



РАЗРАБОТКА ПРОГРАММНОГО ПРИЛОЖЕНИЯ ДЛЯ РАСЧЕТА БАЛОЧНЫХ КОНСТРУКЦИЙ

А. В. Савельева, студентка

E-mail: savelieva_nastia@mail.ru

ФГБОУ ВО «Калининградский государственный
технический университет»

Н. Е. Веселагин, студент

E-mail: veselaginn@mail.ru

ФГБОУ ВО «Калининградский государственный
технический университет»



Данная разработка направлена на автоматизацию расчетов при проектировании балочных конструкций, что позволяет существенно повысить точность и скорость выполнения инженерных задач и свести к минимуму риск ошибок, связанных с человеческим фактором. Внедрение такого приложения оптимизирует расходы времени и ресурсов, облегчает специалистам оперативный анализ и сравнение различных инженерных решений. Кроме того, разработка имеет определенную образовательную ценность: ее применение в учебных заведениях позволит эффективно готовить будущих специалистов по технической механике, будет способствовать более глубокому пониманию расчетных методик и совершенствованию практических навыков студентов.

Ключевые слова: автоматизация расчетов, проектирование балочных конструкций, повышение точности, скорость выполнения, минимизация ошибок, человеческий фактор, оптимизация времени, оптимизация ресурсов, оперативный анализ, сравнение инженерных решений, образовательная ценность, применение в учебных заведениях, подготовка специалистов, техническая механика, понимание расчетных методик, совершенствование практических навыков, студенты.

ВВЕДЕНИЕ

Автоматизация вычислений в проектировании балочных конструкций оказывает значительное влияние на повышение скорости и точности итоговых расчетов, поскольку исключается человеческий фактор и сводится к минимуму вероятность ошибок, часто возникающих при ручных вычислениях. Это особенно актуально при проектировании ответственных конструкций, где некорректные результаты могут привести к серьезным последствиям. Внедрение программных средств способствует и существенной экономии времени и ресурсов, так как позволяет значительно ускорить процесс проектирования, обеспечивая инженерам возможность оперативно анализировать и сравнивать различные варианты конструктивных схем. Кроме того, разработанное программное приложение обладает заметной образовательной ценностью: его можно эффективно использовать в инженерных и технических вузах для подготовки будущих специалистов в области технической механики. Использование такого инструмента в учебном процессе способствует более глубокому освоению методов расчета и формированию практических навыков, необходимых в профессиональной деятельности.

Новизна программного приложения заключается в том, что до разработки предлагаемого программного продукта расчеты балочных конструкций выполнялись с использованием различных методов и инструментов, включая ручные вычисления и общие вычислительные средства; предлагаемое приложение автоматизирует типичные случаи расчетов, объединяет функциональность и удобный интерфейс, что снижает трудозатраты и повышает скорость и наглядность получения результатов.

Разработанное программное обеспечение автоматизирует расчеты типичных схем балочных конструкций, что сокращает трудозатраты и ускоряет получение результатов по сравнению с ручными вычислениями и разрозненными вычислительными средствами.

Программа реализует алгоритмы анализа распространенных видов нагрузок и опор, предоставляет удобный интерфейс для ввода данных и визуализации результатов. А также добавление новых типов конструкций, усовершенствование методов расчета и расширение средств визуализации, что повышает ее практическую и образовательную ценность для студентов инженерных вузов.

Несмотря на наличие на рынке программных комплексов для расчета конструкций (таких как SCAD, ЛИРА-САПР), они зачастую обладают избыточной сложностью для решения учебных задач и требуют значительного времени на освоение. Предлагаемое приложение ориентировано на образовательный процесс и начинающих специалистов, предлагая интуитивно понятный интерфейс для анализа конкретных типовых схем балочных конструкций. Его новизна заключается в узкой специализации, простоте использования и ориентации на базовые задачи технической механики, что позволяет быстрее интегрировать его в учебный процесс.

