы в элементах

# Для сохранения и загрузки моделей применяется формат JSON. Пример фрагмента JSON-файла показан в приложении.

## 2.4 Разработка интерфейса приложения

## 2.4.1 Требования к интерфейсу

К итоговому графическому интерфейсу приложения для расчета статически неопределимых конструкций, в соответствии с положениями стандартов **ГОСТ Р ИСО 9241-161-2016** и **ГОСТ Р ИСО 14915-1-2016**, предъявляются следующие требования:

Согласно **ГОСТ Р ИСО 9241-161-2016**, интерфейс должен соответствовать ряду ключевых принципов:

* **Доступность** — элементы интерфейса должны использовать доступные средства, предоставляемые операционной системой и графическим фреймворком (в данном случае DearPyGui), обеспечивая совместимость с дополнительными технологиями (экранные лупы, голосовой ввод и т.д.).
* **Взаимосвязь методов ввода и графических элементов** — ввод данных (например, координаты узлов, характеристики сечений, величины нагрузок) должен быть интуитивно связан с отображаемыми визуальными элементами (узлы, стержни, опоры).
* **Отображение состояний элементов** — каждый элемент интерфейса (например, выбранный стержень, активная опора, ошибка ввода) должен иметь чётко различимое состояние: цветовое выделение, изменение формы или контуров, наличие подсказки.

В соответствии с **ГОСТ Р ИСО 14915-1-2016**, интерфейс приложения должен соблюдать следующие принципы:

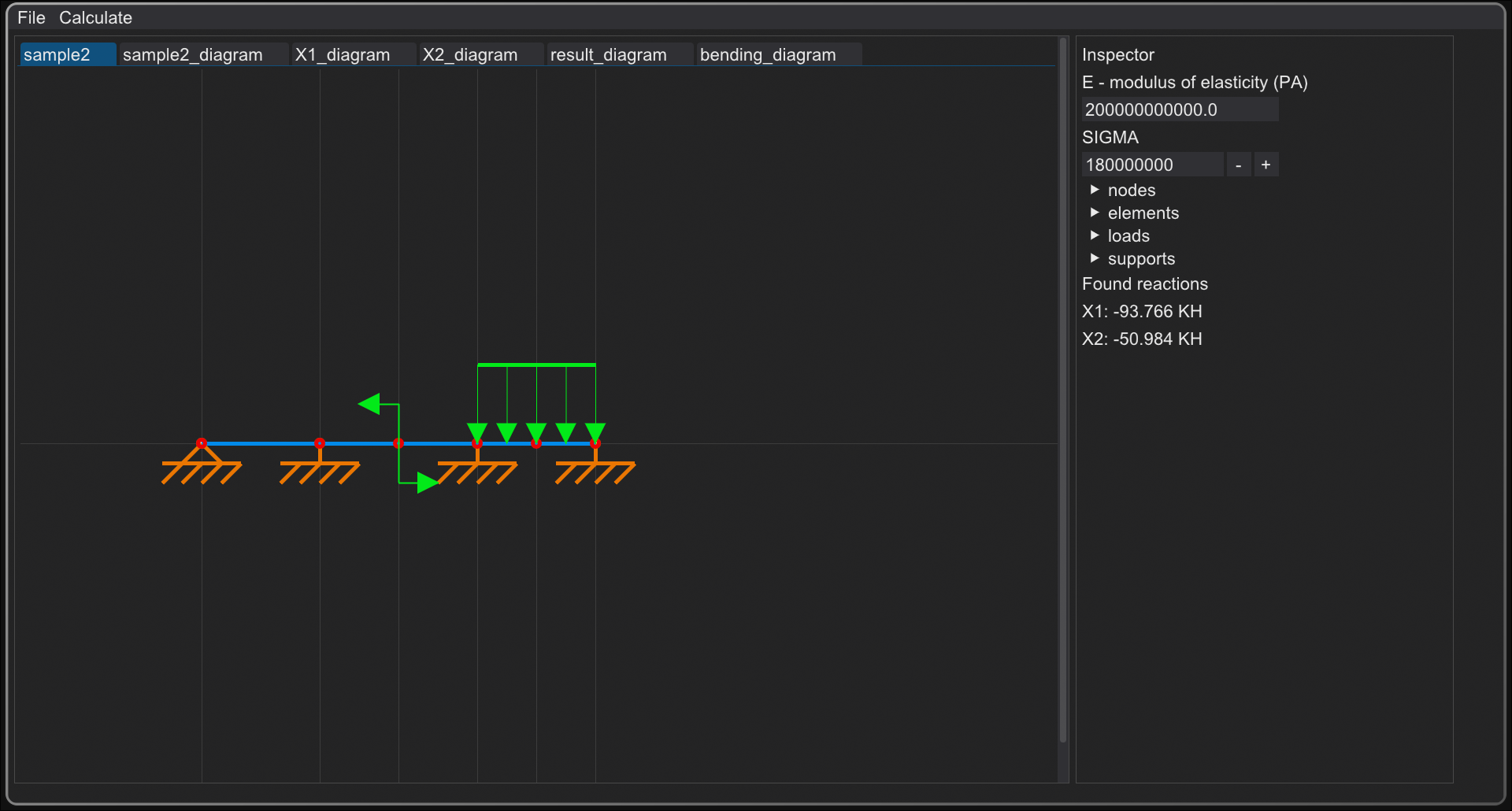
* **Информативность** — интерфейс должен отображать достаточно данных о модели и результатах расчета: геометрия конструкции, опоры, нагрузки, численные результаты.
* **Управляемость** — пользователь должен иметь контроль над действиями: добавление/удаление элементов, выполнение пересчета, сохранение модели.
* **Соответствие ожиданиям пользователя** — взаимодействие с интерфейсом должно быть предсказуемым: кнопки, окна и графические действия должны работать интуитивно.
* **Устойчивость к ошибкам** — программа должна обрабатывать некорректные действия (например, неверные координаты или повторяющиеся узлы) с выдачей соответствующего уведомления.
* **Пригодность для изучения** — интерфейс должен быть понятен для пользователей, не имеющих опыта в профессиональных САПР.
* **Пригодность для восприятия и понимания** — визуальные элементы должны быть логично организованы, не перегружены информацией.
* **Привлекательность** — внешний вид должен быть минималистичным и современным, без излишних визуальных перегрузок.

