Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики и фундаментальной информатики
Базовая кафедра вычислительных и информационных технологий

УТВЕРЖДАЮ
Заведующий кафедрой
________ / В.В. Шайдуров
«16» июня 2017 г.

БАКАЛАВРСКАЯ РАБОТА

Направление 02.03.01 Математика и компьютерные науки

КОГНИТИВНАЯ МОДЕЛЬ ДИАГНОСТИКИ ХОДА РЕШЕНИЯ АЛГОРИТМИЧЕСКИХ ЗАДАЧ

Научный руководитель
доктор педагогических наук,
профессор ________ /Н.И. Пак
16.06.2017

Выпускник ________ / М.Е. Бальхеева
16.06.2017

Красноярск 2017

РЕФЕРАТ

Дипломная работа по теме «Когнитивная модель диагностики хода решения алгоритмических задач» содержит 28 страниц текста, 2 приложения, 4 использованных источника.

КОМПЬЮТЕРНЫЙ ТРЕНАЖЕР, ИЛЛЮСТРИРУЮЩАЯ ПРОГРАММА, ОБРАТНАЯ МАТРИЦА, КОММЕНТИРОВАНИЕ РЕШЕНИЯ, МЕНТАЛЬНАЯ МОДЕЛЬ, КОГНИТИВНЫЙ ПОДХОД, ДИАГНОСТИКА.

Цель работы – обосновать и создать модель компьютерного тренажера по решению алгебраических задач на основе когнитивного подхода, способствующей повышению эффективности самообразовательной деятельности студента.

В результате исследований обоснована необходимость создания компьютерных тренажеров по решению задач, разработана модель тренажера на основе когнитивного подхода, создана программная среда для реализации модели, проведен анализ полезности тренажера и разработаны методические указания по его разработке и использованию.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время наряду с бурным развитием дистанционных образовательных технологий отмечается существенная нехватка наработок, связанных с автоматизацией интеллектуальных средств обучения. Если концепции и технологии создания и эксплуатации виртуальных лабораторных комплексов, лабораторных практикумов удаленного доступа и мультимедийных демонстрационных лабораторных стендов хорошо развиты, то интерактивные компьютерные тренажеры, задачей которых является развитие различных навыков (решение задач, обучение быстрому чтению и т.д.) обделены вниманием.

Компьютерные тренажеры очень разнообразны и применяются в самых разных областях. Принцип, на котором основано большинство компьютерных тренажеров - моделирование реальности. Применительно к математическим тренажерам это означает создание некоторого подобия реального процесса решения поставленной задачи, в котором пользователь ведет себя так же, как и при решении этой задачи на бумаге. Очевидно, что чем более «похожа» созданная модель на свой реальный прототип и чем ближе ее поведение к реальности, тем лучше компьютерный тренажер. Пользователь практикуется в решении задачи, имея дело с их компьютерной электронной аналогией.

Исторически тренажерные технологии возникли и получили наибольшее развитие там, где ошибки при обучении на реальных объектах могут привести к чрезвычайным последствиям, а их устранение - к большим финансовым затратам: в военном деле, медицине, атомной энергетике, авиации и космосе, высокотехнологичном производстве.

Цель исследования – обоснование и создание модели компьютерного тренажера по решению алгебраических задач на основе когнитивного подхода, способствующей повышению эффективности самообразовательной деятельности студента.

Объект исследования – процесс обучения студентов решению математических задач.

Предмет исследования – ментальная модель компьютерного тренажера по решению алгебраических задач, на примере темы "Обратная матрица".

Задачи:

1. обосновать необходимость создания компьютерных тренажеров по решению задач;
2. разработать модель тренажера на основе когнитивного подхода;
3. создать программную среду для реализации модели;
4. провести анализ полезности тренажера.