ОБЪЕКТ ИССЛЕДОВАНИЯ

Объектом исследования является процесс расчета опорных реакций для статически определимых балочных конструкций при различных типах статического нагружения.

ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

Целью работы является разработка кроссплатформенного программного приложения, автоматизирующего расчет опорных реакций балочных конструкций для образовательных и прикладных задач.

Для достижения поставленной цели были решены следующие задачи:

1. Проанализировать типовые схемы балочных конструкций и видов нагрузок, характерные для курсов технической механики.
2. Разработать алгоритмы расчета опорных реакций для выбранных типовых схем.
3. Спроектировать архитектуру программного обеспечения, включая пользовательский интерфейс, расчетное ядро и модуль валидации данных.
4. Реализовать программное приложение на языке Java, обеспечивающее ввод данных, выполнение расчетов, вывод результатов и обработку ошибок.
5. Проверить корректность работы программы на тестовых примерах.

МЕТОДЫ ИССЛЕДОВАНИЯ

В процессе исследования применялись следующие методы:

- Анализ и синтез – для изучения методик расчета балочных конструкций и формирования требований к программному обеспечению.
- Объектно-ориентированное проектирование – для создания модульной архитектуры приложения.
- Моделирование – с использованием диаграмм UML (прецедентов, компонентов, последовательности) для визуализации функциональности и взаимодействия компонентов системы на этапе проектирования.
- Программирование – реализация приложения на языке Java с использованием его стандартных библиотек для построения графического интерфейса пользователя (Swing) и математических вычислений.

ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРИЛОЖЕНИЯ

Чтобы наглядно отразить, каким образом пользователи взаимодействуют с системой и какие функции для них доступны, применяют специальные средства моделирования. К таким средствам относятся различные виды диаграмм UML, например диаграммы прецедентов, диаграммы компонентов и диаграммы последовательности.

Диаграмма прецедентов представлена на рисунке 1 – это элемент UML (Unified Modeling Language), который используется для визуализации функциональных требований системы с точки зрения взаимодействия пользователей и программного обеспечения.

