

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Факультет математики, механики и компьютерных технологий
Кафедра прикладной математики и программирования
Направление подготовки прикладная математика и информатика

РАБОТА ПРОВЕРЕНА

Рецензент,

« ____ » _____ 2018г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
доцент

_____ А.А.Замышляева
« ____ » _____ 2018 г.

Разработка обучающей программы по линейной алгебре

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ–01.03.02.2018.41.ПЗ ВКР

Руководитель работы, старший
преподаватель

_____ /М.Ю. Сартасова
« ____ » _____ 2018 г.

Автор работы

студентка группы ЕТ-412

_____ /Л.Ш. Арсанбекова
« ____ » _____ 2018 г.

Нормоконтролер, кандидат
экономических наук, доцент

_____ /Д.А. Дрозин
« ____ » _____ 2018 г.

Челябинск,

2018

АННОТАЦИЯ

Арсанбекова Л.Ш. Разработка обучающей программы по линейной алгебре. – Челябинск: ЮУрГУ, ЕТ-412, 53 с., 24 ил., 2 табл., библиогр. список – 20 наим., 1 прил..

Выпускная квалификационная работа выполнена с целью разработки обучающего электронного пособия по линейной алгебре, позволяющего проводить занятия со студентами высших учебных заведений.

Исходя из поставленной цели, в дипломной работе приведена классификация компьютерных обучающих средств, рассмотрены существующие обучающие приложения, приведены их достоинства и недостатки, реализовано обучающее программное обеспечение по курсу линейная алгебра.

Результаты дипломной работы – разработано программное обеспечение, позволяющее организовать самостоятельную работу студентов по курсу линейной алгебры, тем самым способствуя повышению эффективности формирования профессиональных компетенций будущих бакалавров.

ОГЛАВЛЕНИЕ

| | |
|--|----|
| Введение..... | 9 |
| 1 Методы и способы решения задачи создания электронного учебного пособия | 10 |
| 1.1 Этапы создания электронного учебного пособия..... | 10 |
| 1.2 Преимущества и недостатки компьютерных систем обучения | 10 |
| 1.3 Классификация электронных учебных пособий..... | 11 |
| 1.3.1 Электронные учебники | 12 |
| 1.3.2 Системы компьютерной алгебры..... | 15 |
| 1.3.3 Контролирующие программы | 17 |
| 1.4 Обзор существующих электронных обучающих программ..... | 22 |
| 1.4.1 Интерактивный электронный учебник Алгебра-1 | 22 |
| 1.4.2 Контролирующая программа онлайн | 25 |
| 1.5 Требования к содержанию учебного пособия по линейной алгебре . | 28 |
| 1.5.1 Основные понятия | 28 |
| 1.5.2 Операция сложения матриц..... | 28 |
| 1.5.3 Операция умножения матрицы на число | 29 |
| 1.5.4 Операция умножения двух матриц | 29 |
| 1.5.5 Операция сложения векторов..... | 30 |
| 1.5.6 Операция умножения вектора на число | 30 |
| 1.5.7 Операция умножения двух векторов | 31 |
| 1.5.8 Операция умножения вектора на матрицу..... | 31 |
| 1.6 Выбор языка программирования и среды разработки | 32 |
| 1.7 Постановка задачи..... | 32 |
| 2 Математическая модель в линейной алгебре..... | 34 |
| 2.1 Операция сложения матриц | 34 |
| 2.2 Операция умножения матрицы на число..... | 34 |
| 2.3 Операция умножения двух матриц | 34 |
| 2.4 Операции с векторами | 34 |
| 3 Разработка обучающей программы..... | 35 |
| 3.1 Описание алгоритма работы программы..... | 35 |
| 3.1.1 Вспомогательный алгоритм обработки сообщений..... | 35 |
| 3.1.2 Вспомогательный алгоритм вывода содержания..... | 37 |

| | |
|---|----|
| 3.2 Разработка пользовательского интерфейса | 40 |
| Заключение | 45 |
| Литература | 46 |
| Приложение | 48 |

ВВЕДЕНИЕ

Математика является значительной частью общечеловеческой культуры и профессиональности. Это определяет необходимость её изучения при получении высшего образования. Математика, прежде всего, развивает абстрактное мышление студентов, включающее логическое (дедуктивное) мышление, алгоритмическое мышление. Она воспитывает аналитический и критический склад ума, необходимый любому профессионалу.

Одним из важнейших разделов высшей математики, изучаемых в вузах, является линейная алгебра. Линейные модели часто используются при моделировании различных процессов. Для того, чтобы будущие бакалавры смогли применять в своей деятельности подобные модели, они должны для начала освоить линейную алгебру.

Главное в обучении математике и математическим методам студентов вуза – научить их учиться, выработать у них глубокую потребность в математических знаниях, стремление к совершенствованию и обновлению знаний, умение применять их в практической деятельности. Одно из условий эффективности учебного процесса – наличие интереса к изучаемому предмету. Одним из способов повышения интереса у студента к обучению является применение программных средств.

Обучающая программа – это полное описание процесса обучения, содержащее учебный материал, практические задания, необходимые для его усвоения, и указания по их выполнению и контролю.

Данная работа посвящена разработке компьютерной обучающей программы по линейной алгебре и включает в себя следующие этапы:

1. анализ предметной области;
2. проведение обзора существующих обучающих тренажеров по линейной алгебре;
3. разработка математической модели;
4. разработка архитектуры приложения;
5. реализация обучающей программы;
6. тестирование и анализ полученных результатов

1 МЕТОДЫ И СПОСОБЫ РЕШЕНИЯ ЗАДАЧИ СОЗДАНИЯ ЭЛЕКТРОННОГО УЧЕБНОГО ПОСОБИЯ

1.1 Этапы создания электронного учебного пособия

При разработке электронного учебного пособия можно выделить следующие основные этапы [21]:

1) идентификация проблемы: определяется выбор курса, аудитория, определяется состав рабочей группы, и назначаются роли создателей обучающего пособия, определяются характеристики и цели используемых ресурсов, рассчитываются затраты и время, необходимые для создания обучающей системы;

2) концептуализация: определяются содержание, цели и задачи изучения учебной дисциплины, происходит тщательный отбор источников информации, выделение наиболее важной информации, сбор полученных данных в единое целое, при этом важно учитывать последовательность и полноту изложения материала;

3) формализация. На этом этапе разработчики определяют, как лучше преподнести информацию, изучают принципы оценивания и обратной связи, а затем строят алгоритмы, по которым будет проходить взаимодействие обучаемых с электронным учебником. Обучение без обратной связи предназначено для подачи материала по определенному сценарию. Интерактивное обучение включает в себя учебный процесс, основанный на взаимодействии с обучаемым.

4) реализация – создание медиаэлементов, реализация замысла разработчиков в виде программного продукта, особенности которого определяются выбранными для его реализации информационными технологиями;

5) тестирование и отладка – на этом этапе проверяют, как функционирует приложение, используя те же условия, при которых с ним будет работать пользователь, проводится серия тестов с целью выявить ошибки программирования, таких как: «мертвые» ссылки, отсутствие доступа к разделам, неправильная работа системы тестов и др;

6) внедрение и эксплуатация: происходит внедрение полностью законченной компьютерной системы обучения в образовательные учреждения и глобальную сеть Интернет. Разрабатывается план занятий с использованием этой системы и начинается ее эксплуатация.

1.2 Преимущества и недостатки компьютерных систем обучения

Очень важно привлечь интерес студента, если не к математической науке в целом, то хотя бы к практическим и лекционным занятиям, т.к. интерес к учебному процессу приводит к тому, что зачастую студенты, которые имеют недостаточные базовые знания, в результате имеют лучшие результаты на экзаменах и зачетах. В этой ситуации применение программных средств в

обучении математике может стать одним из факторов повышения интереса студентов к занятиям по данной дисциплине.

Педагогические возможности компьютерных средств обучения намного превосходят возможности традиционных средств обучения. Следует отметить положительные стороны в использовании компьютерных технологий в образовательном процессе:

- повышение доступности и качества образования;
- расширение наборов применяемых учебных задач, моделирующих реальные ситуации;
- возможность увидеть, к какому результату приводит решение;
- средства мультимедиа расширяют возможности представления информации. Наглядность в электронных обучающих системах выше, чем в печатных изданиях;
- компьютер даёт возможность строить индивидуальную траекторию обучения на основе базовых знаний обучающегося, его способностей;
- повышение мотивации студента к освоению материала.

Также есть и недостатки:

- вредное воздействие на глаза человека при длительном использовании компьютера;
- отсутствует возможность уточнения возникших вопросов в процессе обучения у преподавателя.

1.3 Классификация электронных учебных пособий

Существует большое количество классификаций компьютерных учебных программных средств. В соответствии с одной из них программные средства могут быть классифицированы следующим образом [11]:

- компьютерные тренажеры;
- компьютерные обучающие системы;
- контролирующие программы;
- электронные учебники;
- информационно-поисковые программы;
- лабораторные практикумы;
- инструментальные системы;
- имитационные программные средства.

При обучении алгебры студентов в вузах применяются следующие основные программные средства:

- электронные учебники;
- системы компьютерной алгебры;
- контролирующие программы.

Рассмотрим подробно эти средства.

1.3.1 Электронные учебники

Электронный учебник – это специальное устройство или программное обеспечение, используемое в образовательном процессе и заменяющее собой традиционный бумажный учебник. В настоящее время существуют различные подходы к определению электронного учебника. В некоторых случаях под электронным учебником понимается электронная версия бумажного учебника, в других – сложный комплекс программ на электронных устройствах, позволяющий демонстрировать обучающимся помимо текста мультимедийный материал учебного характера, а также содержащий в себе интерактивные блоки проверки знаний [12].

1.3.1.1 Преимущества электронного учебника

Электронный учебник имеет ряд преимуществ по сравнению с печатным аналогом: простота и удобство обращения, использование дополнительных возможностей (использование мультимедийного материала, гиперссылок и т.п.), возможность обновления информационного материала, расширение восприятия информации, увеличение скорости предоставления образовательной услуги, полнота передаваемой информации, компактность хранения огромных массивов информации.

Гипертекст в электронном учебнике создает «живой», интерактивный учебный материал, снабженный ссылками, позволяющим перемещаться не только между частями учебника, но переходить на внешние источники информации.

Электронный учебник может содержать несколько уровней сложности, что делает учебный процесс личностно-ориентированным, в то время как печатный учебник рассчитан только на один уровень сложности.

Учебное занятие может быть эффективнее благодаря интерактивности электронных учебников. Они повышают наглядность информации, студенты лучше воспринимают информацию.

Электронный учебник позволяет самостоятельно освоить учебный курс или его раздел. Компьютерная помощь в виде подсказок оптимизирует учебный процесс, приводит к более интенсивному участию студентов в процессе обучения.

Возможностью электронных учебников также является контроль выполнения заданий, что обеспечивает самостоятельность студентов и разгрузку деятельности преподавателя.

В настоящее время в учебном процессе используются электронные учебники двух видов:

К первому относятся электронные копии бумажных учебников, но, возможно, с небольшим количеством возможностей: увеличение рисунков, вставка гиперссылок. В этом случае разница между бумажным и электронным учебниками в основном заключается лишь в том, что студент читает текст с экрана компьютера.

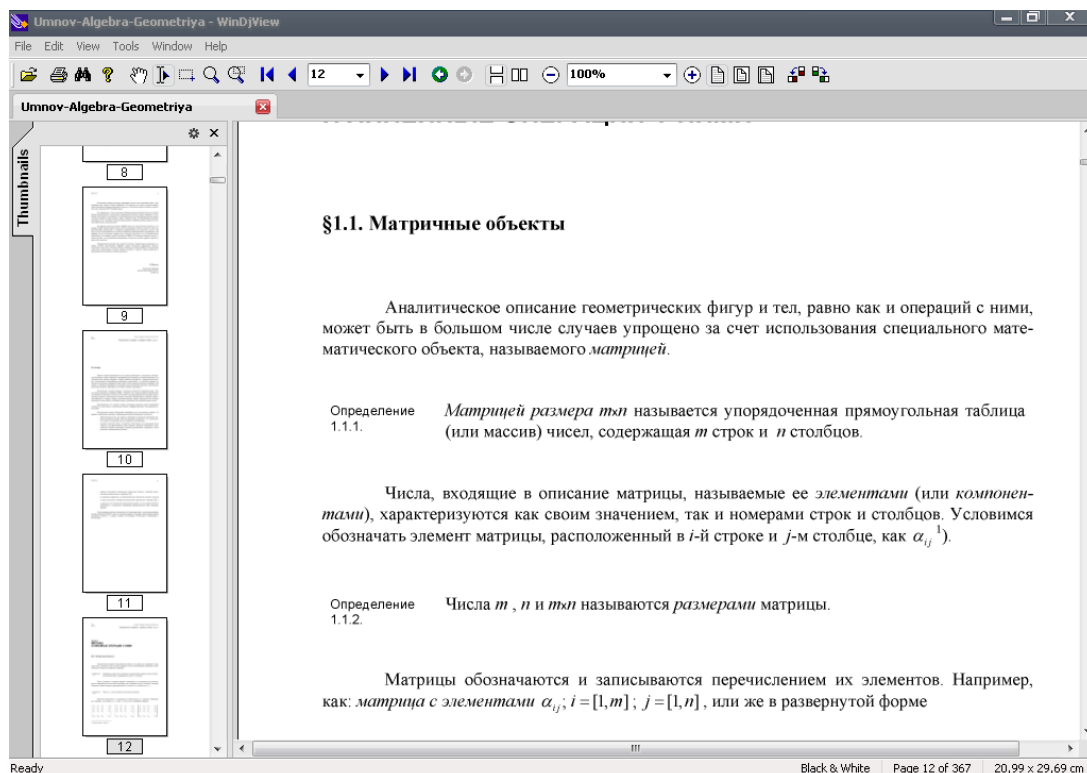


Рисунок 1.1 – Пример электронного учебника 1 типа

Второй тип электронных учебников имеет сложную структуру и может включать в себя:

- Гипертекст, позволяющий совершать внутритекстовые и межтекстовые переходы, предоставляющий читателю возможность воздействия на последовательность композиционных единиц. В гипермедиа-системе в качестве фрагментов могут использоваться изображения, а информация может содержать текст, графику, видеофрагменты, звук.
- Обучающие видеоматериалы, предназначены для более эффективного и наглядного представления обучающего материала.
- Интерактивные словари-справочники, содержащие поиск, с помощью которого можно быстро найти нужную информацию; наличие гиперссылок, позволяющих получить разъяснение терминов, содержащихся в тексте, чтобы получить развернутую информацию по определенной теме; получение регулярно обновляющейся информации.
- Интерактивные тренировочные и контрольные тесты, позволяющие проверить усвоенный материал после каждого параграфа или главы.
- Статистика успеваемости - позволяет наблюдать процент освоения изученного материала.