1 Теоретические основания к проектированию компьютерных тренажеров на основе когнитивного подхода

1.1 О необходимости создания компьютерных тренажеров по решению задач

Для повышения эффективности самообразовательной деятельности студентов по решению математических задач возникает необходимость создания компьютерных тренажеров, моделирующих обучающие функции преподавателя.

Главным принципом работы компьютерного тренажера является многократное повторение конкретных действий, при надобности, с постепенным увеличением уровня сложности. Аналогию можно подметить в тренажерах по развитию физических способностей человека.

В программах обучения и подготовки, в современных тренажерах, основанных на них, закладываются основы становления практических навыков с одновременной теоретической подготовкой. Реализация такого подхода стала возможна в связи с буйным развитием компьютерной техники и прогрессом в области создания технологий виртуальной реальности, машинного зрения, систем искусственного интеллекта и т.п. Уже сейчас на базе этих технологий разработаны многочисленные тренажеры для военного применения, позволяющие имитировать боевые действия с высочайшей детальностью в реальном времени, создано множество приложений технологии виртуальной реальности для медицины, позволяющих проводить операции электронному пациенту с высокой степенью достоверности.

По мере развития и удешевления компьютерной техники тренажерные технологии начинают проникать и в другие отрасли: хозяйство, авто- и судовождение, школьное и вузовское обучение и прочее.

Ряд научных и инженерных коллективов страны уже давно работает над применением компьютерных тренажеров (КТ) в обучении, но ситуация на рынке сохраняется: он перенасыщен различными экзаменаторами и оболочками для тестирования, а вот компьютерных тренажеров по-прежнему мало.

Целью применения КТ является привитие навыков определенного вида практической деятельности. Для компьютерных тренажеров по математическим дисциплинам круг возможностей КТ включает в себя привитие навыка решения задач различного типа, освоение работы численных алгоритмов. В целом, цель математического КТ - научить пользователя решать математические задачи.

Бесспорно, что решение любого типа задачи может быть описано с помощью определенного алгоритма, последовательности аналитических и вычислительных действий, однозначно приводящих к ответу на поставленную задачу исходя из некоторого набора изначально определенных условий задачи и начальных данных.

Функционирование КТ аналогично решению задачи на бумаге. При решении в КТ происходит сверка на каждом шаге с эталонным решением задачи, которое вычисляется автоматически тренажером. В результате проверки на правильность введенных пользователем вычисленных значений КТ осуществляет вывод корректирующей информации (советы, формулы, рекомендации). Компьютерный тренажер может работать в двух режимах: демонстрации решения задачи и тренировки решения задачи. В режиме демонстрации КТ осуществляет решение задачи по шагам, позволяя пользователю рассмотреть решение задачи с комментариями, для чего пользователю предоставляется возможность просмотра решения задачи на каждом шаге алгоритма. Если для решения определенного типа задач, рассматриваемых в КТ, разработано несколько методов, то КТ предоставляет возможность сравнения количественной и качественной оценки эффективности их использования для конкретной задачи.

Современная компьютерная технология (мультимедиа) позволяет создавать диалоговые КТ, включающие компьютерную мультипликацию, аудио и видеотехнику. Как минимум, это усилит ощущение реальности при работе с тренажером и откроет новые возможности в процессе обучения.

КТ имеет в своей основе динамическую математическую модель и поэтому наиболее активно используется для качественного обучения. Разумеется, для расчета достаточно полной математической модели требуются весьма внушительные вычислительные ресурсы. Поэтому модель следует упростить, но так, чтобы ее поведение в обусловленных рамках соответствовало реальной системе с определенной точностью – это сложная творческая задача для программиста и научного работника. Можно с уверенностью сказать, что разработка такого наукоемкого продукта занимает достаточно длительный временной период. Так объясняется относительно небольшое количество разработчиков тренажеров данного класса и ощутимая ниша на рынке подобных продуктов.

КТ с динамическими математическими моделями являются наиболее интересными и сложными программными продуктами. Есть два различных способа изготовления таких КТ. Первый заключается в написании отдельной программы для каждого отдельного КТ, второй - в использовании специального инструмента разработчика (конструктора), который позволяет многократно ускорить разработку.