На основании этих принципов, к интерфейсу разработанного приложения выдвигались следующие практические требования:

* **Отображение расчетной схемы** в виде графической сцены: узлы, стержни, опоры и нагрузки визуализируются с помощью примитивов DearPyGui.
* **Интерактивный ввод параметров**: через боковую панель пользователь может задавать координаты, жесткости, нагрузки и типы опор.
* **Возможность изменения внешнего вида**: реализована поддержка настройки темы оформления (цвета фона, сетки, текста), а также масштабирования сцены.
* **Подсказки при наведении**: большинство элементов интерфейса сопровождаются всплывающими подсказками с пояснениями.
* **Универсальность**: интерфейс кроссплатформенный — успешно работает как в Linux, так и в Windows-среде.
* **Простота взаимодействия**: навигация по сцене, добавление новых элементов и запуск расчета организованы через понятные кнопки и поля ввода.
* **Конфигурируемость**: пользователь может загрузить или сохранить модель в формате JSON, что облегчает повторное использование и обмен данными.

Таким образом, разрабатываемый интерфейс соответствует ключевым требованиям эргономики и удобства, делая программу понятной и доступной для студентов и преподавателей дисциплин, связанных с сопроматом.

## 2.4.2 Интерфейс программы

  
Рисунок 1. — Внешний вид программы.

Интерфейс приложения построен по классической схеме инженерного ПО: в центре расположена основная рабочая область с визуализацией расчетной схемы, сверху — система вкладок, а справа — панель инспектора параметров.

**1. Верхняя панель меню**

* Содержит пункты File и Calculate, отвечающие за загрузку/сохранение расчетных моделей и запуск вычислений.
* Позволяет управлять текущей сессией, в том числе открывать сохранённые примеры и инициировать повторный расчет.

**2. Система вкладок**

* Над основной областью размещены вкладки:
  + sample2 — активная вкладка, содержащая расчетную схему.
  + sample2\_diagram, X1\_diagram, X2\_diagram, result\_diagram, bending\_diagram — вкладки для визуализации различных расчетных диаграмм (например, продольных сил, перемещений, моментов).
* Такая архитектура позволяет удобно переключаться между исходной моделью и результатами расчета.