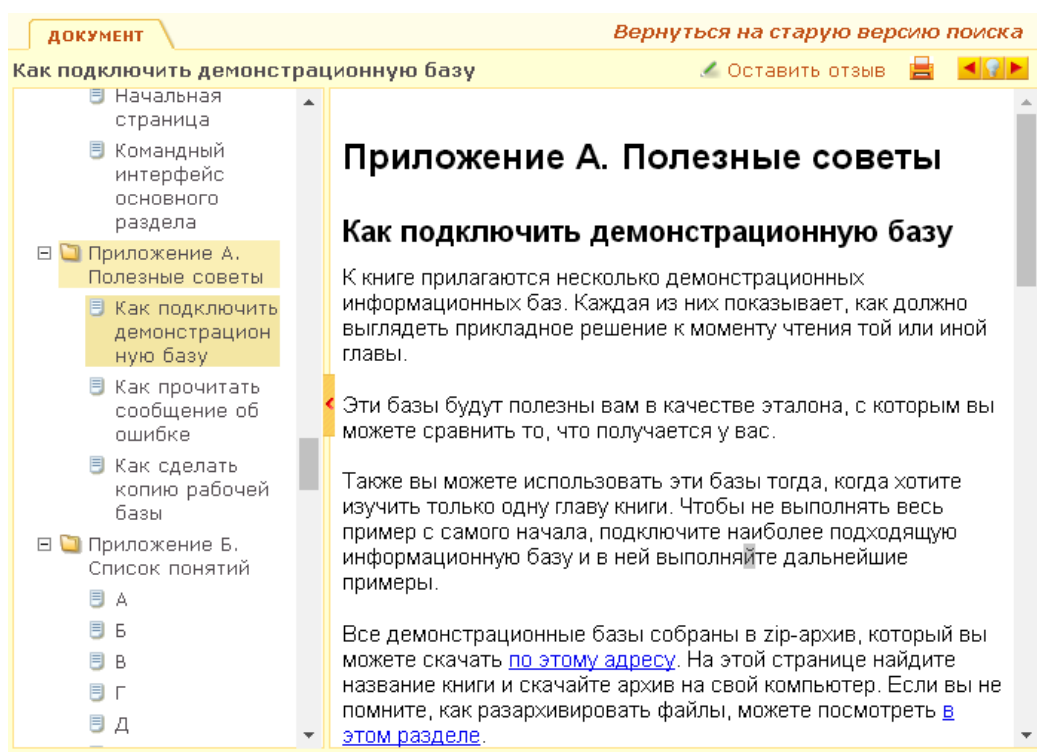


Рисунок 1.2 – Пример электронного учебника 2 типа (текст с гиперссылками) [17]

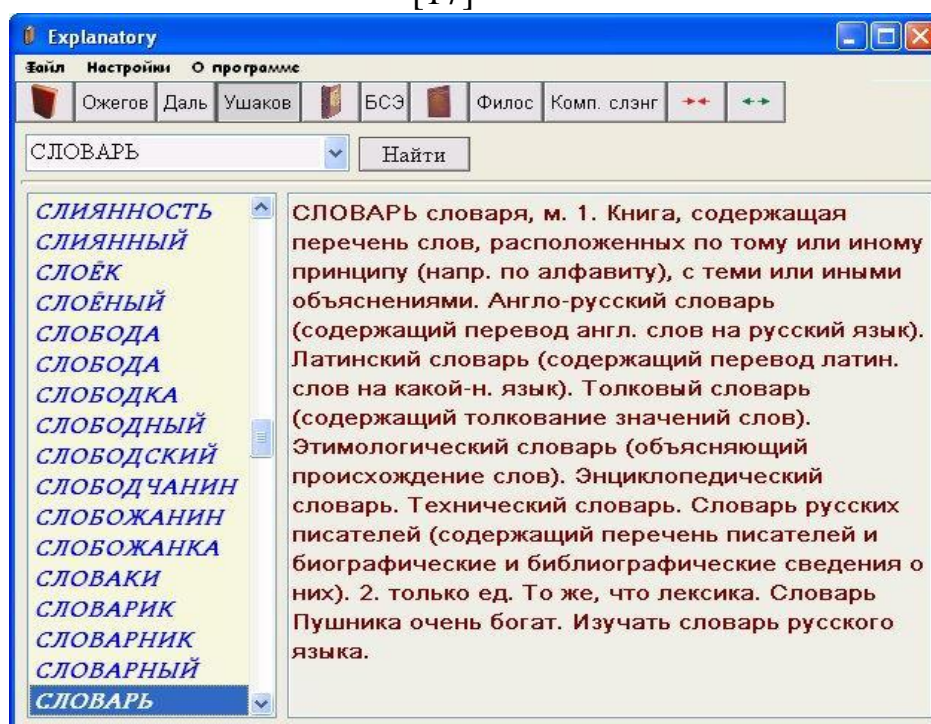


Рисунок 1.3 – Пример электронного учебника 2 типа (словарь) [16]

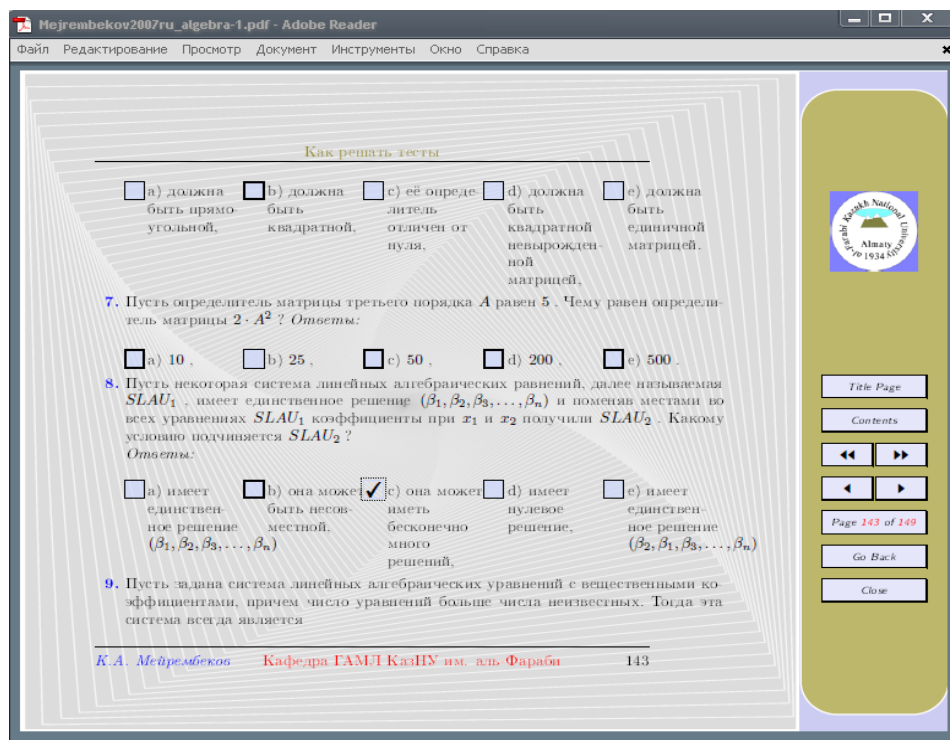


Рисунок 1.4 – Пример электронного учебника 2 типа (тесты) [15]

1.3.2 Системы компьютерной алгебры

Первые ЭВМ изначально создавались для того, чтобы проводить сложные расчёты, на которые человек тратил очень много времени. Следующим шагом развития ЭВМ стали ПК. Эти машины могут проводить вычисления разной сложности (от самых простых до самых сложных). Такая их особенность использовалась в разных областях знаний. Развитие компьютерных математических систем привело к появлению отдельного класса программ, который получил названия Системы Компьютерной Алгебры.

Система компьютерной алгебры (СКА, англ. computer algebra system, CAS) — это прикладная программа для символьных вычислений, то есть выполнения преобразований и работы с математическими выражениями в аналитической (символьной) форме.

К особенностям СКА относят преимущественно интерактивный характер работы – пользователь не знает заранее ни размера, ни формы результатов и поэтому должен иметь возможность корректировать ход вычислений на всех этапах, задавать режим пошагового выполнения с выводом промежуточных результатов.

Системы компьютерной алгебры можно разделить на две группы согласно их развитию [14]. Системы первой группы можно рассматривать как программы специального назначения, т.е. они разработаны для решения конкретных задач в различных областях. К этой группе относятся программы, такие как GAP (теория групп), Calabra (Тензорная алгебра), KANT (алгебра и теория чисел), Singular (полиномиальные выражения с акцентом на нужды коммутативной

алгебры, алгебраическая геометрия), Calc3D (для работы с 3D матрицами, векторами, комплексными числами), GRTensorII (дифференциальная геометрия).

GAP (Groups, Algorithms, Programming) – свободно распространяемая кроссплатформенная система компьютерной алгебры для вычислительной дискретной алгебры с особым вниманием к вычислительной теории групп [13].

Основные особенности системы [10]:

- свободный доступ;
- специальный язык GAP подобный Pascal, но являющийся объектно-ориентированным языком;
- стандартные типы основных алгебраических объектов: групп (подстановок, абстрактных, матричных), колец, полей;
- более 4000 функций библиотек, реализующих алгебраические алгоритмы, написанных на языке GAP;
- обширная документация, рассказывающая о функциональности и использовании GAP;
- подробное и удобное описание (около 1600 стр.) в формате «гипертекст»;
- возможность расширения системы GAP, используя внешние пакеты и библиотеки.

KANT (Computational Algebraic Number Theory) - система компьютерной алгебры для сложных вычислений в алгебраических числах, разработанная под руководством профессора д-ра М. Е. Похста в Техническом университете Берлина [18].

Системы из второй группы являются программами общего назначения, снабжающими пользователя как можно более широкими математическими возможностями. Представителями этой группы являются Maple, MATLAB, Mathematica, Maxima, Axiom, SMathStudio.

Maple - это математическое программное обеспечение, которое сочетает в себе самый мощный математический движок в мире с интерфейсом, который значительно облегчает анализ, исследование, визуализацию и разрешение математических задач. Система объединяет в себе следующие составляющие [14]:

- редактор для подготовки и изменения документов и программных модулей;
- ядро алгоритмов и правил преобразования математических выражений;
- численный и символьный процессоры;
- язык программирования;
- многооконный пользовательский интерфейс с возможностью работы в диалоговом режиме;
- библиотеки встроенных и дополнительных функций;
- пакеты функций сторонних производителей;
- справочную систему с пояснениями всех функций и опций;
- систему диагностики;
- поддержку нескольких других языков программирования.

СКА WolframMathematica - это современная техническая вычислительная система, широко используемая в научных, инженерных, математических и компьютерных областях [19].

Основные аналитические возможности:

- решение систем полиномиальных и тригонометрических уравнений и неравенств, а также трансцендентных уравнений, сводящихся к ним;
- решение рекуррентных уравнений;
- упрощение выражений;
- нахождение пределов;
- интегрирование и дифференцирование функций;
- нахождение конечных и бесконечных сумм и произведений;
- решение дифференциальных уравнений и уравнений в частных производных;
- преобразования Фурье и Лапласа, а также Z-преобразование;
- преобразование функции в ряд Тейлора, операции с рядами Тейлора: сложение, умножение, композиция, получение обратной функции;
- вейвлетный анализ.

Система также осуществляет численные расчёты: определяет значения функций (в том числе специальных) с произвольной точностью, осуществляет полиномиальную интерполяцию функции от произвольного числа аргументов по набору известных значений, рассчитывает вероятности.

Теоретико-числовые возможности — определение простого числа по его порядковому номеру, определение количества простых чисел, не превосходящих данное; дискретное преобразование Фурье; разложение числа на простые множители, нахождение НОД и НОК.

Также в систему заложены линейно-алгебраические возможности — работа с матрицами (сложение, умножение, нахождение обратной матрицы, умножение на вектор, вычисление экспоненты, взятие определителя), поиск собственных значений и собственных векторов.

Система результаты представляет как в алфавитно-цифровой форме, так и в виде графиков. В частности, реализовано построение графиков функций, в том числе параметрических кривых и поверхностей; построение геометрических фигур (ломаных, кругов, прямоугольников и других); построение и манипулирование графами. Кроме того, реализовано воспроизведение звука, график которого задаётся аналитической функцией или набором точек.

1.3.3 Контролирующие программы

Программы этого типа предназначены для проверки знаний обучаемого. Отличие контролирующих программ от тренажеров заключается в том, что в первых программа сообщает только, верен или неверен выбор варианта ответа, в то время как в тестирующих могут выводиться комментарии к неверным ответам. От обучающей контролирующая часть программы отличается значительно меньшим объемом предоставляемой информации, ограниченностью выполнения

задания во времени и почти или полным отсутствием вспомогательных обучающих воздействий. В цепочке "теория - практика - контроль" этот класс программ реализован в третьем звене [20].

Отметим ряд преимуществ компьютерного тестирования:

- Экономятся силы тестирующего - ему не приходится заниматься весьма утомительной рутинной работой (инструктаж тестируемого, выдача заданий, ведение протокола, подсчет и обработка результатов);
- При наличии хорошо отлаженной программы компьютерное тестирование практически исключает ошибки при обработке результатов - машина всегда использует один и тот же алгоритм, она не отвлекается и не утомляется;
- Оценивание результатов тестирования осуществляется мгновенно, автоматически фиксируется и сохраняется на длительное время;
- Существует возможность формирования тестов, различных по уровню обученности испытуемых;
- Отсутствует необходимость в бумажных носителях. Это важно не с точки зрения экономии средств (бумага, множительная техника, подготовленный персонал, технические средства сканирования листов ответа, программное обеспечение для сканирования и распознавания результатов, занесение результатов для дальнейшей обработки и хранения), а скорее – с точки зрения обеспечения секретности (надежности, безопасности), так как подготовка бумажных вариантов теста требует достаточно большого времени и доступа определённого количества обслуживающего персонала, что может служить источником утечки информации до начала процесса тестирования;
- обеспечение большей степени секретности (отсутствие бумаг, которые можно скопировать), более того, в компьютере содержатся не сами тесты, а банки заданий, из которых формируются варианты;
- есть возможность автоматической фиксации времени выполнения каждого задания;
- Появляется возможность формирования достаточно большого количества вариантов теста, которое ограничено лишь размером банка тестовых заданий;
- Отпадает необходимость в синхронизации процесса тестирования для групп испытуемых. Каждый тестируемый выбирает самостоятельный темп работы с тестом.

На рисунках 1.5-1.9 представлен пример работы контролирующей программы по линейной алгебре [21].



1. ЛИНЕЙНАЯ АЛГЕБРА

1.7. Матрицы. Операции над матрицами. Элементарные преобразования строк матрицы. Приведение матрицы к ступенчатому виду и виду гаусса. Ранг матрицы

Тестовые задания для самоконтроля

1. Если матрица $A = \begin{pmatrix} 2 & -1 \\ 3 & -2 \end{pmatrix}$, то матрица $4A$ имеет вид

| | |
|--|---|
| <input type="checkbox"/> $\begin{pmatrix} 2 & -4 \\ 3 & -8 \end{pmatrix}$ | <input type="checkbox"/> $\begin{pmatrix} 8 & -4 \\ 3 & -2 \end{pmatrix}$ |
| <input type="checkbox"/> $\begin{pmatrix} 8 & -4 \\ 12 & -8 \end{pmatrix}$ | <input checked="" type="checkbox"/> $\begin{pmatrix} 2 & -1 \\ 12 & -8 \end{pmatrix}$ |
| <input type="checkbox"/> $\begin{pmatrix} 8 & -1 \\ 12 & -2 \end{pmatrix}$ | |

Рисунок 1.5 – Пример контролирующей программы [21]

Некоторые вопросы требуют выбора нескольких вариантов ответа (рисунок 1.6).

$$4. \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & -2 & -3 & -4 & -5 \end{pmatrix}$$

Введите последовательность номеров
без разделительных знаков:

правильный ответ

5. Для матриц $A = \begin{pmatrix} 1 & 2 \\ -4 & 2 \\ 0 & 1 \end{pmatrix}$ и $B = \begin{pmatrix} 2 & -1 \\ 3 & 4 \end{pmatrix}$ указать те операции, которые можно выполнить

- $B \cdot A$
- $B \cdot A^T$
- $B^T \cdot A$
- $B^T \cdot A^T$
- $A \cdot B$
- $A^T \cdot B$
- $A \cdot B^T$
- $A^T \cdot B^T$
- все указанные операции можно выполнить

6. Ранг матрицы A размера $n \times n$ равен

Рисунок 1.6 – Пример контролирующей программы [21]

Также тест содержит вопросы, где пользователь должен ввести число (рисунок 1.7).

11. Выбрать верные утверждения. Ранг матрицы равен...

- числу ненулевых строк в ступенчатом виде матрицы;
- числу столбцов матрицы;
- произведению числа строк на число столбцов матрицы;
- максимальному числу линейно независимых строк (столбцов) матрицы;
- числу строк матрицы.

12. Для матриц $A = \begin{pmatrix} 1 & -2 & 2 \\ 3 & 0 & -2 \\ 0 & 2 & -2 \end{pmatrix}$ и $B = \begin{pmatrix} 1 & -5 & 3 \\ 0 & -2 & 1 \\ 2 & -3 & 0 \end{pmatrix}$ найти элемент c_{23} произведения $C = B \cdot A$.

Введите ответ
целым числом:


правильный ответ

Рисунок 1.7 – Пример контролирующей программы [21]

После завершения тестирования выводится количество попыток, оценка и список правильных и неправильных ответов (рисунок 1.8).

fen.distant.ru/test/math/3/3.7/3.7.htm

| | | |
|---|---|---|
| $A = \begin{pmatrix} 2 & -1 & 3 \\ 1 & 0 & 2 \end{pmatrix}, B = \begin{pmatrix} -1 & 1 \\ 3 & -2 \\ 1 & -2 \end{pmatrix}$ | 2 | $\begin{pmatrix} 6 & 0 \\ 0 & 4 \end{pmatrix}$ |
| $A = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, B = (-2 \ 3)$ | 3 | $\begin{pmatrix} -1 & 2 \\ 0 & 8 \end{pmatrix}$ |
| $A = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}, B = \begin{pmatrix} 3 & 0 \\ 0 & -2 \end{pmatrix}$ | 4 | $\begin{pmatrix} 1 & -1 \\ 3 & -2 \end{pmatrix}$ |
| | 5 | $\begin{pmatrix} -2 & -2 \\ 1 & -3 \end{pmatrix}$ |



Результаты теста
Попыток: 1

Оценка: 2.
Правильные ответы составили 27 %.
11 неправильных ответов: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
Постарайтесь исправить ошибки!

[предыдущий раздел](#) [к оглавлению](#) [следующий раздел](#)

© Кафедра высшей математики РХТУ им. Д.И. Менделеева.