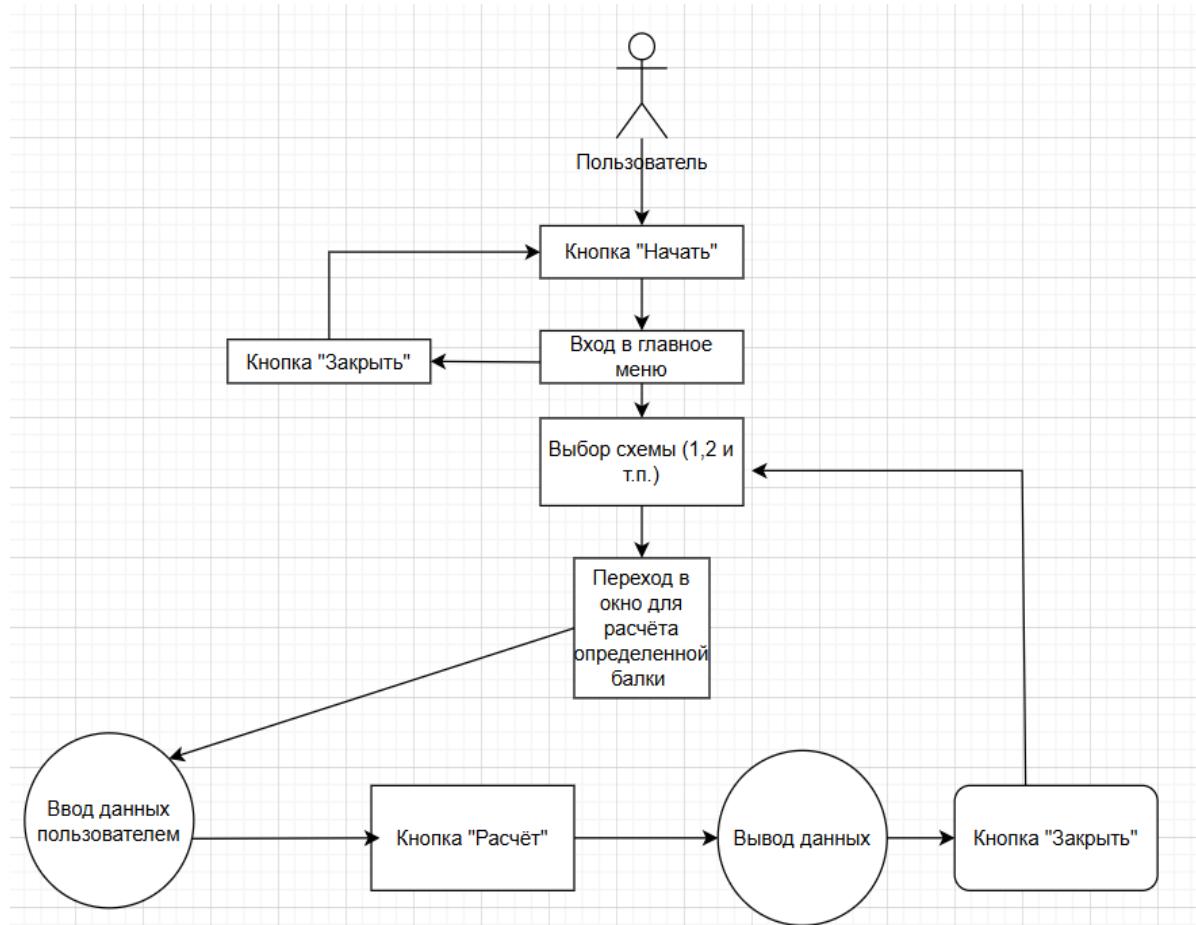


Рисунок 1 – Диаграмма прецедентов

Для представления внутренней модульной структуры применяют диаграмму компонентов. Диаграмма компонентов (component diagram) – это графический инструмент UML, предназначенный для отображения структуры программной системы через компоненты и связи между ними. Компонент в данном контексте представляет собой независимую часть системы, обладающую определенными контрактами и интерфейсами. Взаимосвязи между компонентами описывают зависимости, обмен данными и взаимодействие через интерфейсы, что важно для правильной организации архитектуры и возможности независимой разработки.

Для описания модульной структуры системы была применена диаграмма компонентов (рисунок 2). Она наглядно показывает ключевые компоненты приложения (модуль пользовательского интерфейса, расчетное ядро, модуль валидации данных) и интерфейсы их взаимодействия, что обеспечивает ясность архитектуры и возможность независимой разработки компонентов.