В первом случае возможно достижение красивых специальных эффектов, но очень затруднена модификация КТ. Как правило, программы этого класса (и математические модели в них) очень просты, поэтому такие локальные тренажеры более распространены (что является показателем не качества, а рентабельности их производства).

При использовании конструктора разработчиком КТ должен быть не программист, а специалист конкретной области науки (математик, технолог, физик), владеющий аппаратом необходимой науки. Общее свойство большинства конструкторов - составление динамической модели из стандартных элементов, описывающих определенные объекты управления и стандартные математические операции, передаточные функции, логику.

Способы управления КТ обычно отличаются от реального, но это не следует считать недостатком, поскольку он позволяет абстрагироваться от частностей и сосредоточиться на понимании сути моделируемых явлений.

1.2 Ментальная модель компьютерного тренажера

Ментальная карта или схема - это инструмент визуального представления и записи информации, метод, альтернативный привычному линейному способу. Это особый вид творчества, который развивает наше мышление и память. В некоторых переводах могут приводиться другие названия данного метода: «интеллект-карты», «карты разума», «карты памяти». Данный способ представления информации исследовался У. Найсером, А. Хуторским, а также Т. и Б. Бьюзенами и другими авторами.

Этот метод, по сути, позволяет наглядно отобразить процесс общего системного мышления. Ментальная карта реализуется в виде схемы, на которой изображены термины, идеи, задачи, связанные ветвями, отходящими от центрального понятия идеи. Можно сказать, что структура ментальной карты имеет вид древовидной схемы. Ментальная схема состоит из вершин (объектов) и ребер, которые определяют действие. Объекты могут быть как исходными данными, так и необходимыми целями, также в качестве объекта может быть очередная ментальная схема. С помощью ментальной схемы мы можем видеть все возможные маршруты достижения цели, благодаря определенным исходным данным, что позволяет нам выбрать оптимальный.

Рисунок 1 – Схема функционирования компьютерного тренажера

На рисунке 1 представлена схема функционирования компьютерного тренажера, главными модулями которой являются: модуль «Пользователь», модуль «Условие задачи» и модуль «Тренировка». Данные модули взаимодействуют между собой с помощью модулей-действий, таких как: «Устройства взаимодействия», «Рабочая среда», «Обработка событий».

2 Разработка программной среды компьютерного тренажера по решению математических задач

2.1 Программная реализация модели тренажера по теме "Обратная матрица" и методические указания по ее разработке и использованию

Рассмотрим структуры данных, используемые для описания моделей сценария практического занятия для интерактивного КТ.

В первую очередь необходимо задать алгоритм работы с интерактивным тренажером. Исходя из того, что большинство алгоритмов описываются последовательностью шагов, а ход решения представляет собой направленный граф, для удобства представления алгоритма можно использовать дерево, в котором поддеревья являются подзадачами, листья – элементами списка (последовательности шагов, описывающих алгоритм), а ребра, присутствующие в графе, но отсутствующие в дереве, задаются листьями с командой типа «перейти на пункт N».

Ход решения целесообразно представлять в виде стека фазовых состояний. Здесь под фазовым состоянием понимается описание некой структуры данных, однозначно описывающей состояние алгоритма на заданном этапе его выполнения. Данная структура, в дальнейшем называемая фазовой переменной, содержит в себе все элементы, описывающие состояние алгоритма и однозначно определяющие его дальнейшее выполнение.

Использование стека фазовых состояний позволяет осуществлять переходы по дереву алгоритма вперед и назад, проводить контроль пользователя по достаточно широкому спектру параметров, предоставлять большие возможности по протоколированию работы пользователя.

Рисунок 2 – Блок-схема КТ

Наиболее важной компонентой является компонента представления фазового состояния алгоритма. Она представляет собой окно, содержащие именованные поля, в которые выводятся тренажером контрольные значения при переходе от одного узла дерева алгоритма к другому.