Рисунок 1.8 – Пример контролирующей программы [21]

У пользователя есть возможность исправить неверно решенные задания, количество попыток будет увеличиваться (рисунок 1.9).

fen.distant.ru/test/math/3/3.7/3.7.htm

| | | |
|---|---|---|
| $A = \begin{pmatrix} 2 & -1 & 3 \\ 1 & 0 & 2 \end{pmatrix}, B = \begin{pmatrix} 3 & -2 \\ 1 & -2 \end{pmatrix}$ | 2 | $\begin{pmatrix} 0 & 4 \end{pmatrix}$ |
| $A = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, B = (-2 \ 3)$ | 3 | $\begin{pmatrix} -1 & 2 \\ 0 & 8 \end{pmatrix}$ |
| $A = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}, B = \begin{pmatrix} 3 & 0 \\ 0 & -2 \end{pmatrix}$ | 4 | $\begin{pmatrix} 1 & -1 \\ 3 & -2 \end{pmatrix}$ |
| | 5 | $\begin{pmatrix} -2 & -2 \\ 1 & -3 \end{pmatrix}$ |

Результаты теста
Попыток: 4

Оценка: 3.
Правильные ответы составили 53 %.
7 неправильных ответов: 4, 5, 6, 7, 8, 9, 10,
Постарайтесь исправить ошибки!

[предыдущий раздел](#) [к оглавлению](#) [следующий раздел](#)

© Кафедра высшей математики РХТУ им. Д.И. Менделеева.
© Лаборатория НМИ ВО РХТУ им. Д.И. Менделеева.

Рисунок 1.9 – Пример контролирующей программы [21]

1.4 Обзор существующих электронных обучающих программ

1.4.1 Интерактивный электронный учебник Алгебра-1

Интерактивный электронный учебник К. А. Мейрембекова Алгебра-1 (рисунок 1.10).

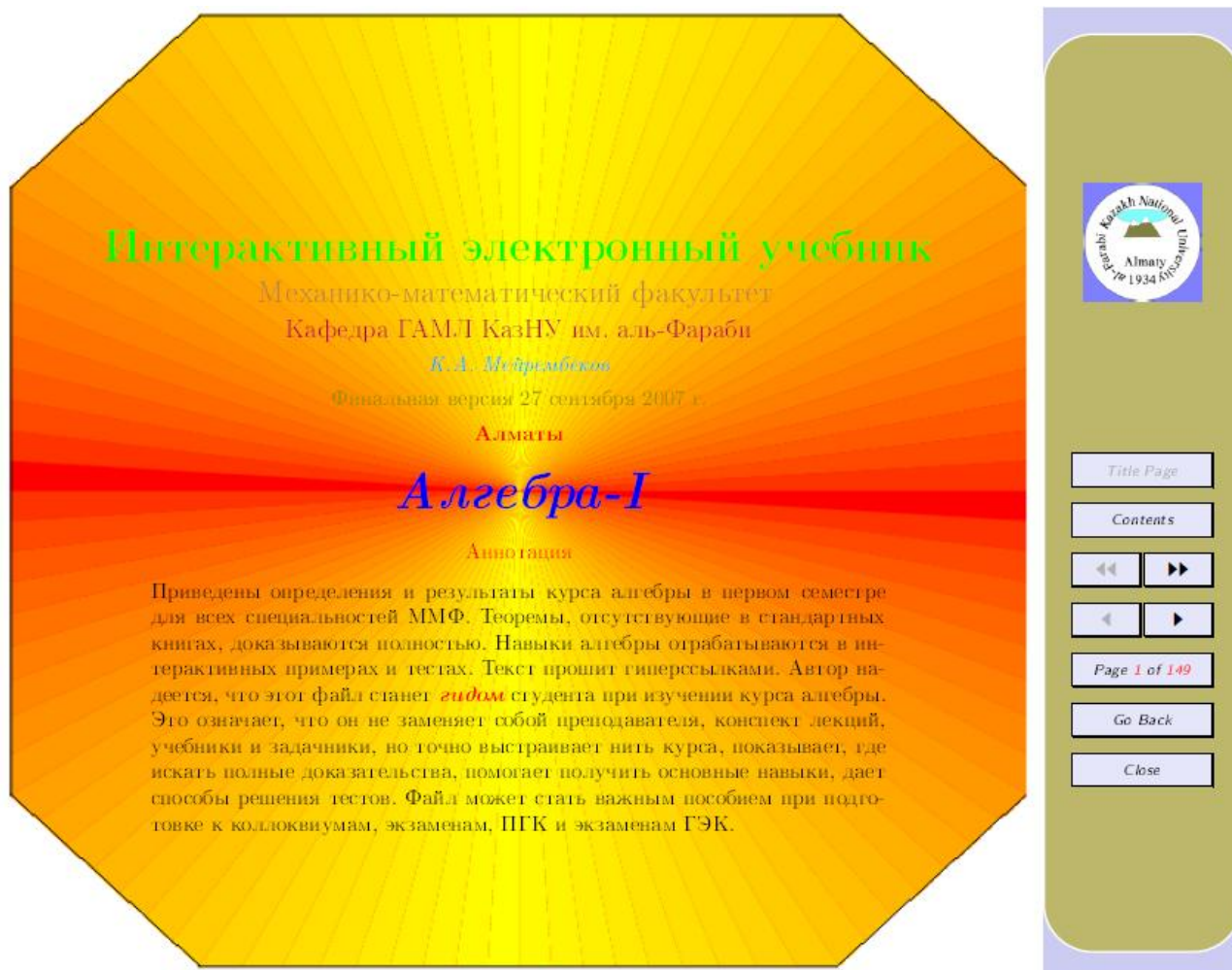


Рисунок 1.10 – Пример электронного учебного пособия 1 [15]

Требования к компьютеру и программному обеспечению. Компьютер может быть любым, начиная с Pentium-II. Для чтения файла нужен AdobeAcrobat или AdobeReader версии не ниже 6.0 с Javascript-модулем. Для проигрывания flash-роликов необходим MacromediaFlashplayer версии 7.0 или выше. Если у пользователя есть лишь AdobeReader, большая часть файлов будет читаться, но вложенные не pdf-файлы – нет.

В правой части окна есть навигация, с помощью которой можно легко перейти на нужную страницу. Кнопка Contents – содержание так же включает гиперссылки, что тоже упрощает пользование учебником.

Обучающее пособие включает разделы: матрицы, комплексные числа, определители, система линейных уравнений, многочлены, полиномы, Евклидовы пространства.

По ходу обучения встречаются задачи для проверки освоенного материала.

В конце учебника студент может проверить себя, пройдя тестирование по всему курсу. Чтобы начать тестирование, нужно нажать кнопку Start. Для проверки решений нажать в конце задач кнопку End и проверить ответы задач на совпадения, нажав Correct. Нажатие на зеленый кружок или галочку с правильным ответом приведет на страницу с решением соответствующей задачи.

Нажатие на красный квадрат на странице с решением вернет к рассматриваемому заданию.

The screenshot displays a digital learning environment. At the top, the section is titled "1.4. Определители". A pink-bordered box contains instructions: "Чтобы решать следующие задачи, требуемые для проверки усвоения, надо нажать кнопку **Start**, затем заполнить в каждой задаче правильные по вашему мнению ответы, ткнув мышкой на нужные радиокнопки. Повторное нажатие **Start** очищает ранее введенный выбор и система готова, чтобы заново приступить к обучению. Для проверки решений нажать в конце задач кнопку **end** и проверить ответы задач на совпадения, нажав на **Correct**. Нажатие на **зеленый кружочек** с правильным ответом переведет вас на страницу с решением соответствующей задачи. Нажатие на **красный квадрат** на странице с решением вернет вас к рассматриваемому заданию."

Below the instructions, there is a "Start" button. The first question asks: "1. Как изменится определитель, если из удвоенной первой строки отнять вторую строку?" The user has selected "не изменится" (not changed) with a checked radio button. The second question asks: "2. Как изменится определитель, если одновременно от первой строки отнять вторую, от второй третью и от третьей первую?" The user has selected "станет равным нулю" (will be equal to zero) with a checked radio button. A "Correct" button is visible, and the score is shown as "Score: 1 out of 2".

At the bottom, there is a reference: "Литература – книга Куроша [3].", the author "К.А. Мейрелбеков", the department "Кафедра ГАМЛ КазНУ им. аль-Фараби", and the page number "29".

On the right side, there is a navigation panel with buttons for "Title Page", "Contents", navigation arrows, "Page 29 of 149", "Go Back", and "Close". A logo for "Kazakh National University" is also present.

Рисунок 1.11 – Пример электронного учебного пособия 1 [15]

По окончании сеанса работы с файлом изменения не сохраняются.

1.4.2 Контролирующая программа онлайн

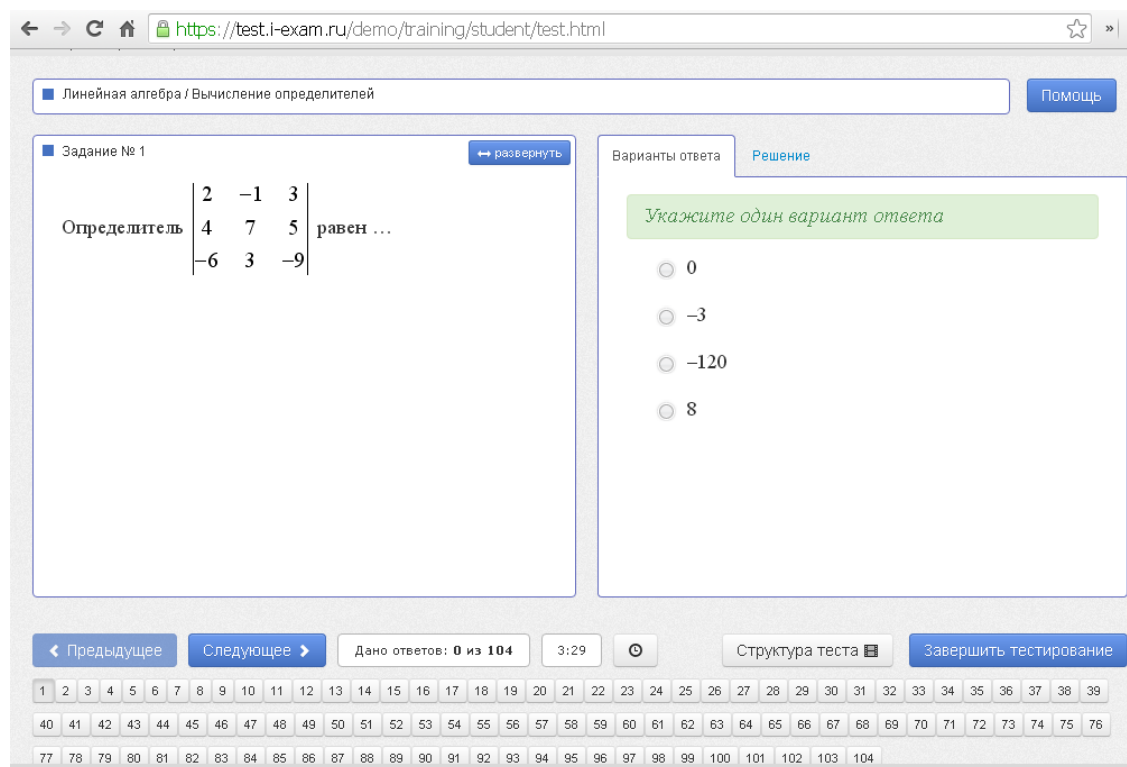


Рисунок 1.12 – Пример электронного учебного пособия 2 [22]

Данная обучающая программа включает разделы:

- линейная алгебра;
- абстрактная алгебра;
- аналитическая геометрия;
- дифференциальная геометрия;
- дифференциальное и интегральное исчисление;
- векторная алгебра;
- функциональный анализ;
- комплексный анализ;
- гармонический анализ;
- ряды;
- дифференциальные уравнения;
- операционное исчисление;
- теория вероятностей;
- математическая статистика;
- дискретная математика;
- численные методы;
- экономико-математические методы;
- экономико-математические модели.

В процессе тестирования пользователь работает с экраном, разделенным на 2 части: в левой части находится текст задания, в правой – варианты ответов. Вкладка Решение включает решение данного задания, а также содержит медиалекции (рисунок 1.13).

Помощь

Варианты ответа Решение

По свойству определителей получаем: $\begin{vmatrix} 2 & -1 & 3 \\ 4 & 7 & 5 \\ -6 & 3 & -9 \end{vmatrix} = -3 \cdot \begin{vmatrix} 2 & -1 & 3 \\ 4 & 7 & 5 \\ 2 & -1 & 3 \end{vmatrix} = 0.$

Определители

$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{13}a_{21}a_{32} +$

0:55 НИИ мониторинга качества образования 2:13

Рисунок 1.13 – Пример электронного учебного пособия 2 [22]

Тестирование включает в себя разные типы заданий:

- Задание с выбором одного правильного ответа – данный тип задания предполагает выбор одного варианта ответа из предложенных;
- Задание с двумя и более правильными ответами – данный тип задания предполагает выбор двух и более вариантов ответа из предложенных;
- Задание на установление правильной последовательности – данный тип задания предполагает указание правильной последовательности ответов;
- Задание на установление соответствия – данный тип задания предполагает установление соответствия между элементами двух множеств;
- Задание с кратким ответом – данный тип задания предполагает ввод с клавиатуры краткого ответа.

При выборе неверного ответа программа выводит совет по решению, при выборе верного – переходит к следующему вопросу (рисунок 1.14).

Задание № 2 ← развернуть

Даны матрицы $A = \begin{pmatrix} 1 & 2 & 0 \\ -2 & 3 & 5 \\ -3 & 4 & 7 \end{pmatrix}$ и $B = \begin{pmatrix} 4 & -3 & 2 \\ 5 & -6 & -4 \\ 0 & 2 & -5 \end{pmatrix}$

. Тогда матрица $C = A + 2B$ имеет вид...

Варианты ответа Решение

Укажите один вариант ответа

$\begin{pmatrix} 5 & -1 & 2 \\ 3 & -3 & 1 \\ -3 & 6 & 2 \end{pmatrix}$ Следующее >

Ответ неверный!

При выполнении операций над матрицами не учтен коэффициент 2, на который следует умножить все элементы матрицы B. Элементы матрицы C находятся как:

$$c_{ij} = a_{ij} + 2b_{ij}.$$

$\begin{pmatrix} 6 & 1 & 2 \\ 1 & 0 & 6 \end{pmatrix}$

Рисунок 1.14 – Пример электронного учебного пособия 2 [22]

В правом верхнем углу находится кнопка помощь: здесь подробно описано, как работать с обучающей программой.

Тестирование ведется на время, но эту функцию можно отключить, нажав на часы в нижней части экрана.

После завершения тестирования пользователь может посмотреть результат (рисунок 1.15).

Результаты тестирования

| | |
|--------------------------------|---------------|
| Дисциплина | Математика |
| Режим тестирования | Обучение |
| Дата тестирования | 17.04.2018 |
| Продолжительность тестирования | 29 минут |
| Дано ответов | 5 из 104 |
| Правильных ответов | 2% (3 из 104) |

Вернуться к текущему тестированию
Пройти новое тестирование

Рисунок 1.15 – Пример электронного учебного пособия 2 [22]

1.5 Требования к содержанию учебного пособия по линейной алгебре

Целью освоения курса «Линейная алгебра» является ознакомление студентов с основами линейной алгебры. В данной же работе будут рассмотрены только действия с векторами и матрицами. При необходимости преподаватель сможет добавлять в пособие новые главы и параграфы по данной дисциплине.

1.5.1 Основные понятия

Матрицей размерности $p \times q$ называют прямоугольную таблицу из чисел, которые расположены в p строках и q столбцах

Числа, образующие матрицу называются элементами матрицы. Матрицы, у которых число строк равно числу столбцов, называют квадратными, а число строк такой матрицы называют её порядком. Например, матрица $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ является квадратной матрицей второго порядка.

Если матрица состоит из одной строки, то её называют вектор-строка:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \end{pmatrix}.$$

Если матрица состоит из одного столбца, то её называют вектор-столбец:

$$\begin{pmatrix} a_{11} \\ a_{21} \\ \dots \\ a_{n1} \end{pmatrix}.$$

1.5.2 Операция сложения матриц

Операция сложения определена только для матриц одной размерности. Размерность матрицы задает число строк p и число столбцов n . Размерность матрицы записывается в виде $p \times n$.

Сумма двух матриц $A = \|a_{ij}\|$, $i = 1, 2, \dots, p$, $j = 1, 2, \dots, n$ и $B = \|b_{ij}\|$, $i = 1, 2, \dots, p$, $j = 1, 2, \dots, n$ - это матрица, элементы которой равны сумме соответствующих элементов матриц A и B , то есть, $A + B = \|a_{ij} + b_{ij}\|$, $i = 1, 2, \dots, p$, $j = 1, 2, \dots, n$.

(1)

Таким образом, результатом операции сложения двух матриц является матрица той же размерности.

1.5.3 Операция умножения матрицы на число

Произведение матрицы $A = \|a_{ij}\|$, $i = 1, 2, \dots, p$, $j = 1, 2, \dots, n$ и действительного (или комплексного) числа λ - это матрица, элементы которой получаются умножением соответствующих элементов исходной матрицы на число λ , то есть, $\lambda A = \|\lambda a_{ij}\|$, $i = 1, 2, \dots, p$, $j = 1, 2, \dots, n$.