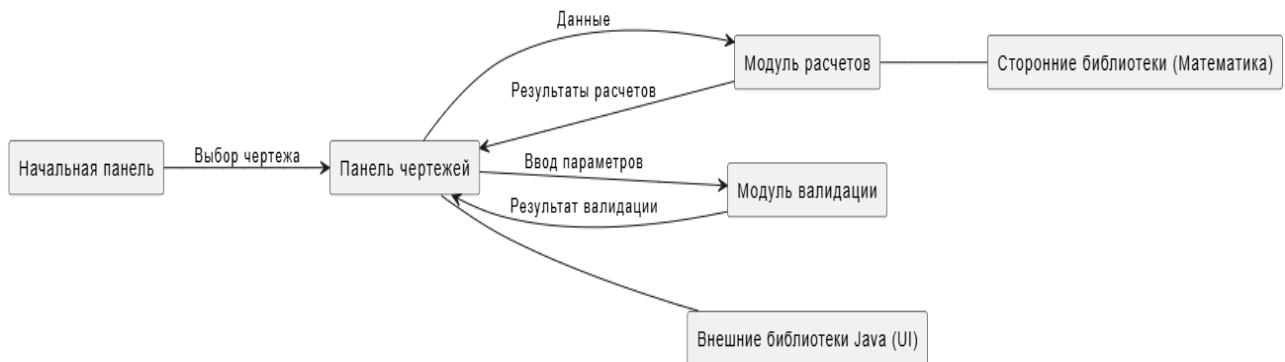


Рисунок 2 – Диаграмма компонентов

Использование диаграмм последовательности играет важную роль в анализе требований и проектировании программных систем. Такие диаграммы способствуют выявлению деталей обмена информацией, протяженности и порядка взаимодействий, выявлению потенциальных узких мест и конфликтов в логике работы системы.

Диаграмма последовательности (рисунок 3) была использована на этапе детального проектирования для моделирования алгоритма выполнения расчета. Она специфицирует временную последовательность взаимодействий между объектами системы (пользователь, интерфейс, расчетный модуль) при выполнении основного прецедента «Произвести расчет».

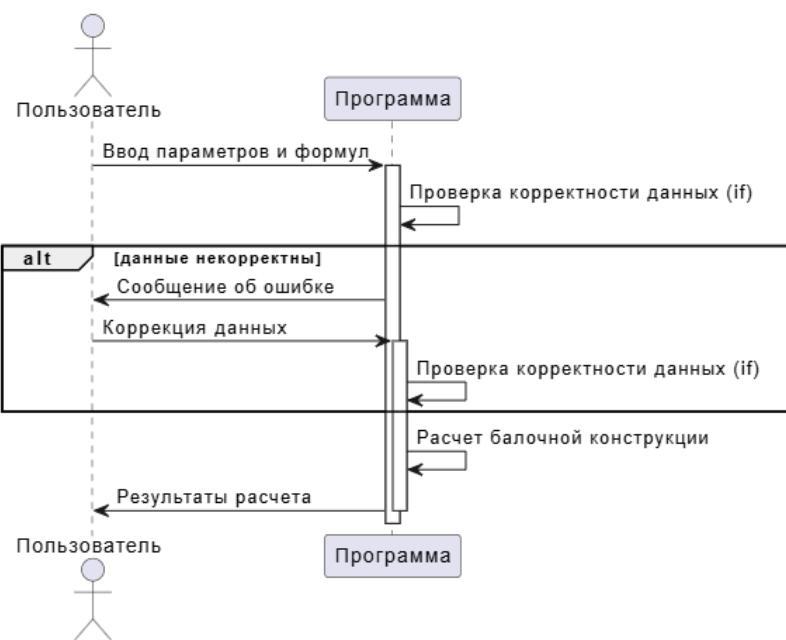


Рисунок 3 – Диаграмма последовательности

Использование указанных диаграмм UML на ранних этапах разработки позволило формализовать требования, проработать логику системы, что впоследствии снизило количество ошибок и упростило процесс кодирования.

СРЕДСТВА РАЗРАБОТКИ

Разработка программного приложения осуществлялась с использованием языка программирования Java, что обусловлено рядом его весомых преимуществ. Одним из ключевых достоинств Java является кроссплатформенность, обеспечивающая функционирование приложений на различных операционных системах, таких как Windows, Linux и macOS, без необходимости перекомпиляции исходного кода. Важным аспектом при выборе данной тех-

нологии также стала надежность: встроенный механизм обработки исключительных ситуаций позволяет своевременно выявлять и корректно обрабатывать возможные ошибки, возникающие в процессах вычислений [1–4].