**3. Основная графическая сцена**

* Центральная часть окна отведена под отображение расчетной схемы:
  + Стержни показаны синими отрезками.
  + Узлы помечены красными кругами.
  + Опоры представлены в виде классических инженерных символов (горизонтальные штрихи с наклонными линиями) оранжевого цвета.
  + Нагрузки визуализированы зелеными стрелками: вертикальные треугольники обозначают равномерно распределённые нагрузки, одиночные стрелки — сосредоточенные силы.
* Сцена масштабирована и выровнена по сетке, обеспечивая визуальную читаемость и точность.

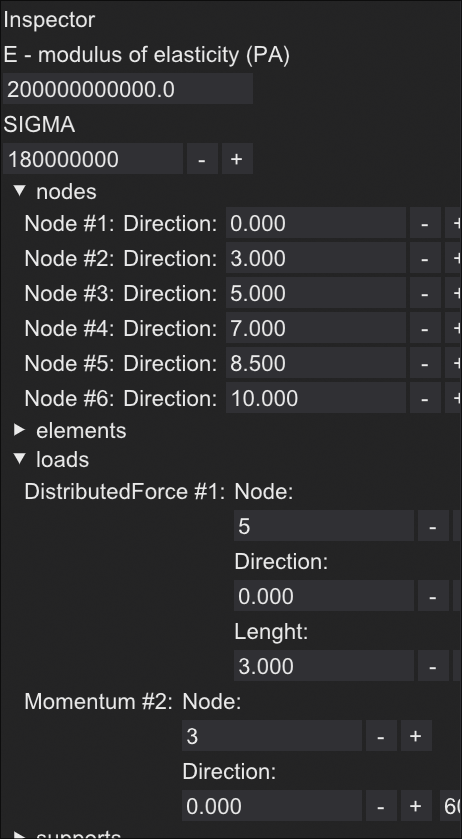
**4. Правая панель инспектора**

* Заголовок: Inspector.
* Содержит поля для задания параметров модели:
  + E - modulus of elasticity (PA) — модуль упругости материала.
  + SIGMA — допуски или напряжения.
* Ниже — раскрывающиеся списки:
  + nodes, elements, loads, supports — позволяют вручную просматривать и редактировать параметры отдельных компонентов модели.
* Блок Found reactions выводит результаты расчета опорных реакций:
  + X1: -93.766 кН
  + X2: -50.984 кН

**5. Цветовая схема и стиль**

* Интерфейс выполнен в тёмной теме с высокой контрастностью.
* Основные элементы визуально разделены: графика конструкции, численные значения и структура управления сведены в отдельные зоны.
* Использование цветовых кодов (оранжевый — опоры, зелёный — нагрузки, синий — стержни) делает визуализацию интуитивно понятной.

Ввод параметров производится в окне инспектора:



# Глава 3. Оценка эффективности

## 3.1 Проверка работоспособности приложения

**3.2 Тестирование интерфейса**

**3.3 Ограничения и возможные улучшения**

Разработанное приложение представляет собой работоспособный прототип программного комплекса для расчета статически неопределимых стержневых систем с использованием метода сил. Тем не менее, в текущей версии имеются ряд ограничений, связанных как с функциональностью, так и с удобством использования.

**Ограничения текущей версии:**

* Поддерживаются только плоские стержневые схемы (2D), без учета пространственных конструкций (3D).
* Ввод и редактирование схемы ограничены простым интерфейсом — отсутствует полноценный графический редактор (перетаскивание узлов, интерактивное добавление элементов и т.д.).
* Не реализована проверка на наличие ошибок в модели (например, избыточные связи, некорректные нагрузки, изолированные узлы).
* Реализована только одна расчетная методика — метод сил. Методы конечных элементов (МКЭ) или метод перемещений не используются.