(2)

Таким образом, результатом умножения матрицы на число является матрица той же размерности.

1.5.4 Операция умножения двух матриц

Операция умножения двух матриц A и B определена для случая, когда число столбцов матрицы A равно числу строк матрицы B .

Произведение матрицы A порядка $p \times n$ и матрицы B порядка $n \times q$ - это такая матрица C порядка $p \times q$, каждый элемент которой равен сумме произведений элементов i -ой строки матрицы A на соответствующие элементы j -ого столбца матрицы B , то есть, $c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \dots + a_{in} \cdot b_{nj}$, $i = 1, 2, \dots, p$, $j = 1, 2, \dots, q$.

(3)

Таким образом, результатом операции умножения матрицы порядка $p \times n$ на матрицу порядка $n \times q$ является матрица порядка $p \times q$.

1.5.5 Операция сложения векторов

Так как вектор является матрицей размерностью $1 \times n$ (вектор-строка) или $n \times 1$ (вектор-столбец), то операции с векторами выполняются аналогично операциям с матрицами.

Для векторов одной размерности определена операция сложения.

Сумма векторов $A = \parallel a_{1j} \parallel, j = 1, 2, \dots, n$ и $B = \parallel b_{1j} \parallel, j = 1, 2, \dots, n$ - это вектор-строка, элементы которого равны сумме соответствующих элементов векторов A и B , то есть, $A + B = \parallel a_{1j} + b_{1j} \parallel, j = 1, 2, \dots, n$.

Сумма векторов $A = \parallel a_{i1} \parallel, i = 1, 2, \dots, n$ и $B = \parallel b_{i1} \parallel, i = 1, 2, \dots, n$ - это вектор-столбец, элементы которого равны сумме соответствующих элементов векторов A и B , то есть, $A + B = \parallel a_{i1} + b_{i1} \parallel, i = 1, 2, \dots, n$.

(4)

Результатом операции сложения двух векторов является вектор той же размерности.

1.5.6 Операция умножения вектора на число

Произведение вектора $A = \parallel a_{1j} \parallel, j = 1, 2, \dots, n$ и действительного (или комплексного) числа λ - это вектор-строка, элементы которого получаются умножением соответствующих элементов исходного вектора на число λ , то есть, $\lambda A = \parallel \lambda a_{1j} \parallel, j = 1, 2, \dots, n$.

Произведение вектора $A = \parallel a_{i1} \parallel, i = 1, 2, \dots, n$ и действительного (или комплексного) числа λ - это вектор-столбец, элементы которого получаются умножением соответствующих элементов исходного вектора на число λ , то есть, $\lambda A = \parallel \lambda a_{i1} \parallel, i = 1, 2, \dots, n$.

(5)

Результатом операции умножения вектора на число является вектор той же размерности.

1.5.7 Операция умножения двух векторов

Операция умножения двух векторов A и B определена для случая, когда один из векторов является строкой, другой – столбцом, при этом количество элементов у них одинаково.

Руководствуясь правилом перемножения "строка на столбец", можно составить из векторов два произведения: $A^T B$ и AB^T , где A^T и B^T – транспонированные вектора или вектор-строки.

Произведение вектора A порядка $1 \times n$ и вектора B порядка $n \times 1$ - это число c , равное сумме произведений элементов вектора A на соответствующие элементы вектора B , то есть, $c = a_{11} \cdot b_{11} + a_{12} \cdot b_{21} + \dots + a_{1n} \cdot b_{n1}$.

$$A \cdot B = (a_{11} \ a_{12} \ \dots \ a_{1n}) \cdot \begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \end{pmatrix} =$$

$$= a_{11}b_{11} + a_{12}b_{21} + \dots + a_{1n}b_{n1} \quad a_{11}b_{12} + a_{12}b_{22} + \dots + a_{1n}b_{n2} \quad \dots \quad a_{11}b_{1q} + a_{12}b_{2q} + \dots + a_{1n}b_{nq} \quad (6)$$

Результатом операции умножения вектор-строки на вектор-столбец является число. Такое произведение называется скалярным или внутренним.

Произведение вектора A порядка $n \times 1$ и вектора B порядка $1 \times n$ - это матрица C порядка $n \times n$, каждый элемент которой равен произведению элементов i -ой строки матрицы A на соответствующие элементы j -ого столбца матрицы B , то есть, $c_{ij} = a_{i1} \cdot b_{1j}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$.

$$A \cdot B = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \cdot (b_{11} \ b_{12} \ \dots \ b_{1n}) = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & \dots & a_{11}b_{1n} \\ a_{21}b_{11} & a_{21}b_{12} & \dots & a_{21}b_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}b_{11} & a_{n1}b_{12} & \dots & a_{n1}b_{1n} \end{pmatrix} \quad (7)$$

Результатом операции умножения вектор-столбца на вектор-строку является матрица размерности $n \times n$. Такое произведение называется внешним.

1.5.8 Операция умножения вектора на матрицу

По правилу умножения матриц число столбцов матрицы A равно числу строк матрицы B . Следовательно, умножение вектора на матрицу производится только в том случае, когда вектор-строка умножается слева, а вектор-столбец – справа.

Пусть A – вектор-строка размерности $1 \times n$, B – матрица размерности $n \times q$.

$$A \cdot B = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1q} \\ b_{21} & b_{22} & \dots & b_{2q} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nq} \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} + \dots + a_{1n}b_{n1} & a_{11}b_{12} + a_{12}b_{22} + \dots + a_{1n}b_{n2} & \dots & a_{11}b_{1q} + a_{12}b_{2q} + \dots + a_{1n}b_{nq} \end{pmatrix} \quad (8)$$

Пусть A – матрица размерности $p \times n$, B – вектор-столбец размерности $n \times 1$.

$$A \cdot B = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{p1} & a_{p2} & \dots & a_{pn} \end{pmatrix} \cdot \begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} + \dots + a_{1n}b_{n1} \\ a_{21}b_{11} + a_{22}b_{21} + \dots + a_{2n}b_{n1} \\ \vdots \\ a_{p1}b_{11} + a_{p2}b_{21} + \dots + a_{pn}b_{n1} \end{pmatrix} \quad (9)$$

В результате умножения вектора на матрицу получаем вектор той же размерности.

1.6 Выбор языка программирования и среды разработки

Для реализации электронного обучающего пособия использовалась среда разработки Borland C++ Builder for Microsoft Windows Version 10.0, язык разработки – C++.

1.7 Постановка задачи

Целью данной работы является разработка обучающей программы по линейной алгебре на языке C++.

Обучающая программа состоит из двух частей – теоретической и практической. Теоретическая часть содержит лекционный материал и мультимедиа. Практическая часть включает в себя тренировочные задания.

После запуска программы появляется главное окно (рисунок 1.16):

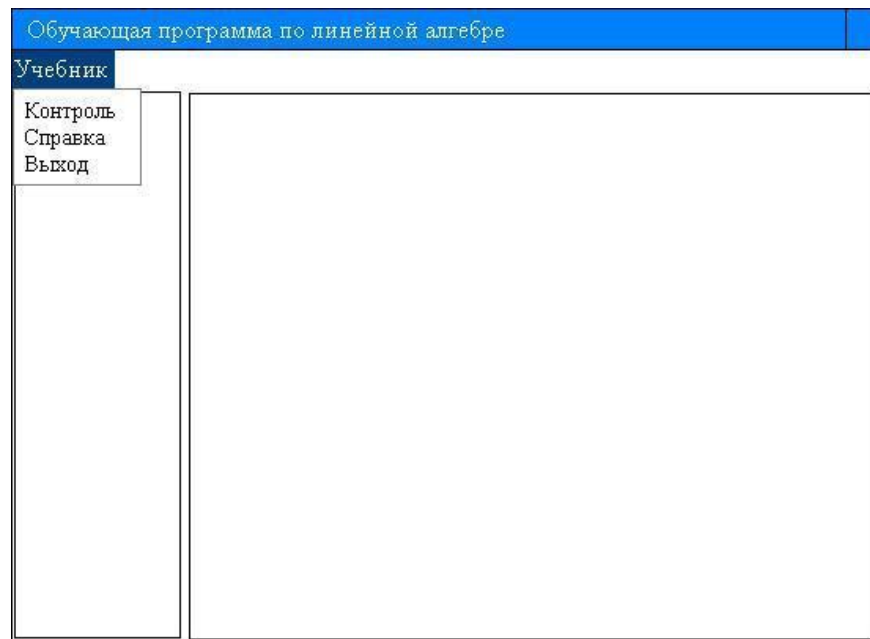


Рисунок 1.16 – Окно программы

Главное окно программы разделено на 2 части. Правая часть содержит теоретический материал, в левой части находится содержание, снабженное гиперссылками, позволяющими перемещаться по тексту.

В верхней части окна располагается меню с выпадающим списком, содержащим пункты: контроль, справка и выход.

В пункте «Контроль» содержатся тренировочные задания (дидактические материалы), предназначенные для организации самостоятельной работы студентов. Они содержат однотипные задания, чтобы обучаемые по собственной инициативе или под контролем преподавателя могли отрабатывать практические навыки решения определенных задач.

При выборе меню «Справка» пользователь найдет ответы на вопросы, возникшие у него при работе с приложением.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. рассмотреть классификацию компьютерных учебных программ;
2. провести обзор существующих обучающих электронных пособий по линейной алгебре;
3. выбор языка и среды разработки;
4. разработка архитектуры приложения;
5. разработка математической модели;
6. разработка интерфейса программы;
7. реализация ПО;
8. тестирование приложения.

2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ В ЛИНЕЙНОЙ АЛГЕБРЕ

2.1 Операция сложения матриц

Операция сложения определена только для матриц одной размерности. Размерность матрицы задает число строк i и число столбцов j . Размерность матрицы записывается в виде $i \times j$. Каждый элемент новой матрицы (с индексами ij) будет равен сумме элементов слагаемых матриц (с теми же индексами ij).

(10)

Таким образом, результатом операции сложения двух матриц является матрица той же размерности.

2.2 Операция умножения матрицы на число

Результатом произведения матрицы на число является матрица, размерность которой равна размерности исходной матрицы.

(11)

2.3 Операция умножения двух матриц

Операция умножения двух матриц A и B определена для случая, когда число столбцов матрицы A равно числу строк матрицы B .

Так если имеем матрицу A с размерами i, k и матрицу B с размерами k, j то результатом будет матрица C с размерами i, j . Каждый элемент новой матрицы (с индексами ij) будет равен сумме произведений элементов i -строки первой матрицы на элементы j -столбца второй матрицы.

(12)

2.4 Операции с векторами

Так как вектор является матрицей размерностью $1 \times n$ (вектор-строка) или $n \times 1$ (вектор-столбец), то операции с векторами выполняются аналогично операциям с матрицами.

3 РАЗРАБОТКА ОБУЧАЮЩЕЙ ПРОГРАММЫ

3.1 Описание алгоритма работы программы.

После инициализации приложения создается главное окно программы, и запускается цикл обработки сообщений. Соответствующая схема алгоритма приведена на рисунке 3.1.

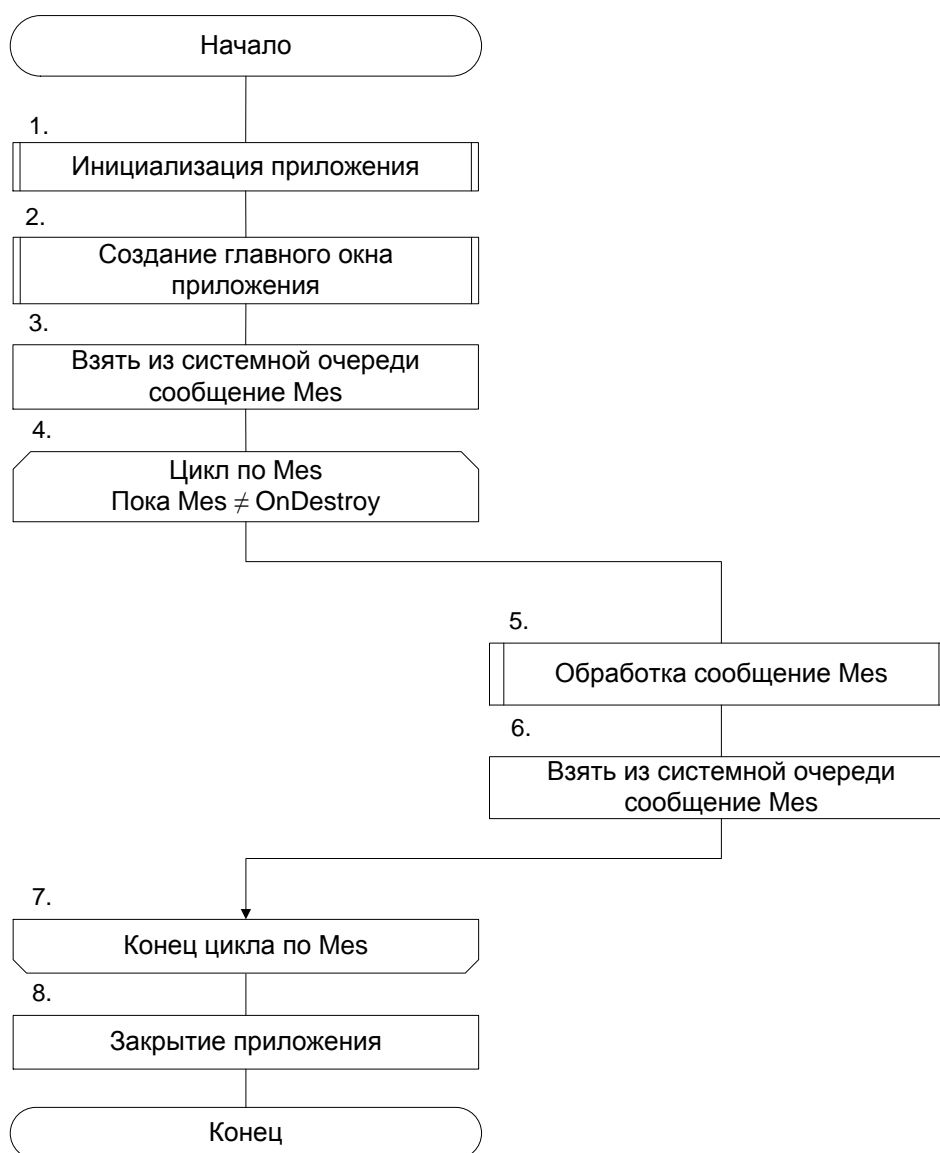


Рисунок 3.1 – Алгоритм работы программы

3.1.1 Вспомогательный алгоритм обработки сообщений.

При создании окна выполняется инициализация списка оглавления для учебного пособия. При выборе пункта содержания, обучающий материал выводится в окно веб-браузера.

Если выбран пункт меню «Контроль», открывается окно с тренировочными упражнениями;

Если выбран пункт меню «Справка», открывается окно со справочным материалом;

При выборе пункта меню «Выход», приложение закрывается.