ОПИСАНИЕ ПРОГРАММНОГО КОДА

На начальном этапе разработки программы проводится идентификация и определение всех переменных, необходимых для дальнейших расчетов. Переменные объявляются с использованием типа двойной точности, что обеспечивает высокую точность вычислений и минимизацию ошибок округления. Для каждой переменной создается отдельное диалоговое окно для ввода данных. Пользователь вводит данные непосредственно в соответствующее окно. После ввода значения программа автоматически извлекает соответствующее число из диалогового окна и учитывает его в последующих вычислениях. Соответствующий фрагмент кода приведен на рисунке 4.

```
public class SHEMA1 extends javax.swing.JFrame {
    public static double a, L1, L2, L3, L4, P, Px, Py, Rax, Ray, Rby;

    /**
     * Creates new form Glavnaya
     */
    public SHEMA1() {
        initComponents();
    }

    public static void getDados (){// получаем данные из окон

        L1=Double.parseDouble(jTextField3.getText());
        L2=Double.parseDouble(jTextField4.getText());
        L3=Double.parseDouble(jTextField5.getText());
        L4=Double.parseDouble(jTextField6.getText());
        P=Double.parseDouble(jTextField1.getText());
        a=Double.parseDouble(jTextField2.getText());
    }
}
```

Рисунок 4 – Фрагмент кода задания и получения значений переменных двойной точности из окон ввода

На втором этапе разработки приложения реализуются расчетные формулы, по которым осуществляется вычисление требуемых параметров. Соответствующий фрагмент листинга программы показан на рисунке 5. Для выполнения данных расчетов используются значения, предварительно введенные пользователем в соответствующие окна интерфейса программы. Это обеспечивает автоматизацию вычислительного процесса и минимизирует вероятность ошибок, связанных с ручным вводом и расчетами.

```
public static void raschet(){//рассчитываем параметры
    Px = (P * Math.cos(Math.toRadians(a)));
    Py = (P * Math.sin(Math.toRadians(a)));
    Rax = Px;
    Rby = (Py * (L2 - L1)) / (L3-L1);
    Ray = Py - Rby;
    DecimalFormat decimalFormat = new DecimalFormat( "#.#####" );
```

Рисунок 5 – Фрагмент программного кода, по которым программа выполняет расчеты

На третьем этапе осуществляются проверки на выполнение всех необходимых условий при активации кнопки «Расчет». Для переменных, значения которых задаются пользователем, устанавливаются соответствующие ограничения. Для реализации данной проверки применяется оператор «if», обеспечивающий формулирование условий и контроль их выполнения в процессе расчета. Соответствующий фрагмент кода представлен на рисунке 6.

```
// Кнопка расчет
getDados();
if(L1 < 0) {
    JOptionPane.showMessageDialog(null,"Расстояние L1 не может быть меньшим нуля!","Ошибка ввода данных!", JOptionPane.ERROR_MESSAGE);
    return;
}

if(L2 < 0) {
    JOptionPane.showMessageDialog(null,"Расстояние L2 не может быть меньшим нуля!","Ошибка ввода данных!", JOptionPane.ERROR_MESSAGE);
    return;
}
```

Рисунок 6 – Пример программной проверки корректности пользовательского ввода с использованием условного оператора «if» и вывода сообщения об ошибке через JOptionPane

ОПИСАНИЕ РАБОТЫ ПРОГРАММЫ

При запуске программы пользователь встречает заставку, выполняющую роль приветственного интерфейса, как показано на рисунке 7. На данной форме размещена интерактивная кнопка «Начать», по нажатию на которую осуществляется переход к основной части приложения.