Таким образом, структура тренажера включает в себя блок считывания и интерпретации сценария работы, проигрыватель сценария, визуализатор на основе разработанных компонент графического представления работы алгоритма, менеджер изменения фазового состояния, блок взаимодействия с пользователем (интерфейс) и блок контроля и протоколирования.

Определение 2.1. Под обратной матрицей будем понимать такую матрицу A^{-1}, при умножении на которую исходная матрица A даёт в результате единичную матрицу E:

$$AA^{-1} = A^{-1}A = E.$$

Данная программа предназначена для выработки у студентов навыков в выполнении элементарных преобразований в процессе построения обратной матрицы. Обращение матрицы является многоэтапной задачей, в ходе выполнения которой студенту необходимо знать структуру элементов обратной матрицы, уметь вычислять алгебраические дополнения и определитель. Проверка знаний и навыков студентов является для преподавателя рутинной и трудоемкой работой. Именно такую работу удобно возложить на компьютер.

Тренажер имеет 7 блоков. В первом блоке проводится регистрация пользователя (рисунок 3). Это нужно для того, чтобы в далее могла производиться диагностика.

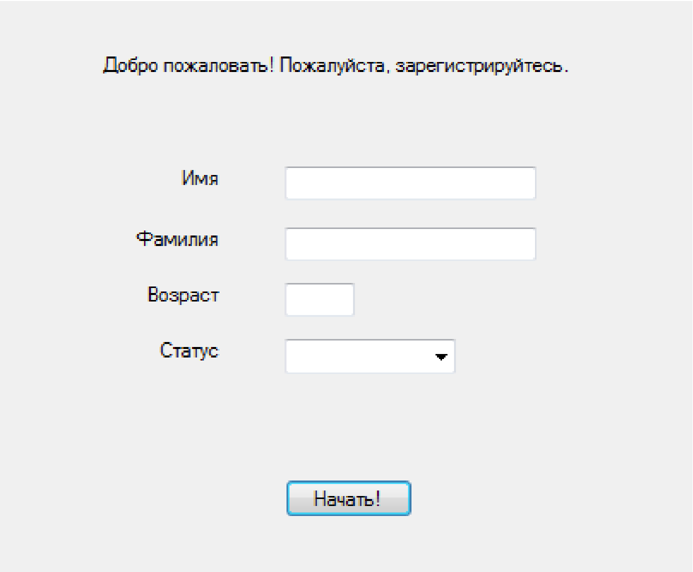

Рисунок 3 – Регистрация пользователя

Во втором блоке контроль не производится, поскольку пользователь выбирает порядок исходной матрицы (рисунок 4). Порядок может изменяться от 2 до 4.