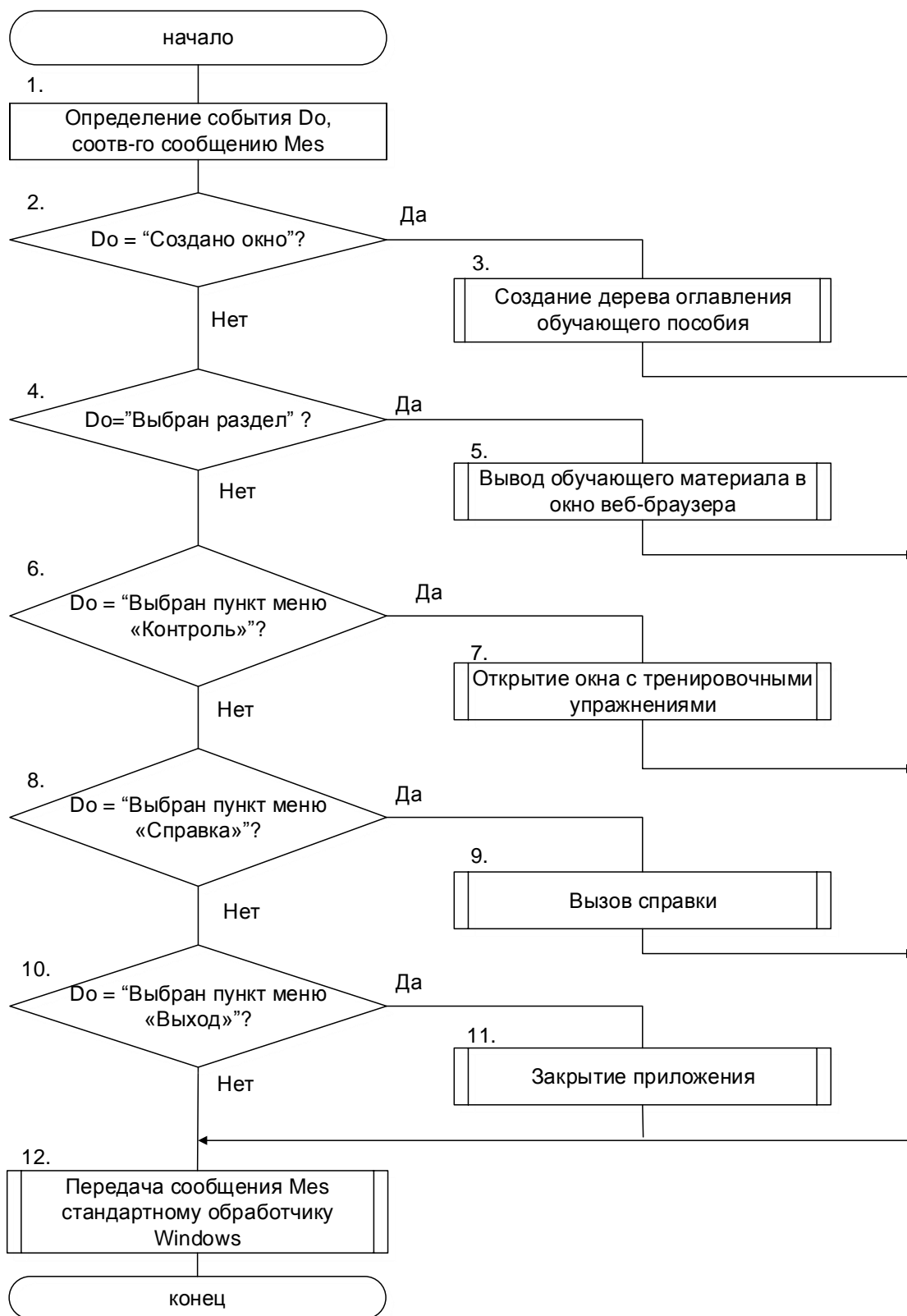


Рисунок 3.2 – Вспомогательный алгоритм обработки сообщений

3.1.2 Вспомогательный алгоритм вывода разделов пособия.

После выбора пользователем определенного раздела пособия происходит поиск файла и его вывод в окно веб-браузера.

На рисунке 3.3 приведена схема алгоритма работы данного модуля.

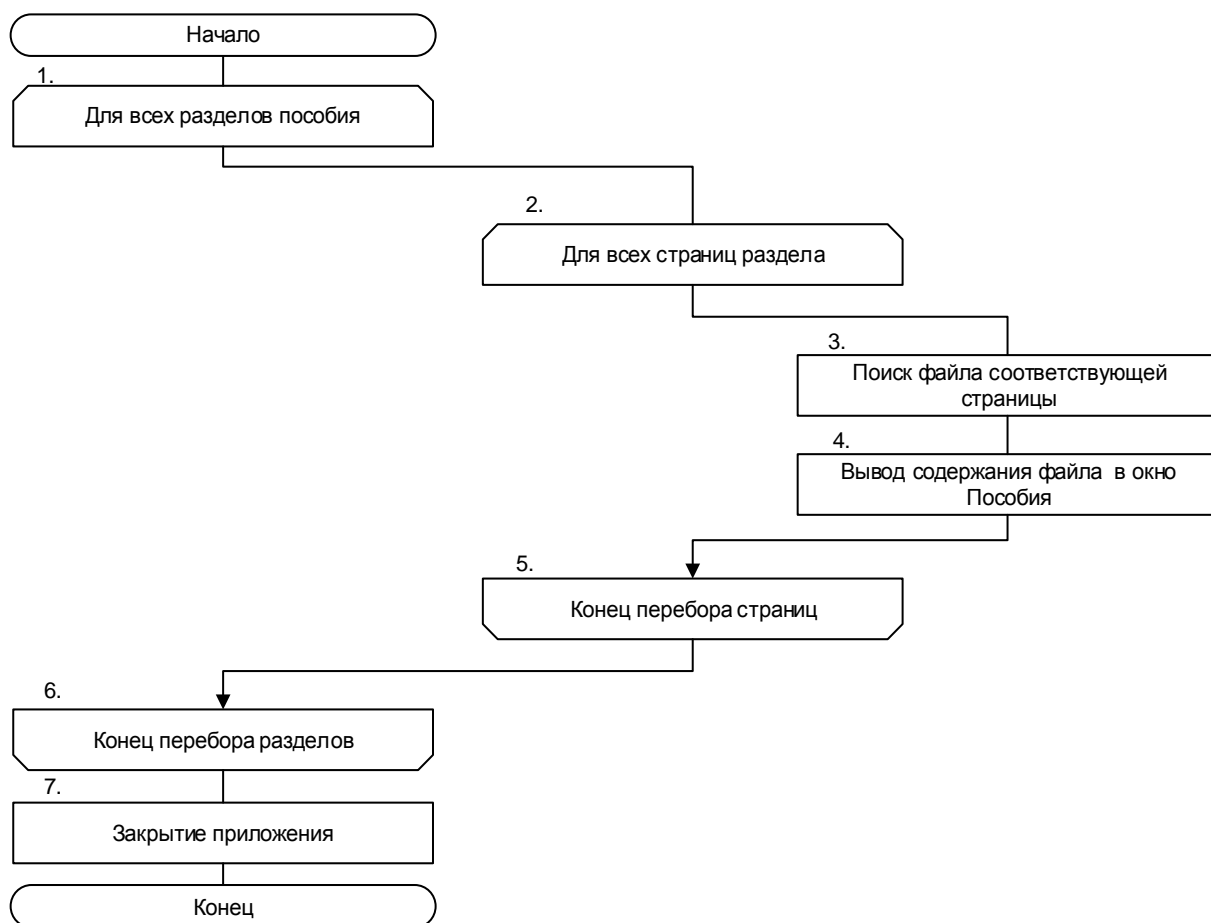


Рисунок 3.3 – Вспомогательный алгоритм вывода разделов пособия

3.1.3 Вспомогательный алгоритм

В процессе прорешивания примеров можно переключаться с одной вкладки на другую и возвращаться к уже решенным примерам. При нажатии на кнопку «Проверить» приложение помечает красным цветом неправильно введенные ответы. При возврате в «Учебник» открывается форма с обучающим материалом, при этом введенные ранее ответы стираются.

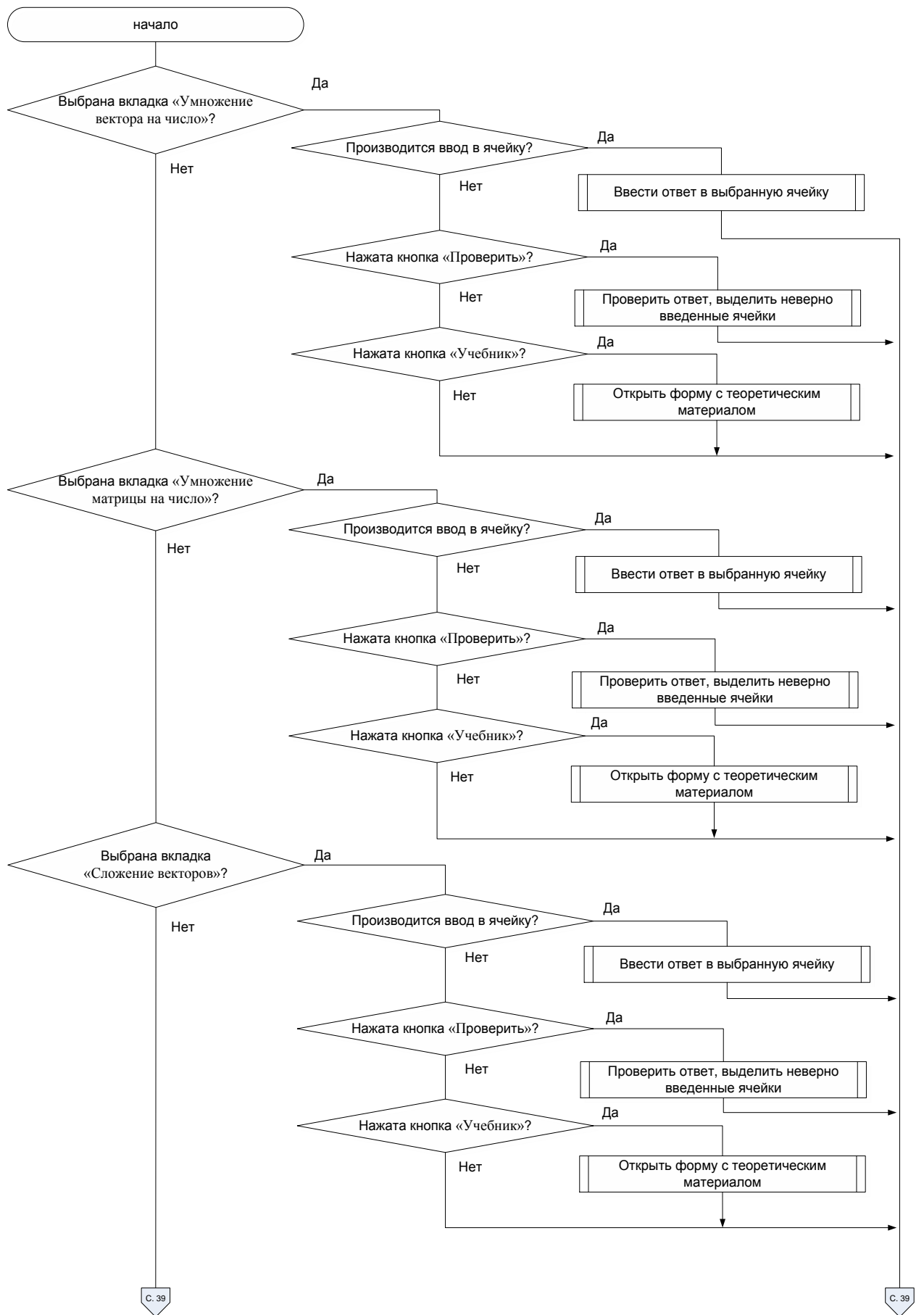


Рисунок 3.4 – Вспомогательный алгоритм

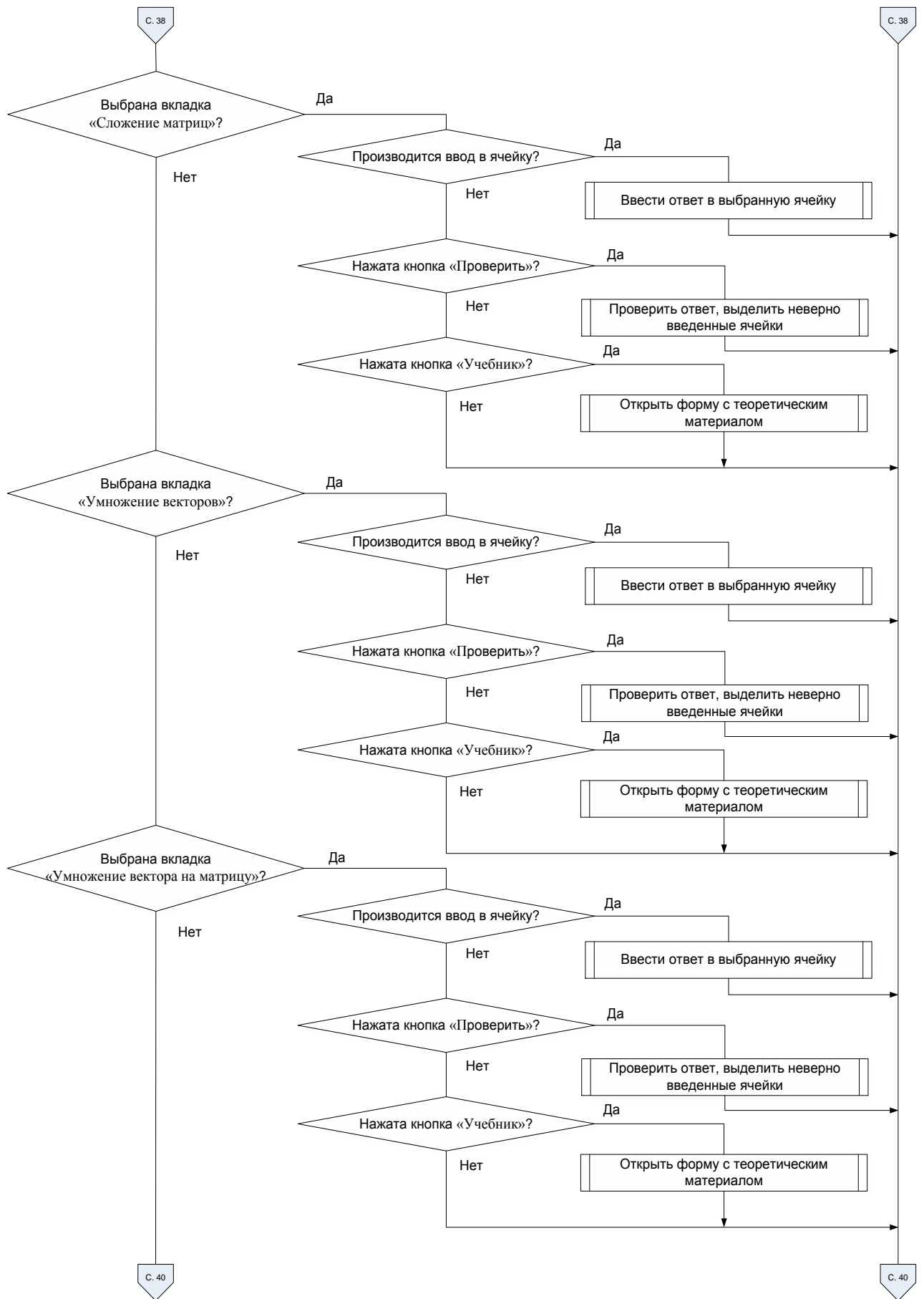


Рисунок 3.5 – Вспомогательный алгоритм

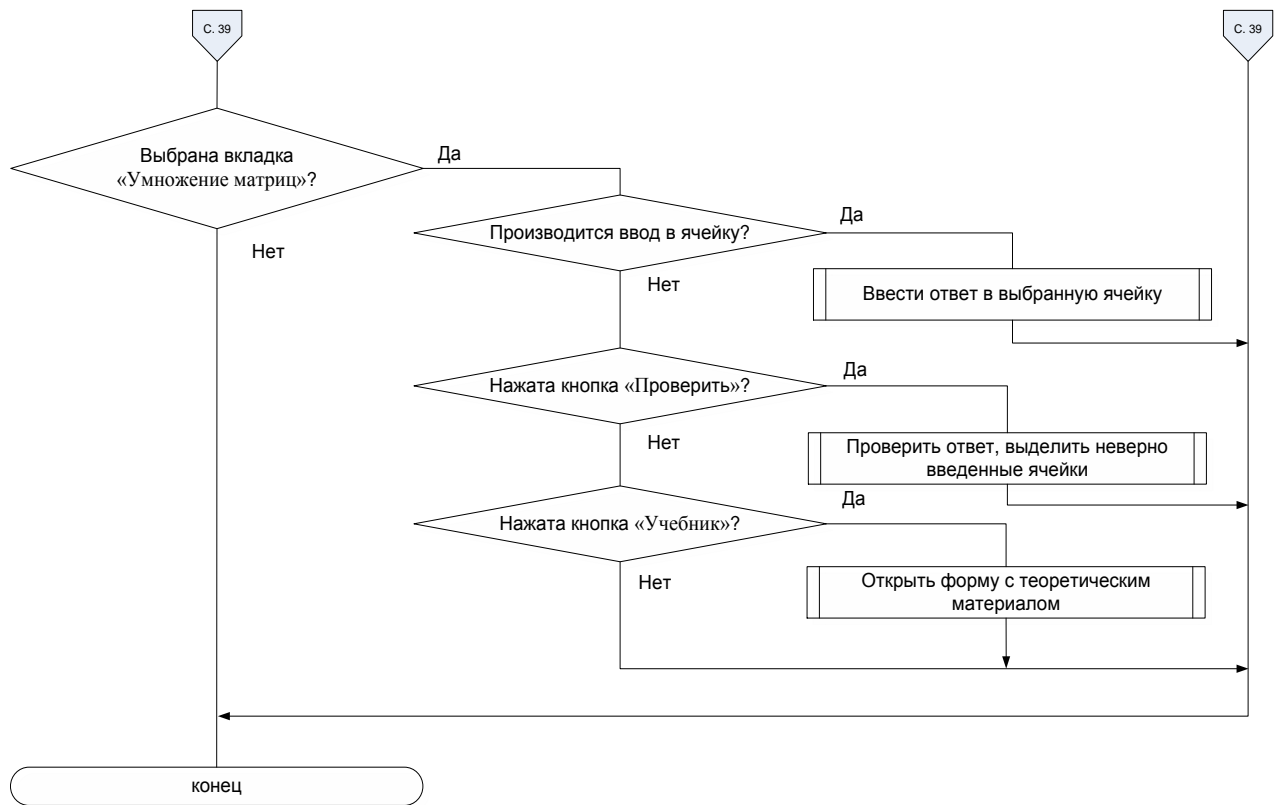


Рисунок 3.6 – Вспомогательный алгоритм

3.2 Разработка пользовательского интерфейса

После запуска приложения «Обучающая программа по линейной алгебре» открывается главное окно программы (Рисунок 3.7), которое содержит дерево оглавления пособия и окно Web-браузера.