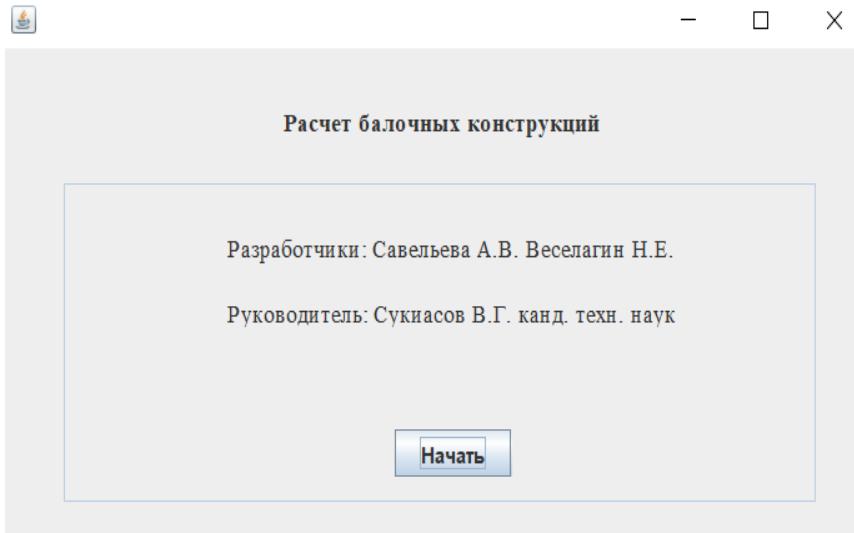


Рисунок 7 – Заставка

Главная форма. После нажатия кнопки «Начать» открывается Главная форма, где представлены десять различных схем балочных конструкций со схемами нагружений, которые может обработать программа. Пользователь имеет возможность выбрать необходимую ему балку, схему нагружения и вид нагрузки посредством нажатия на соответствующий чертеж, в результате чего осуществляется переход к следующей форме, предназначеннной для ввода данных и выполнения расчетов реакций в опорах выбранной балки.