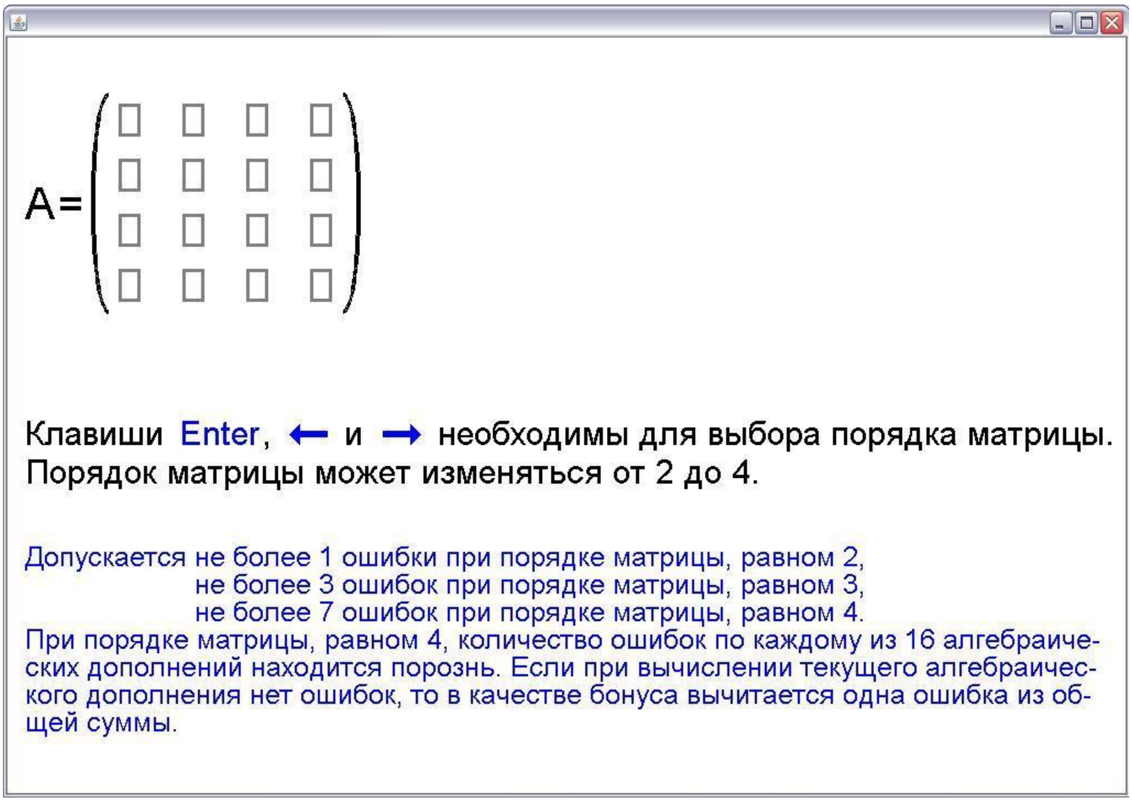

Рисунок 4 – Выбор порядка матрицы

Несмотря на простоту решения задачи для матрицы 2 порядка, она представляет интерес, поскольку алгебраические дополнения не являются определителями. Каждое алгебраическое дополнение является элементом матрицы, для которого необходимо правильно выбрать знак.

В третьем блоке необходимо выбрать шаблон элемента обратной матрицы (рисунок 5). Место для правильного вида шаблона среди неправильных видов выбирается случайным образом. Положения различных формул в квадратах выбираются случайным образом.

Рисунок 5– Выбор структуры для элемента обратной матрицы

Переход к следующему четвертому блоку производится после одного правильного щелчка мышью. В третьем блоке необходимо задать конкретный вид для каждого алгебраического дополнения (рисунок 6). Для того, чтобы выбираемые алгебраические дополнения расположить в разрядку, они распределены случайным образом на местах элементов матрицы шестого порядка.