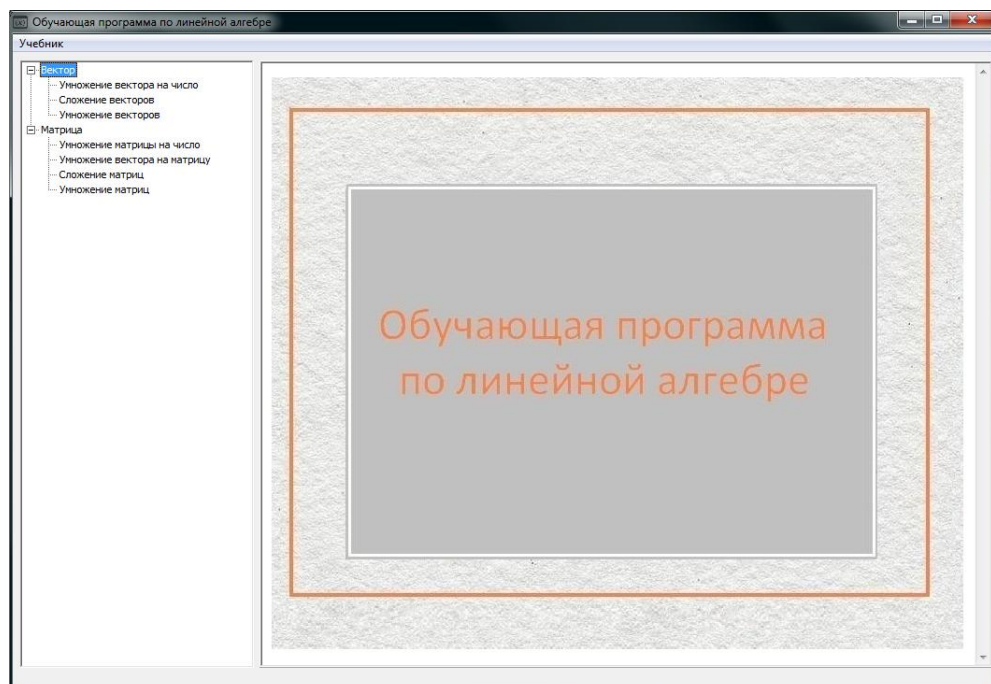


Рисунок 3.7 – Заставка

Вначале в окне Web-браузера выводится заставка с названием пособия. Работа студента с приложением начинается обучающим материалом. Для выбора раздела студент должен щелкнуть по соответствующему узлу из дерева (пунктов оглавления).

Оглавление реализуем с помощью компонента TreeView (Рисунок 3.8). Выбор нужного пункта осуществляется нажатием левой кнопки мыши.

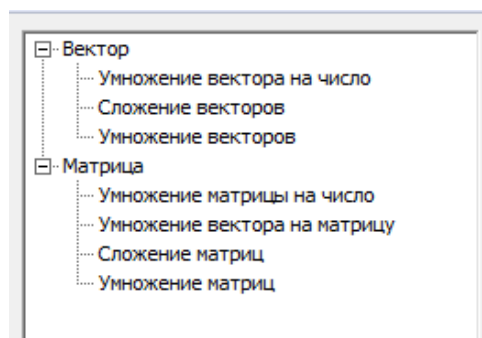


Рисунок 3.8 – Содержание

При выборе одного из пунктов оглавления материал выводится на форму из html-файла в WebBrowser. Теоретический материал содержит основные понятия и описание алгоритмов для проведения операций над матрицами и векторами (Рисунок 3.9-3.10).

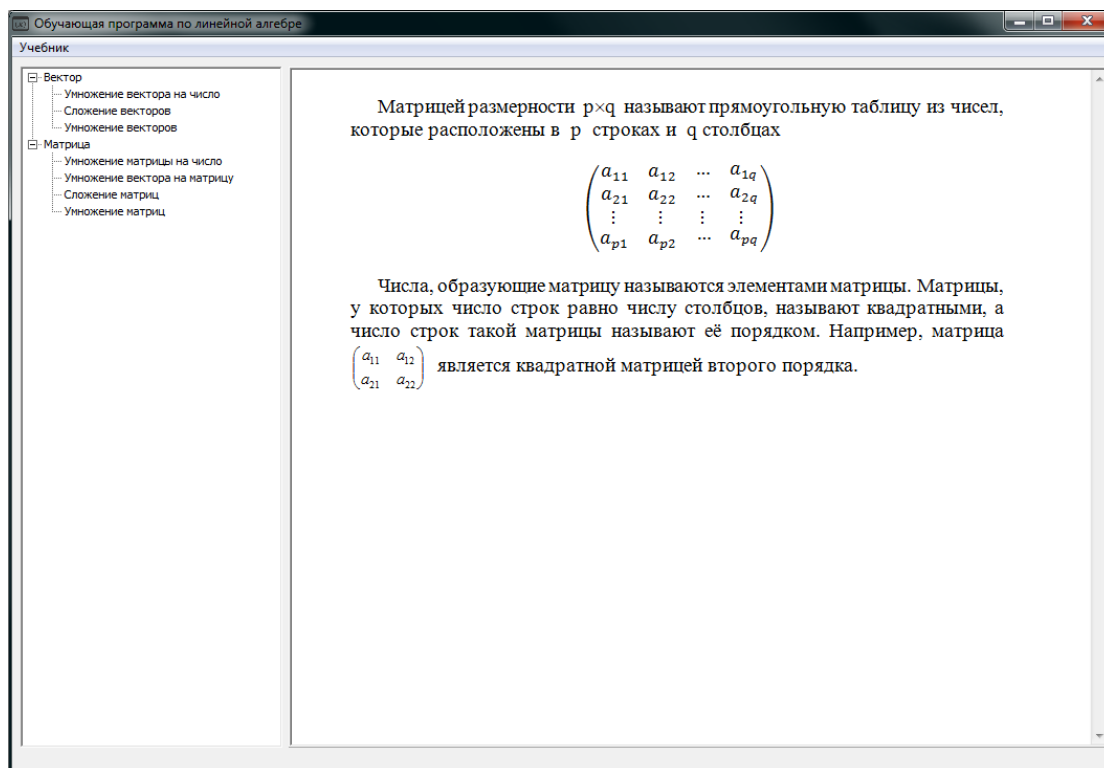


Рисунок 3.9 – Понятие матрицы

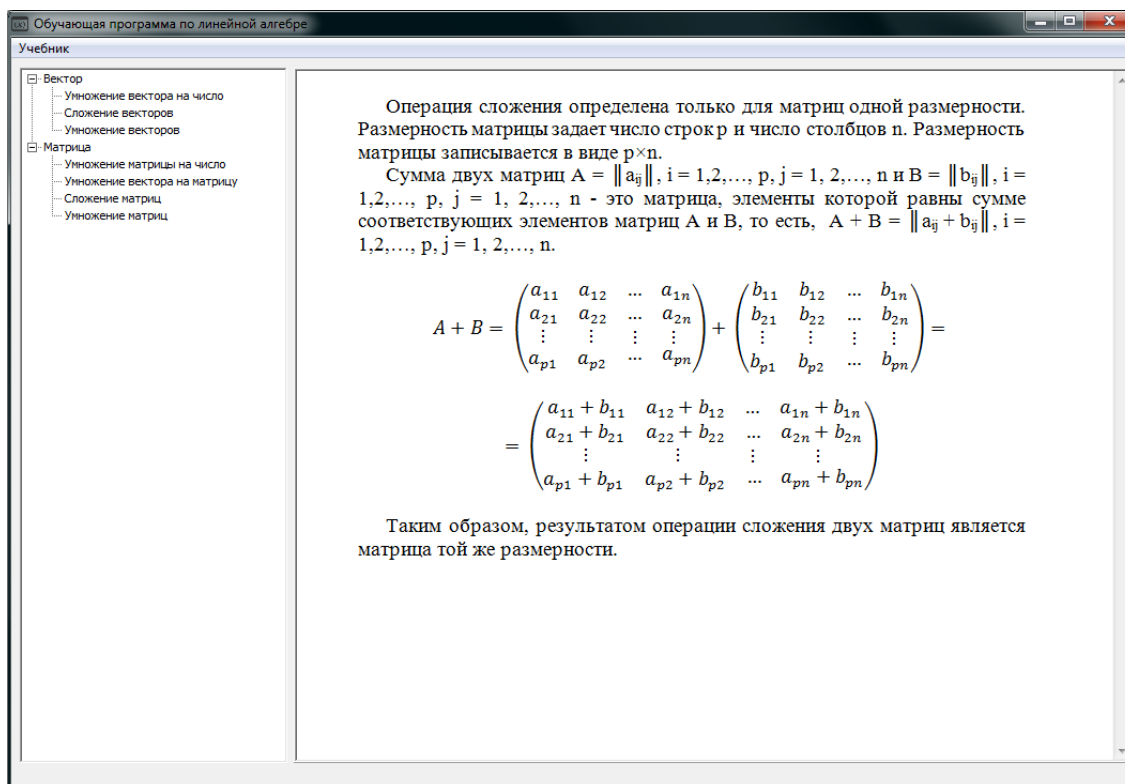


Рисунок 3.10 – Операция умножения векторов

В левом верхнем углу находится выпадающий список с пунктами «Контроль» «Справка» и «Выход» (Рисунок 3.11).

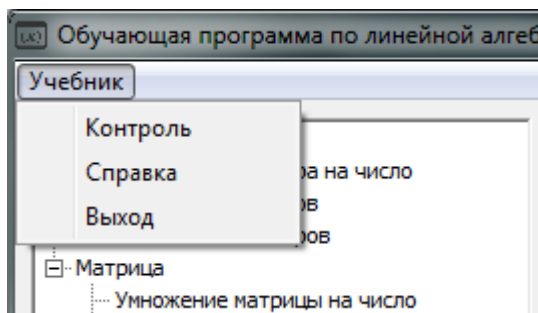


Рисунок 3.11 – Выпадающий список

Вкладка «Контроль» включает в себя тренировочные задания. Реализованы следующие типы заданий:

1. Сложение векторов;
2. Умножение вектора на число;
3. Умножение векторов;
4. Сложение матриц;
5. Умножение матрицы на число;
6. Умножение матриц;
7. Умножение матрицы на вектор.

Задания решаются по стандартному алгоритму, исходные данные отличаются размерностью матриц и числовыми данными в них, которые генерируются случайным образом.

Управление на форме «Контроль» осуществляется с помощью мыши и клавиатуры. С клавиатуры осуществляется ввод ответов.

На форме «Контроль» расположены 7 вкладок с разными типами заданий. Каждая вкладка включает в себя определенное задание, место для ввода ответа, кнопку проверки результата и кнопку «Учебник». При нажатии на кнопку «Учебник» студент переходит на форму с обучающими материалами, а все ответы будут потеряны. При решении примеров можно переключать вкладки и возвращаться к уже решенным заданиям для проверки или изменения результатов. Если ответ введен неверно, программа выделяет соответствующие ячейки красным цветом. Примеры работы обучающего пособия изображены на рисунках 3.12-3.13.

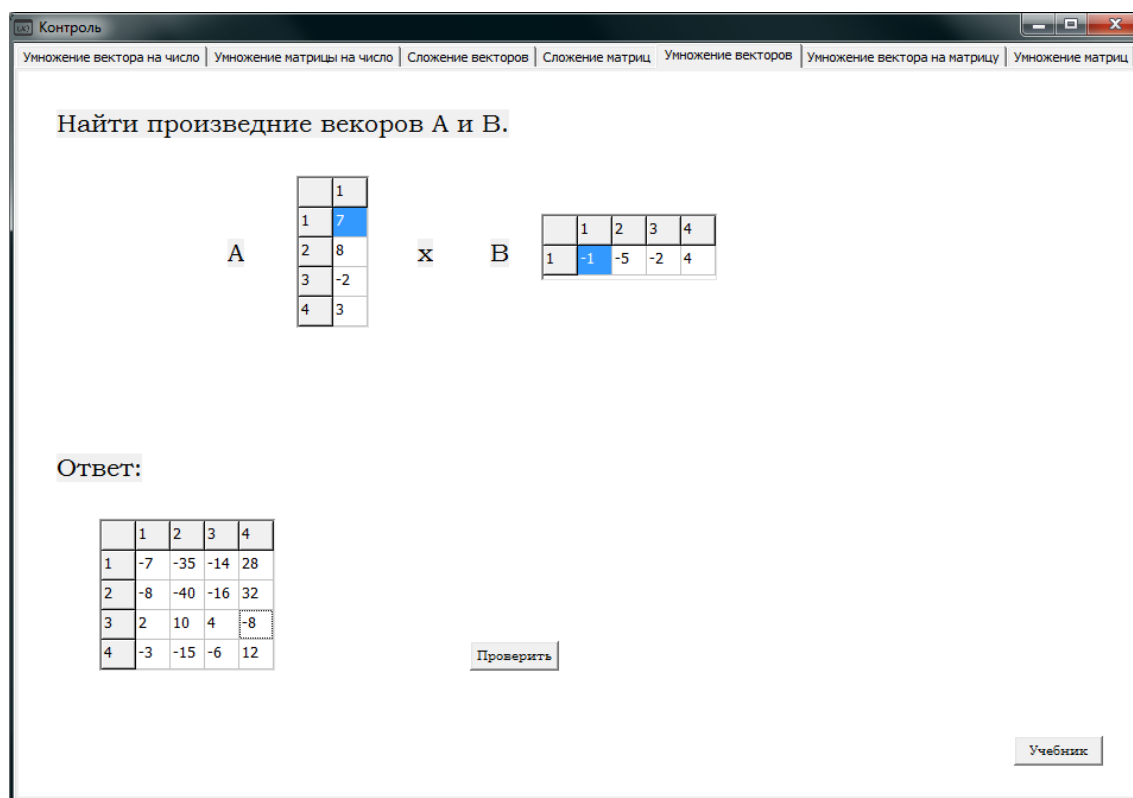


Рисунок 3.12 – Вкладка умножения векторов

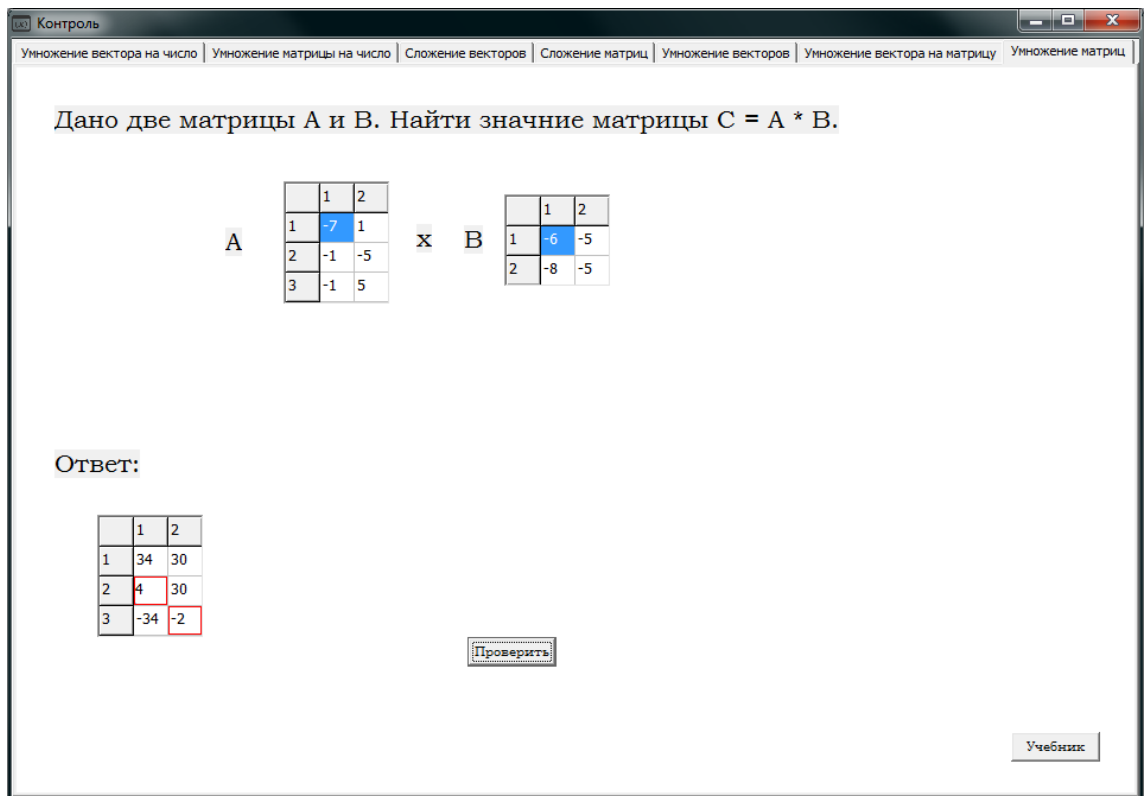


Рисунок 3.13 – Вкладка Умножения матриц матриц

Получить необходимую справочную информацию студент может, выбрав пункт меню «Справка» (Рисунок 3.14).