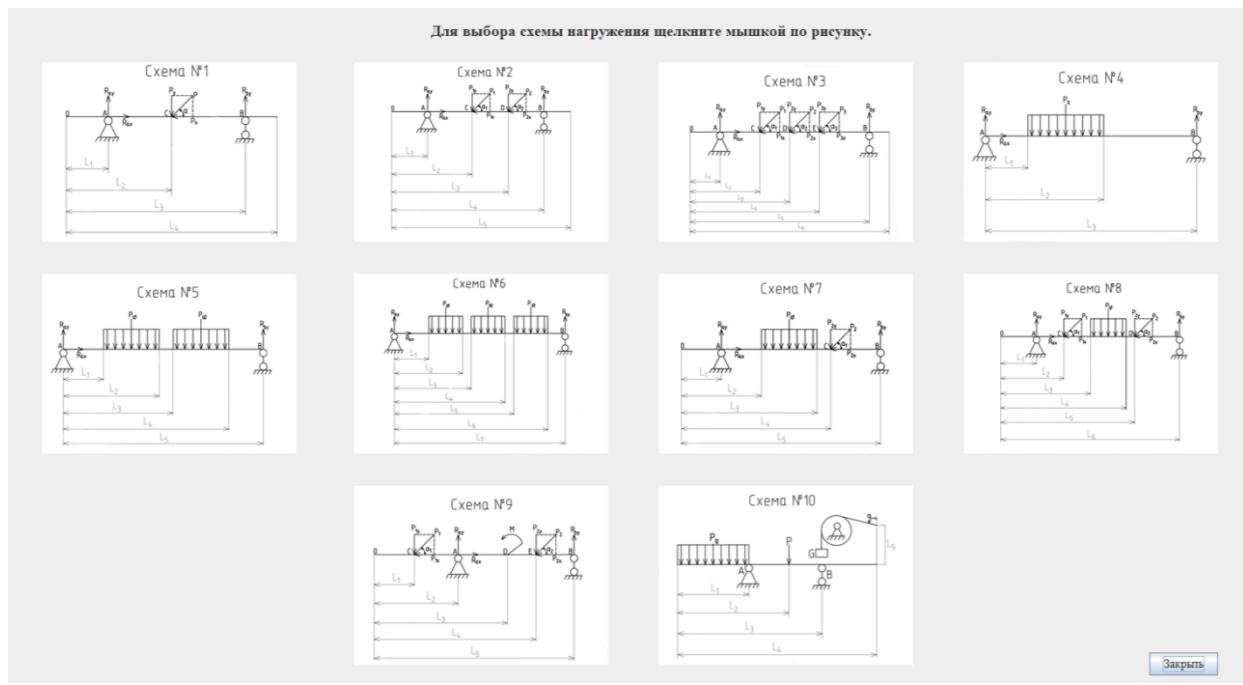


Рисунок 8 – Главная форма

При выборе пользователем требуемого чертежа балочной конструкции осуществляется автоматический переход в следующую форму, предназначенную для ввода данных, расчета и вывода данных расчета реакций связей. При этом расчет реакций производится по вертикальным и горизонтальным осям.

Алгоритм работы расчетной подсистемы предусматривает автоматизированное определение реакций в опорных точках конструкции. После внесения исходной информации программа осуществляет вычисление реакций в точках А (Ra) и В (Rb), по осям (x, y), а также определяет величину силы, эквивалентной распределенной нагрузке (Pq). Введенные пользователем и подсчитанные значения отображаются в соответствующих визуальных компонентах на форме, как показано на рисунок 9.

Схема нагружения №7. Расчет реакций связей.

Схема №7

Введите данные

Длина (L1), м	1
Длина (L2), м	2
Длина (L3), м	4
Длина (L4), м	8
Длина (L5), м	10
Распределенная нагрузка (q1), кН/м	5
Величина силы (P2), кН	5
Угол наклона силы (a1), градусов	5

Результаты расчета

Сила эквивалентная q1 (Pq1), кН	10
Реакция в точке А (Ray), кН	8,9857
Реакция в точке А (Rax), кН	4,981
Реакция в точке В (Rby), кН	1,4501

Закрыть

Рисунок 9 – Форма расчета

В процессе взаимодействия пользователя с программой предусмотрено введение произвольных числовых параметров, например таких как величины сил (кН), интенсивность распределенной нагрузки (кН/м), расстояние до точек приложения сил (м), углы наклона сил от 0 до 360 ($^{\circ}$), величины изгибающих моментов (Нм) и т. д., определяющих исходные условия задачи. Для обеспечения корректности расчетов и стабильно высокой надежности программного продукта на вводимые величины наложены определенные логические ограничения [см. рисунок 10]. Данные меры направлены на предотвращение возможных ошибок со стороны пользователя, возникающих вследствие ввода недопустимых или некорректных значений [5–7].