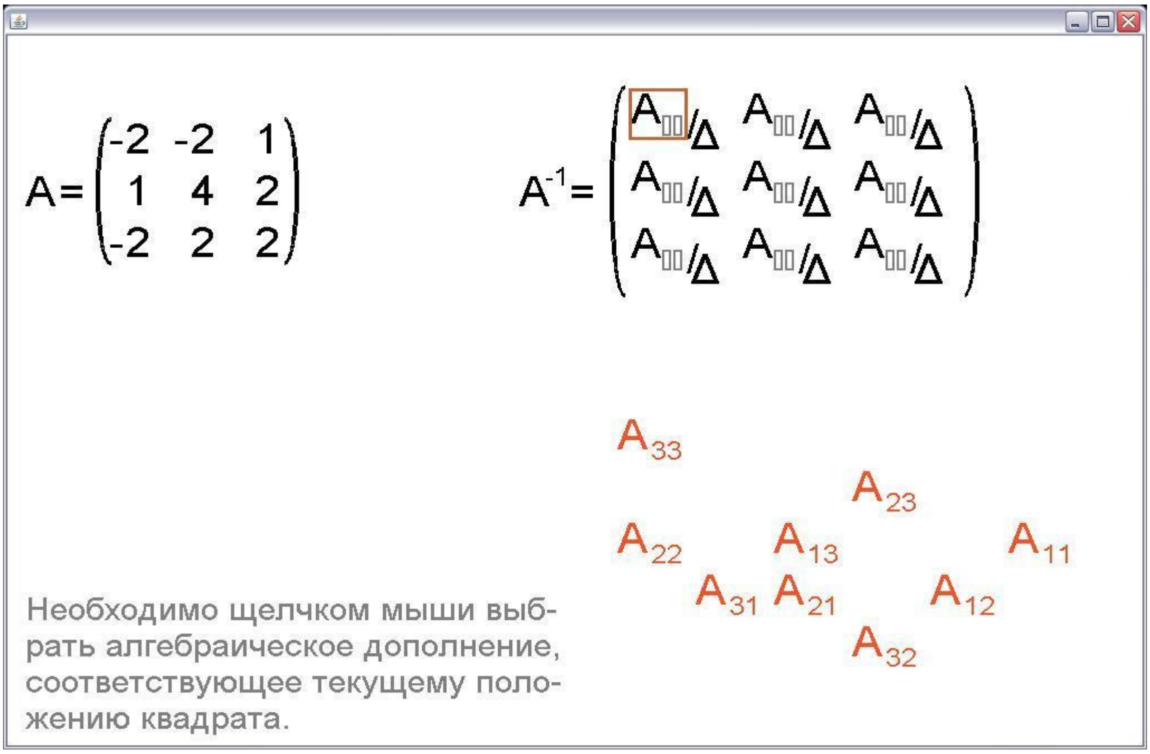

Рисунок 6 – Расстановка алгебраических дополнений в обратной матрице

Данный блок может генерировать ошибки при неправильном выборе алгебраического дополнения. На рисунке 6 видно, что на отмеченном месте должен находиться элемент A_{21}. При нажатии клавиши <Escape> счетчик ошибок увеличится на 1.