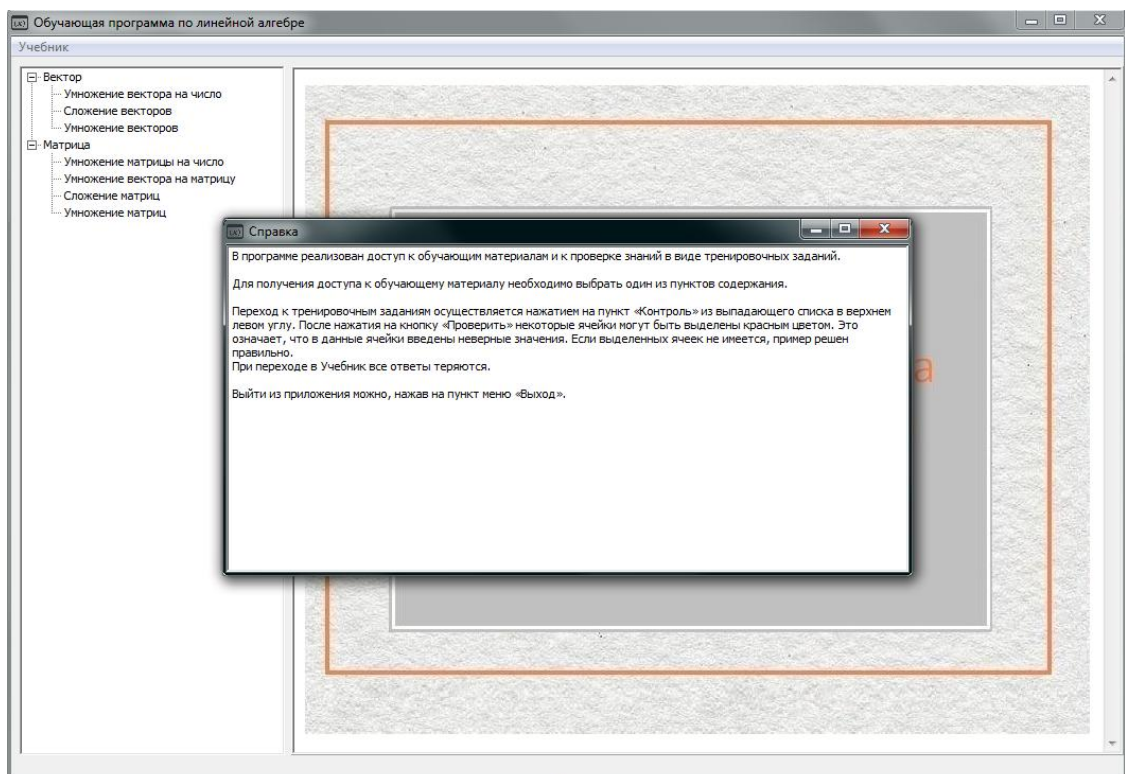


Рисунок 3.14 – Справка

ЗАКЛЮЧЕНИЕ

Цель, поставленная в работе, достигнута. Разработана и реализована обучающая программа по линейной алгебре. Были рассмотрены классификации компьютерных учебных программ, проведен обзор существующих обучающих электронных пособий по линейной алгебре, разработана математическая модель и интерфейс программы, реализовано приложение.

Приложение включает в себя теоретическую часть с обучающим материалом и практическую – с тренировочными заданиями.

По итогам работы сделаны следующие выводы:

1. Применение данной системы будет способствовать повышению эффективности формирования профессиональных компетенций будущих бакалавров прикладной математики и информатики при обучении линейной алгебре.
2. Простой интерфейс системы не вызовет затруднений у студентов при обучении работе с системой, данное обучение не займет много времени.
3. Нестандартный подход к проведению практических занятий и организации самостоятельной работы позволит повысить мотивацию студентов к освоению курса линейной алгебры.

Стоит отметить, что в дальнейшем программа может быть улучшена. Дальнейшие работы могут быть направлены на реализацию модуля преподавателя, включающего возможность добавления глав теоретического материала и новых заданий, на реализацию заданий разного уровня сложности, реализацию режима тренировки и контроля знаний. Также можно добавить возможность студенту заходить в систему под своим именем.

ЛИТЕРАТУРА

1. Киселев, Г.М. Информационные технологии в педагогическом образовании: Учебник / Г.М. Киселев, Р.В. Бочкова. – М.: Издательско-торговая корпорация «Дашков и К^о», 2012. – 308 с.
2. Корчемкина Ю.В. Обучение линейной алгебре с применением практико-ориентированной учебно-аналитической информационной системы // Современные проблемы науки и образования. – 2016. – № 4. [Электронный ресурс]. URL: <http://www.science-education.ru/ru/article/view?id=24924>
3. Кравченя, Э. М. Информационные и компьютерные технологии в образовании – БНТУ, 2017 – 171 с.
4. Актуальные задачи педагогики (II): материалы междунар. заоч. науч. конф. (г. Чита, июнь 2012 г.). – Чита: Издательство Молодой ученый, 2012. – vi, 150 с.
5. Отличие электронного издания от печатного. Цикл статей / Институт социологии РАН, Григорьева Е.И., Ситдииков И.М. // Официальный сайт ИС РАН - 2013. 76 стр. [Электронный ресурс]. URL: <http://www.isras.ru/publ.html?id=2801>
6. Акритас А. Основы компьютерной алгебры с приложениями: Пер. с англ. — М., Мир, 1994. — 544 с.
7. Система компьютерной алгебры GAP 4.7 / А.Б. Коновалов // Brief GAP Guidebook in Russian [Электронный ресурс]. – 2014. – 87 с.
8. Полат, Е.С. Современные педагогические и информационные технологии в системе образования: Учебное пособие / Е.С. Полат, М.Ю. Бухаркина. – М.: Издательский центр «Академия», 2010. – 368 с.
9. Электронный учебник [Электронный ресурс]. Режим доступа – https://ru.wikipedia.org/wiki/Электронный_учебник
10. Система компьютерной алгебры GAP [Электронный ресурс]. Режим доступа – <https://ru.wikipedia.org/wiki/GAP>
11. Основные функции систем компьютерной алгебры : пособие для студентов факультета прикладной математики и информатики / В. Б. Таранчук. – Минск: БГУ, 2013. – 59 с.
12. Мейрембеков К.А. Интерактивный электронный учебник. Алгебра-1. Алматы: КазНУ, 2007
13. KANT group (algebra and number theory) [Электронный ресурс]. Режим доступа – <http://page.math.tu-berlin.de/~kant/#kant>
14. GAP - Groups, Algorithms, Programming - a System for Computational Discrete Algebra) [Электронный ресурс]. Режим доступа – <https://www.gap-system.org/>
15. Система компьютерной алгебры Mathematica [Электронный ресурс]. Режим доступа – <https://ru.wikipedia.org/wiki/Mathematica#Возможности>
16. Балыкина, Е.Н. Подходы к проектированию компьютерных тестов/ Е.Н. Балыкина // Информационное обеспечение исторического образования: Сб. ст. / Под. ред. В. Н. Сидорцова, А. Н. Нечухрина, Е. Н. Балыкиной. – Минск: БГУ;

- Гродно: ГрГУ, 2003. – 67–75 с. [Электронный ресурс]. Режим доступа – <http://aik-sng.ru/text/krug/7/23.html>
17. Интернет-тестирование по линейной алгебре [Электронный ресурс]. Режим доступа – <http://fen.distant.ru/test/math/3/3.7/3.7.htm>
 18. Интернет-тестирование [Электронный ресурс]. Режим доступа – <https://test.i-exam.ru/demo/training/student/test.html>
 19. Linear Algebra [Электронный ресурс]. Режим доступа – <http://immersivemath.com/ila/tableofcontents.html?>
 20. Захарова И.Г. Информационные технологии в образовании: Учеб. пособие для студ. высш. пед. учеб. заведений – М.: Издательский центр «Академия», 2003. – 192 с.

ПРИЛОЖЕНИЕ

```
    Файл «Teoriya.h»  
//-----  
#ifndef Unit1H  
#define Unit1H  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include <ComCtrls.hpp>  
#include <ImgList.hpp>  
#include <OleCtrls.hpp>  
#include <SHDocVw.hpp>  
#include <Menus.hpp>  
#include <ExtCtrls.hpp>  
//-----  
class TForm1 : public TForm  
{  
    __published:      // IDE-managed Components  
        TTreeView *TreeView1;  
        TMainMenu *MainMenu1;  
        TMenuItem *N1;  
        TMenuItem *N2;  
        TMenuItem *N3;  
        TMenuItem *N4;  
        TPanel *Panel1;  
        TWebBrowser *WebBrowser1;
```

```

void __fastcall N2Click(TObject *Sender);
void __fastcall N3Click(TObject *Sender);
void __fastcall N4Click(TObject *Sender);
void __fastcall FormCanResize(TObject *Sender, int &NewWidth, int
&NewHeight,
bool &Resize);
void __fastcall FormCreate(TObject *Sender);
void __fastcall TreeView1Change(TObject *Sender, TTreeNode *Node);
void __fastcall TreeView1MouseDown(TObject *Sender, TMouseButton Button,
TShiftState Shift, int X, int Y);
private: // User declarations
public: // User declarations
__fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif

```

Файл «Teoriya.cpp»

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
#include "Unit2.h"  
#include "Unit3.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
    TTreeNode *Tn1, *Tn2;  
  
    Tn1 = TreeView1->Items->Add(NULL, "Вектор");  
    Tn1->SelectedIndex = 1;  
  
    Tn2 = TreeView1->Items->AddChild(Tn1, "Умножение вектора на число");  
    Tn2->SelectedIndex = 1;  
  
    Tn2 = TreeView1->Items->AddChild(Tn1, "Сложение векторов");  
    Tn2->SelectedIndex = 1;  
  
    Tn2 = TreeView1->Items->AddChild(Tn1, "Умножение векторов");  
    Tn2->SelectedIndex = 1;  
  
    Tn1 = TreeView1->Items->Add(NULL, "Матрица");  
    Tn1->SelectedIndex = 1;  
  
    Tn2 = TreeView1->Items->AddChild(Tn1, "Умножение матрицы на число");  
    Tn2->SelectedIndex = 1;  
  
    Tn2 = TreeView1->Items->AddChild(Tn1, "Умножение вектора на матрицу");  
    Tn2->SelectedIndex = 1;  
  
    Tn2 = TreeView1->Items->AddChild(Tn1, "Сложение матриц");  
    Tn2->SelectedIndex = 1;  
}
```

```

    Tn2 = TreeView1->Items->AddChild(Tn1, "Умножение матриц");
    Tn2->SelectedIndex = 1;
}
//-----

void __fastcall TForm1::N2Click(TObject *Sender)
{
    Form2->Show();
    Form1->Hide();
}
//-----

void __fastcall TForm1::N3Click(TObject *Sender)
{
    Form3->Show();
}
//-----

void __fastcall TForm1::N4Click(TObject *Sender)
{
    Close();
}
//-----

void __fastcall TForm1::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
    Resize = false;
}
//-----

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    WebBrowser1->Navigate(L"file:///W:/diplom/zastavka.jpg");
}
//-----

void __fastcall TForm1::TreeView1Change(TObject *Sender, TTreeNode *Node)
{
    for (int i=0;i<TreeView1->Items->Count;i++)
        TreeView1->Items->Item[i]->Expand(0) ; //развернутый просмотр дерева в
компоненте TreeView
}
//-----

```

```
void __fastcall TForm1::TreeView1MouseDown(TObject *Sender, TMouseButton
Button,
    TShiftState Shift, int X, int Y)
{
    if(TreeView1->GetNodeAt(X, Y) == NULL) return;
    TreeView1->Selected = TreeView1->GetNodeAt(X, Y);
    TTreeNode *tmp_node = TreeView1->Selected;
    WideString s = "file:///W:/diplom/" + tmp_node->Text+".bmp";
    WebBrowser1->Navigate(s);
}
//-----
```


Файл «control.h»

```
//-----  
  
#ifndef Unit2H  
#define Unit2H  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include <Grids.hpp>  
#include <ComCtrls.hpp>  
//-----  
class TForm2 : public TForm  
{  
    __published: // IDE-managed Components  
        TPageControl *PageControl1;  
        TTabSheet *TabSheet1;  
        TTabSheet *TabSheet2;  
        TTabSheet *TabSheet3;  
        TTabSheet *TabSheet4;  
        TTabSheet *TabSheet5;  
        TStringGrid *StringGrid6;  
        TTabSheet *TabSheet6;  
        TTabSheet *TabSheet7;  
        TButton *Button3;  
        TStringGrid *StringGrid7;  
        TLabel *Label1;  
        TStringGrid *StringGrid1;  
        TStringGrid *StringGrid2;  
        TStringGrid *StringGrid3;  
        TStringGrid *StringGrid4;  
        TButton *Button2;  
        TButton *Button1;  
        TStringGrid *StringGrid9;  
        TButton *Button4;  
        TStringGrid *StringGrid10;  
        TStringGrid *StringGrid11;  
        TStringGrid *StringGrid12;  
        TStringGrid *StringGrid13;  
        TButton *Button5;  
        TStringGrid *StringGrid14;  
        TStringGrid *StringGrid15;
```

TButton *Button6;
TStringGrid *StringGrid16;
TStringGrid *StringGrid18;
TButton *Button7;
TStringGrid *StringGrid17;
TStringGrid *StringGrid19;
TStringGrid *StringGrid20;
TStringGrid *StringGrid21;
TStringGrid *StringGrid22;
TButton *Button8;
TStringGrid *StringGrid23;
TStringGrid *StringGrid24;
TStringGrid *StringGrid25;
TStringGrid *StringGrid26;
TStringGrid *StringGrid27;
TStringGrid *StringGrid28;
TLabel *Label5;
TLabel *Label6;
TLabel *Label7;
TLabel *Label8;
TLabel *Label9;
TLabel *Label10;
TLabel *Label11;
TLabel *Label12;
TLabel *Label13;
TLabel *Label14;
TLabel *Label15;
TLabel *Label16;
TLabel *Label17;
TLabel *Label18;
TLabel *Label19;
TLabel *Label20;
TLabel *Label3;
TLabel *Label2;
TEdit *Edit1;
TEdit *Edit2;
TLabel *Label4;
TLabel *Label21;
TLabel *Label22;
TLabel *Label23;
TLabel *Label24;
TLabel *Label25;
TLabel *Label26;
TLabel *Label27;

```

TLabel *Label28;
TLabel *Label29;
TLabel *Label30;
TLabel *Label31;
TLabel *Label32;
TLabel *Label33;
TLabel *Label34;
TLabel *Label35;
void __fastcall Button1Click(TObject *Sender);
void __fastcall FormCanResize(TObject *Sender, int &NewWidth, int
&NewHeight,
    bool &Resize);
void __fastcall FormActivate(TObject *Sender);
void __fastcall Button3Click(TObject *Sender);
void __fastcall Button4Click(TObject *Sender);
void __fastcall Button5Click(TObject *Sender);
void __fastcall Button6Click(TObject *Sender);
void __fastcall Button7Click(TObject *Sender);
void __fastcall Button8Click(TObject *Sender);
void __fastcall Button2Click(TObject *Sender);
void __fastcall FormCreate(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm2(TComponent* Owner);
};
//-----
extern PACKAGE TForm2 *Form2;
//-----
#endif

```

Файл «control.cpp»

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit2.h"  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm2 *Form2;  
//-----  
__fastcall TForm2::TForm2(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
void __fastcall TForm2::Button1Click(TObject *Sender)  
{  
    Form1->Show();  
    Form2->Hide();  
    for (int i=0; i < StringGrid1->RowCount; i++) {  
        StringGrid3->Cols[i]->Clear();  
    }  
}  
//-----  
  
void __fastcall TForm2::FormCanResize(TObject *Sender, int &NewWidth,  
    int &NewHeight, bool &Resize)  
{  
    Resize = false;  
}  
//-----  
  
void __fastcall TForm2::FormActivate(TObject *Sender)  
{  
    int i, j;  
    //УМНОЖЕНИЕ МАТРИЦ  
    int n, m, l;  
  
    //УМНОЖЕНИЕ ВЕКТОРА НА ЧИСЛО  
    //номер столбца и строки
```

```

for (i=1; i<StringGrid6->ColCount; i++) {
    StringGrid6->Cells[i][0] = i;
    StringGrid7->Cells[i][0] = i;
    StringGrid24->Cells[i][0] = i;
}
StringGrid6->Cells[0][1] = 1;
StringGrid7->Cells[0][1] = 1;
StringGrid24->Cells[0][1] = 1;

//заполнение рандомными числами
for (i=1; i < StringGrid6->ColCount; i++){
    StringGrid6->Cells[i][1] = IntToStr(rand() % 20 - 10);
}
Edit1->Text = IntToStr(rand() % 20 - 10);

//УМНОЖЕНИЕ МАТРИЦЫ НА ЧИСЛО
//номера строк и столбцов
for (i=1; i<StringGrid9->ColCount; i++) {
    StringGrid9->Cells[0][i] = i;
    StringGrid10->Cells[0][i] = i;
    StringGrid25->Cells[0][i] = i;
}
for (i=1; i<StringGrid9->RowCount; i++) {
    StringGrid9->Cells[i][0] = i;
    StringGrid10->Cells[i][0] = i;
    StringGrid25->Cells[0][i] = i;
}
//заполнение рандомными числами
for (i=1; i < StringGrid9->ColCount; i++) {
    for (j=1; j < StringGrid9->RowCount; j++){
        StringGrid9->Cells[i][j] = IntToStr(rand() % 20 - 10);
    }
}
Edit2->Text = IntToStr(rand() % 20 - 10);

//СЛОЖЕНИЕ ВЕКТОРОВ
//номер столбца и строки
for (i=1; i<StringGrid11->ColCount; i++) {
    StringGrid11->Cells[i][0] = i;
    StringGrid12->Cells[i][0] = i;
    StringGrid13->Cells[i][0] = i;
    StringGrid26->Cells[i][0] = i;
}
StringGrid11->Cells[0][1] = 1;