**Направления для дальнейшего развития:**

* **Добавление новых расчетных методов**, в частности метода перемещений и простейшей реализации МКЭ.
* **Улучшение визуального редактора схемы**: ввод с помощью мыши, настройка геометрии, поворот элементов, редактирование связей и опор.
* **Поддержка 3D-схем** и пространственных конструкций.
* **Реализация постпроцессора**: автоматическая генерация отчетов.
* **Интеграция с другими форматами хранения данных**, в том числе совместимость с DXF, IFC или другими открытыми стандартами.
* **Создание пользовательских шаблонов**: библиотека типовых схем, возможность быстро собрать стандартную балку или раму

Таким образом, приложение может стать основой для создания образовательного или даже инженерного инструмента с расширенными возможностями. По мере развития проекта возможно как углубление расчетной части, так и повышение удобства интерфейса, что позволит сделать инструмент полезным не только для студентов, но и для практикующих инженеров.

# ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе была поставлена и решена задача разработки программного обеспечения для расчета статически неопределимых конструкций с использованием метода сил. Целью разработки являлось создание учебного инструмента, способного наглядно демонстрировать принципы работы метода сил, а также обеспечивать достаточную точность вычислений, сравнимую с промышленными САПР-системами.

В первой части работы был проведен обзор существующих решений, используемых в инженерной практике: Лира-САПР, SAP2000 и Frame3DD. Также был подробно изложен теоретический аппарат метода сил, включая формулировку уравнений равновесия, методику формирования системы уравнений и основные допущения.

Во второй части описан процесс разработки программного обеспечения, включающий выбор технологий, проектирование архитектуры приложения и реализацию графического интерфейса. Программа была создана с использованием Python и библиотеки DearPyGui для визуализации, а также C++ для выполнения вычислений. Для интеграции между языками использована библиотека ctypes. В качестве формата хранения моделей выбран JSON, что обеспечивает удобство хранения и гибкость расширения.

В третьей части проведено тестирование работоспособности приложения. Выполнено сравнение результатов, полученных с помощью разработанного инструмента, с результатами расчетов в Лира-САПР и SAP2000. Также проанализирована производительность расчетного модуля и проведено сравнение скорости вычислений на Python и C++. Были выявлены ключевые преимущества C++-реализации, особенно на больших моделях. Интерфейс программы протестирован с точки зрения удобства использования и логики взаимодействия с пользователем.

**Ключевые результаты:**

* Реализовано кроссплатформенное приложение для расчета плоских стержневых систем;
* Обеспечена визуализация расчетных схем и результатов;
* Поддержана интеграция Python и C++ для повышения производительности;
* Подготовлена база для дальнейшего расширения функциональности.

**Перспективы развития:**

Возможные направления развития проекта включают:

* Расширение на пространственные конструкции;
* Поддержку нелинейных нагрузок и нестандартных граничных условий;
* Импорт и экспорт моделей из форматов других САПР;
* Расширение визуализации (анимация, диаграммы усилий, динамика);
* Внедрение систем автоматического тестирования и CI/CD.

Разработанное программное обеспечение может использоваться в учебных целях, а также как прототип более функциональной системы для инженерных расчетов в малых проектных коллективах и стартапах.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

# ПРИЛОЖЕНИЕ А. Код

**ПРИЛОЖЕНИЕ Б. Пример файла модели**

{

"nodes": [

{

"id": 1,

"coordinates": [

0,

0

]

},

{

"id": 2,

"coordinates": [

1,

0

]

},

{

"id": 3,

"coordinates": [

2,

0

]

},

{

"id": 4,

"coordinates": [

3,

0

]

},

{

"id": 5,

"coordinates": [

4,

0

]

}

],

"elements": [

{

"id": 1,

"start\_node": 1,

"end\_node": 2

},

{

"id": 2,

"start\_node": 2,

"end\_node": 3

},

{

"id": 3,

"start\_node": 3,

"end\_node": 4

},

{

"id": 4,

"start\_node": 4,

"end\_node": 5

}

],

"loads": [

{

"id": 1,

"node": 4,

"type": "force",

"direction": [

0,

10

]

},

{

"id": 3,

"node": 5,

"type": "force",

"direction": [

0,

-2.81

]

},

{

"id": 2,

"node": 2,

"type": "distributed\_force",

"lenght": 2,

"direction": [

0,

10

]

}

],

"supports": [

{

"id": 1,

"node": 1,

"type": "pinned",

"direction": [0, 1]

},

{

"id": 2,

"node": 3,

"type": "roller",

"direction": [0, 1]

},

{

"id": 3,

"node": 5,

"type": "roller",

"direction": [0, 1]

}

]

}