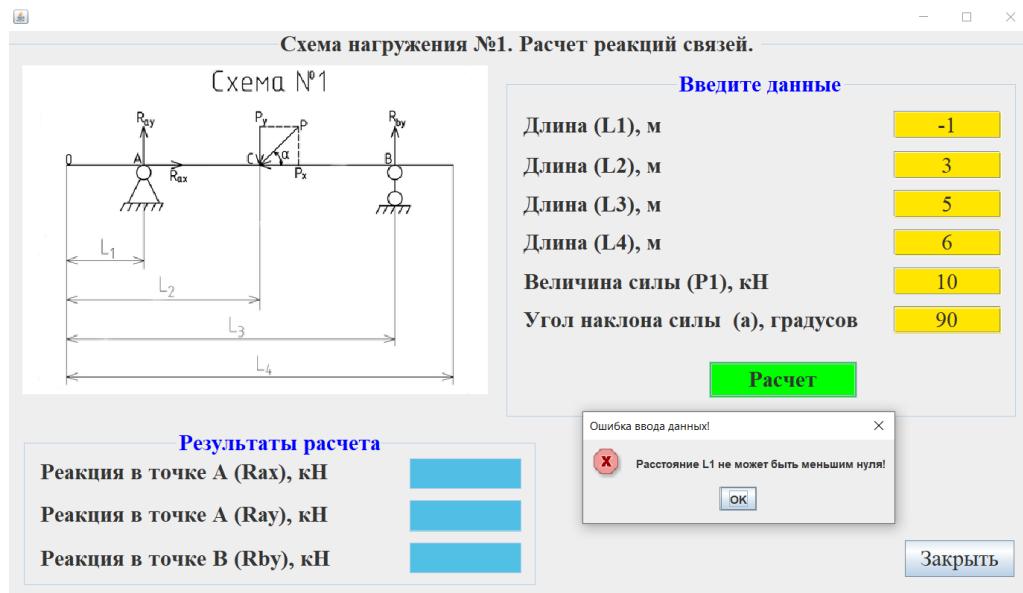


Рисунок 10 – Ограничения

В случае ввода некорректных данных приложение сообщает об ошибке посредством всплывающей формы, как показано на рисунке 10.

ЗАКЛЮЧЕНИЕ

Разработанное программное средство обеспечивает эффективное выполнение инженерных расчетов реакций в опорах балочной конструкций, предоставляя пользователю возможность ввода данных разного рода величин и возможность исследования реакций опор при различных схемах нагружений и различных нагрузках, производить анализ множественных сценариев при разных схемах нагружений. Кроссплатформенность на базе Java делает программу доступной и удобной для широкого круга пользователей.

СПИСОК ЛИТЕРАТУРЫ

1. Шилдт, Г. Java. Полное руководство. Классическая книга для новичков и продвинутых, охватывает все аспекты Java. – Изд.: Компьютерное издательство «Диалектика». – 1345 с.
2. Яблонский, А. А. Курс теоретической механики / А. А. Яблонский, В. М. Никифорова. – Ч. 1. Статика. Кинематика. Учебник для вузов. – Изд. 5-е, испр. – Москва: Высшая школа», 1977. – 368 с.
3. Маркелов, К. Д. Разработка приложения-организатора для студента / К. Д. Маркелов // Молодой ученый. – 2021. – № 18 (360). – С. 35–39.
4. Левушкин, А. В. Основные современные языки программирования / А. В. Левушкин, М. К. Турчанинов, А. А. Жиганов, В. В. Ермолаева // Молодой ученый. – 2018. – № 25 (211). – С. 96–98.

5. Попов, Д. В. Автоматизированное тестирование: выбор языка программирования / Д. В. Попов // Молодой ученый. – 2025. – № 18 (569). – С. 27–29.
6. Vestdam, Th. Elucidative Programming in Open Integrated Development Environments for Java / Department of Computer Science, Aalborg University Fredrik Bajers Vej 7E 9220 Aalborg, Denmark. – 2003. – P. 6.
7. Sampada K. Satav. A Comparative Study and Critical Analysis of Various Integrated Development Environments of C, C++, and Java Languages for Optimum Development* / Sampada K. Satav, S. K. Satpathy, K. J. Satao [et al.] // UNIASCIT. – 2011. – Vol. 1 (1). – 9–15.

DEVELOPMENT OF A SOFTWARE APPLICATION FOR CALCULATING GIRDER STRUCTURES

A. V. Savelieva, student
 E-mail: savelieva_nastia@mail.ru
 Kaliningrad State Technical University

N. E. Veselagin, student
 E-mail: veselaginn@mail.ru
 Kaliningrad State Technical University

This development is aimed at automating calculations in the design of beam structures, which significantly improves the accuracy and speed of engineering tasks and minimizes the risk of errors related to the human factor. Implementation of such an application optimizes the expenditure of time and resources, makes it easier for specialists to quickly analyze and compare various engineering solutions. In addition, the development has a certain educational value: its application in educational institutions will effectively train future specialists in technical mechanics, will contribute to a deeper understanding of calculation techniques and improve students' practical skills.

Keywords: automation of calculations, design of beam structures, improvement of accuracy, speed of execution, minimization of errors, human factor, optimization of time, optimization of resources, operational analysis, comparison of engineering solutions, educational value, application in educational institutions, training of specialists, technical mechanics, understanding of calculation techniques, improvement of practical skills, students.