```

```

StringGrid12->Cells[0][1] = 1;
StringGrid13->Cells[0][1] = 1;
StringGrid26->Cells[0][1] = 1;
//заполнение рандомными числами
for (i=1; i < StringGrid11->ColCount; i++){
    StringGrid11->Cells[i][1] = IntToStr(rand() % 20 - 10);
    StringGrid12->Cells[i][1] = IntToStr(rand() % 20 - 10);
}

```

//СЛОЖЕНИЕ МАТРИЦ

```

//номера строк и столбцов
for (i=1; i<StringGrid16->ColCount; i++) {
    StringGrid14->Cells[0][i] = i;
    StringGrid15->Cells[0][i] = i;
    StringGrid16->Cells[0][i] = i;
    StringGrid27->Cells[0][i] = i;
}
for (i=1; i<StringGrid16->RowCount; i++) {
    StringGrid14->Cells[i][0] = i;
    StringGrid15->Cells[i][0] = i;
    StringGrid16->Cells[i][0] = i;
    StringGrid27->Cells[i][0] = i;
}
//заполнение рандомными числами
for (i=1; i < StringGrid14->ColCount; i++) {
    for (j=1; j < StringGrid14->RowCount; j++){
        StringGrid14->Cells[i][j] = IntToStr(rand() % 20 - 10);
        StringGrid15->Cells[i][j] = IntToStr(rand() % 20 - 10);
    }
}

```

//УМНОЖЕНИЕ ВЕКТОРОВ

```

//номер столбца и строки
for (i=1; i<StringGrid18->ColCount; i++) {
    StringGrid17->Cells[i][0] = i;
    StringGrid18->Cells[i][0] = i;
    StringGrid19->Cells[i][0] = i;
    StringGrid28->Cells[i][0] = i;
}
for (j=1; j<StringGrid17->RowCount; j++) {
    StringGrid17->Cells[0][j] = j;
    StringGrid18->Cells[0][j] = j;
    StringGrid19->Cells[0][j] = j;
    StringGrid28->Cells[0][i] = i;
}

```

```

}
//заполнение рандомными числами
for (i=1; i < StringGrid17->RowCount; i++){
    StringGrid17->Cells[1][i] = IntToStr(rand() % 20 - 10);
}
for (i=1; i < StringGrid18->ColCount; i++){
    StringGrid18->Cells[i][1] = IntToStr(rand() % 20 - 10);
}

//УМНОЖЕНИЕ ВЕКТОРА НА МАТРИЦУ
//номер столбца и строки
for (i=1; i<StringGrid20->ColCount; i++) {
    StringGrid20->Cells[i][0] = i;
    StringGrid21->Cells[i][0] = i;
    StringGrid22->Cells[i][0] = i;
    StringGrid23->Cells[i][0] = i;
}
for (j=1; j<StringGrid21->RowCount; j++) {
    StringGrid20->Cells[0][j] = j;
    StringGrid21->Cells[0][j] = j;
    StringGrid22->Cells[0][j] = j;
    StringGrid23->Cells[0][j] = j;
}
//заполнение рандомными числами
for (i=1; i < StringGrid20->ColCount; i++){
    StringGrid20->Cells[i][1] = IntToStr(rand() % 20 - 10);
}
for (i=1; i < StringGrid21->ColCount; i++){
    for (j=1; j < StringGrid21->RowCount; j++) {
        StringGrid21->Cells[i][j] = IntToStr(rand() % 20 - 10);
    }
}

//УМНОЖЕНИЕ МАТРИЦ
//кол-во строк первой, третьей и четвертой матриц
n = random(3);
StringGrid1->RowCount = n+3;
StringGrid3->RowCount = n+3;
StringGrid4->RowCount = n+3;

for (i=1; i<n+3; i++) {
    StringGrid1->Cells[0][i] = i;
    StringGrid3->Cells[0][i] = i;
    StringGrid4->Cells[0][i] = i;
}

```

```

}

//кол-во столбцов первой матрицы и кол-во строк второй матрицы
m = random(3);
StringGrid1->ColCount = m+3;
StringGrid2->RowCount = m+3;

for (i=1; i<m+3; i++) {
    StringGrid1->Cells[i][0] = i;
    StringGrid2->Cells[0][i] = i;
}

//кол-во столбцов второй, третьей и четвертой матриц
l = random(3);
StringGrid2->ColCount = l+3;
StringGrid3->ColCount = l+3;
StringGrid4->ColCount = l+3;

for (i=1; i<l+3; i++) {
    StringGrid2->Cells[i][0] = i;
    StringGrid3->Cells[i][0] = i;
    StringGrid4->Cells[i][0] = i;
}

//устанавливаем размер матрицы 1
StringGrid1->Height = 26*(n+3);
StringGrid1->Width = 30*(m+3);

//размер матрицы 2
StringGrid2->Height = 26*(m+3);
StringGrid2->Width = 30*(l+3);

//размер матрицы 3
StringGrid3->Height = 26*(n+3);
StringGrid3->Width = 30*(l+3);

//двигаем матрицу B в зависимости от матрицы A
StringGrid2->Left = 225+28*(m+3)+100;

Label2->Top = 96+(26*(n+2))/2;
Label3->Left = 152+66*(m+3)+25;
Label3->Top = 96+(26*(m+2))/2;
// Label33->Top = 70+(26*(m+3))/2;

```



```

//заполнение рандомными числами
for (i=1; i < StringGrid1->ColCount; i++){
    for (j=1; j < StringGrid1->RowCount; j++) {
        StringGrid1->Cells[i][j] = IntToStr(rand() % 20 - 10);
    }
}

for (i=1; i < StringGrid2->ColCount; i++){
    for (j=1; j < StringGrid2->RowCount; j++) {
        StringGrid2->Cells[i][j] = IntToStr(rand() % 20 - 10);
    }
}
}
//-----

void __fastcall TForm2::Button3Click(TObject *Sender)
{
int i;
    for (i = 1; i <= StringGrid6->ColCount-1; i++) {
        StringGrid7->Cells[i][1] = Edit1->Text.ToDouble( ) * StringGrid6-
>Cells[i][1].ToDouble();
    }

//проверка ответа
for (i = 1; i <= StringGrid7->ColCount-1; i++) {

    {
        if(StringGrid24->Cells[i][1] != StringGrid7->Cells[i][1])
        {
            StringGrid24->Canvas->Pen->Color=clRed; //Задаем
цвет
            int x1=StringGrid24->CellRect(i,1).left; //определяем
ячейку
            int y1=StringGrid24->CellRect(i,1).top;
            int x2=StringGrid24->CellRect(i,1).right;
            int y2=StringGrid24->CellRect(i,1).bottom;
            StringGrid24->Canvas->Rectangle(x1,y1,x2,y2);
            StringGrid24->Canvas->Pen->Color=clBlack; //Задаем
цвет
        }
    }
}
}
//-----

```

```

void __fastcall TForm2::FormCreate(TObject *Sender)
{
    for (int i = 0; i <7; i++) {
        PageControl1->Pages[i]->Brush->Color=clWhite;
    }
}
//-----

void __fastcall TForm2::Button4Click(TObject *Sender)
{
    int i, j;
    for (i=1; i < StringGrid10->ColCount; i++) {
        for (j=1; j < StringGrid10->RowCount; j++){
            StringGrid10->Cells[i][j] = Edit2->Text.ToDouble( ) * StringGrid9-
>Cells[i][j].ToDouble( );
        }
    }
    //проверка ответа
    for (i = 1; i <= StringGrid25->ColCount; i++) {
        for (j = 1; j <= StringGrid25->RowCount; j++) {
            if(StringGrid25->Cells[j][i] != StringGrid10->Cells[j][i])
            {
                StringGrid25->Canvas->Pen->Color=clRed;    //Задаем цвет
                int x1=StringGrid25->CellRect(j,i).left; //определяем ячейку
                int y1=StringGrid25->CellRect(j,i).top;
                int x2=StringGrid25->CellRect(j,i).right;
                int y2=StringGrid25->CellRect(j,i).bottom;
                StringGrid25->Canvas->Rectangle(x1,y1,x2,y2);
                StringGrid25->Canvas->Pen->Color=clBlack; //Задаем цвет
            }
        }
    }
}
//-----

void __fastcall TForm2::Button5Click(TObject *Sender)
{
    int i;
    for (i = 1; i <= StringGrid13->ColCount-1; i++) {
        StringGrid13->Cells[i][1] = StringGrid11->Cells[i][1].ToDouble( ) +
StringGrid12->Cells[i][1].ToDouble( );
    }
    //проверка ответа
}

```

```

for (i = 1; i <= StringGrid13->ColCount; i++) {
    if(StringGrid26->Cells[i][1] != StringGrid13->Cells[i][1])
    {
        StringGrid26->Canvas->Pen->Color=clRed;    //Задаем цвет
        int x1=StringGrid26->CellRect(i,1).left; //определяем ячейку
        int y1=StringGrid26->CellRect(i,1).top;
        int x2=StringGrid26->CellRect(i,1).right;
        int y2=StringGrid26->CellRect(i,1).bottom;
        StringGrid26->Canvas->Rectangle(x1,y1,x2,y2);
        StringGrid26->Canvas->Pen->Color=clBlack;    //Задаем цвет
    }
}
}
//-----

void __fastcall TForm2::Button6Click(TObject *Sender)
{
    int i, j;
    for (i=1; i < StringGrid16->ColCount; i++) {
        for (j=1; j < StringGrid16->RowCount; j++){
            StringGrid16->Cells[i][j] = StringGrid14->Cells[i][j].ToDouble( ) +
StringGrid15->Cells[i][j].ToDouble( );
        }
    }
    //проверка ответа
    for (i = 1; i <= StringGrid27->ColCount; i++) {
        for (j = 1; j <= StringGrid27->RowCount; j++){
            if(StringGrid27->Cells[i][j] != StringGrid16->Cells[i][j])
            {
                StringGrid27->Canvas->Pen->Color=clRed;    //Задаем
цвет
                int x1=StringGrid27->CellRect(i,j).left;    //определяем
ячейку
                int y1=StringGrid27->CellRect(i,j).top;
                int x2=StringGrid27->CellRect(i,j).right;
                int y2=StringGrid27->CellRect(i,j).bottom;
                StringGrid27->Canvas->Rectangle(x1,y1,x2,y2);
                StringGrid27->Canvas->Pen->Color=clBlack;    //Задаем
цвет
            }
        }
    }
}
//-----

```

```

void __fastcall TForm2::Button7Click(TObject *Sender)
{
    int i, j, k;
    for (i = 1; i < StringGrid17->RowCount; i++) {
        for (j = 1; j < StringGrid18->ColCount; j++) {
            for (k = 1; k < StringGrid17->ColCount; k++) {
                StringGrid19->Cells[j][i] = StringGrid17-
>Cells[k][i].ToDouble( ) * StringGrid18->Cells[j][k].ToDouble( );
            }
        }
    }
    //проверка ответа
    for (i = 1; i <= StringGrid28->ColCount; i++) {
        for (j = 1; j <= StringGrid28->RowCount; j++){
            if(StringGrid28->Cells[i][j] != StringGrid19->Cells[i][j])
            {
                StringGrid28->Canvas->Pen->Color=clRed; //Задаем
ЦВЕТ
                int x1=StringGrid28->CellRect(i,j).left; //определяем
ячейку
                int y1=StringGrid28->CellRect(i,j).top;
                int x2=StringGrid28->CellRect(i,j).right;
                int y2=StringGrid28->CellRect(i,j).bottom;
                StringGrid28->Canvas->Rectangle(x1,y1,x2,y2);
                StringGrid28->Canvas->Pen->Color=clBlack; //Задаем
ЦВЕТ
            }
        }
    }
}
//-----

```

```

void __fastcall TForm2::Button8Click(TObject *Sender)
{
    int i, j, k, a;
    for (i = 1; i < StringGrid20->RowCount; i++) {
        for (j = 1; j < StringGrid21->ColCount; j++) {
            a = 0;
            for (k = 1; k < StringGrid20->ColCount; k++) {
                a += StringGrid20->Cells[k][i].ToDouble( ) * StringGrid21-
>Cells[j][k].ToDouble( );
            }
            StringGrid22->Cells[j][i] = a;
        }
    }
}

```

```

    }
}
//проверка ответа
for (i = 1; i <= StringGrid23->ColCount; i++) {
    if(StringGrid23->Cells[i][1] != StringGrid22->Cells[i][1])
    {
        StringGrid23->Canvas->Pen->Color=clRed;    //Задаем цвет
        int x1=StringGrid23->CellRect(i,1).left; //определяем ячейку
        int y1=StringGrid23->CellRect(i,1).top;
        int x2=StringGrid23->CellRect(i,1).right;
        int y2=StringGrid23->CellRect(i,1).bottom;
        StringGrid23->Canvas->Rectangle(x1,y1,x2,y2);
        StringGrid23->Canvas->Pen->Color=clBlack;    //Задаем цвет
    }
}
}
//-----

void __fastcall TForm2::Button2Click(TObject *Sender)
{
int n, m, l, i, j, k, prav_otv;

    double a;

    n = StringGrid1->RowCount-1;
    m = StringGrid1->ColCount-1;
    l = StringGrid2->ColCount-1;

    for (i = 1; i <= n; i++) {
        for (j = 1; j <= l; j++) {
            a = 0;
            for (k = 1; k <= m; k++) {
                a += StringGrid1->Cells[k][i].ToDouble( ) * StringGrid2-
>Cells[j][k].ToDouble( );
            }
            StringGrid4->Cells[j][i] = a;
            //AnsiString buffer(a);
            //StringGrid3->Cells[j][i] = buffer;
        }
    }

    //проверка ответа
    // prav_otv=0;
    for (i = 1; i <= n; i++) {

```

```

for (j = 1; j <= 1; j++) {
    if(StringGrid3->Cells[j][i] != StringGrid4->Cells[j][i])
//      prav_otv++;
//      else
        {
            StringGrid3->Canvas->Pen->Color=clRed;    //Задаем цвет
            int x1=StringGrid3->CellRect(j,i).left; //определяем ячейку
            int y1=StringGrid3->CellRect(j,i).top;
            int x2=StringGrid3->CellRect(j,i).right;
            int y2=StringGrid3->CellRect(j,i).bottom;
            StringGrid3->Canvas->Rectangle(x1,y1,x2,y2);
            StringGrid3->Canvas->Pen->Color=clBlack;    //Задаем цвет
        }
    }
}
//-----

```

Файл «spravka.h»

```
//-----  
  
#ifndef Unit3H  
#define Unit3H  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include <ExtCtrls.hpp>  
//-----  
class TForm3 : public TForm  
{  
    __published:      // IDE-managed Components  
        TMemo *Memo1;  
        void __fastcall FormCreate(TObject *Sender);  
        void __fastcall FormCanResize(TObject *Sender, int &NewWidth, int  
&NewHeight,  
        bool &Resize);  
private:      // User declarations  
public:      // User declarations  
    __fastcall TForm3(TComponent* Owner);  
};  
//-----  
extern PACKAGE TForm3 *Form3;  
//-----  
#endif
```

Файл «spravka.cpp»

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit3.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm3 *Form3;  
//-----  
__fastcall TForm3::TForm3(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
void __fastcall TForm3::FormCreate(TObject *Sender)  
{  
    Memo1->Lines->LoadFromFile("spravka.txt");  
}  
//-----  
  
void __fastcall TForm3::FormCanResize(TObject *Sender, int &NewWidth,  
    int &NewHeight, bool &Resize)  
{  
    Resize = false;  
}  
//-----
```