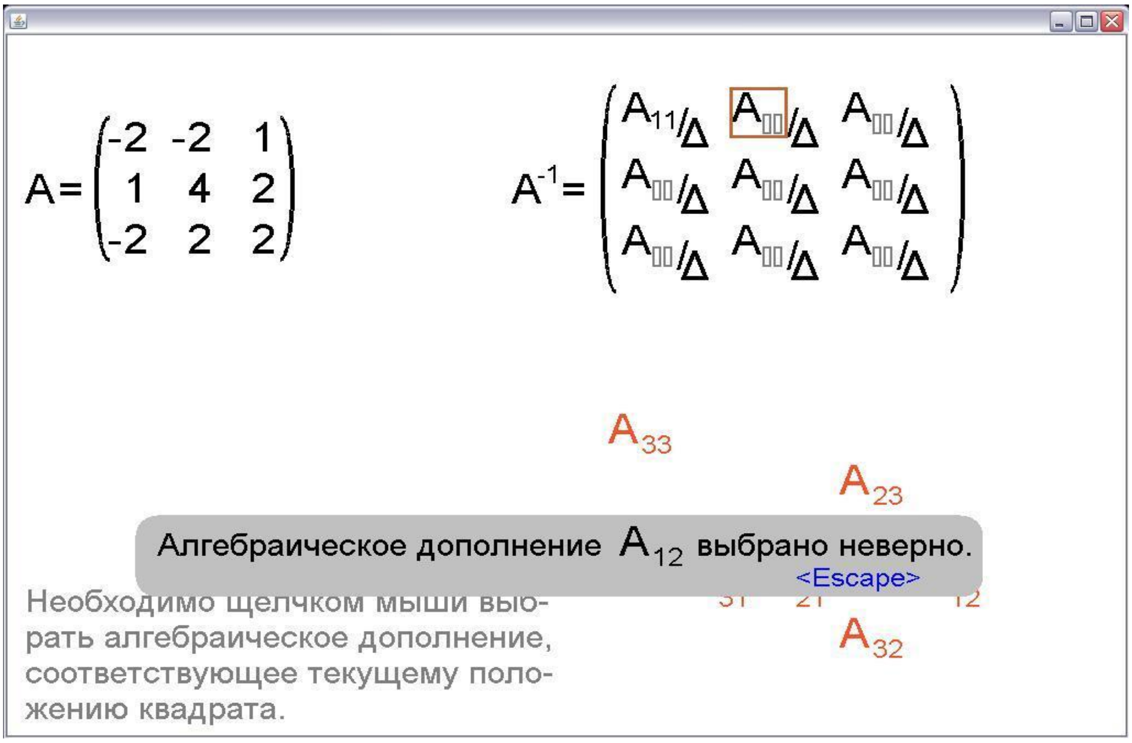

Рисунок 7 – Неверный выбор шаблона алгебраического дополнения

В пятом блоке вычисляются значения алгебраических дополнений (рисунок 8).

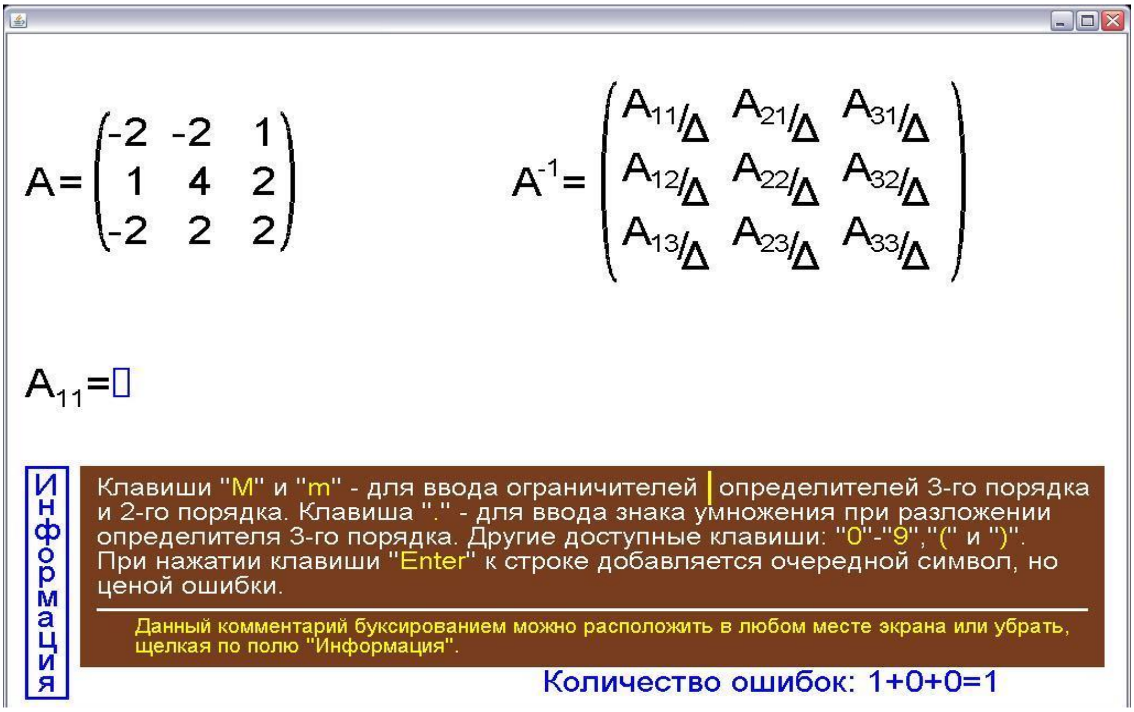

Рисунок 8 – Составляется алгебраическое дополнение

Для исключения неоднозначности в выборе действия выводятся необходимые комментарии (рисунок 9).

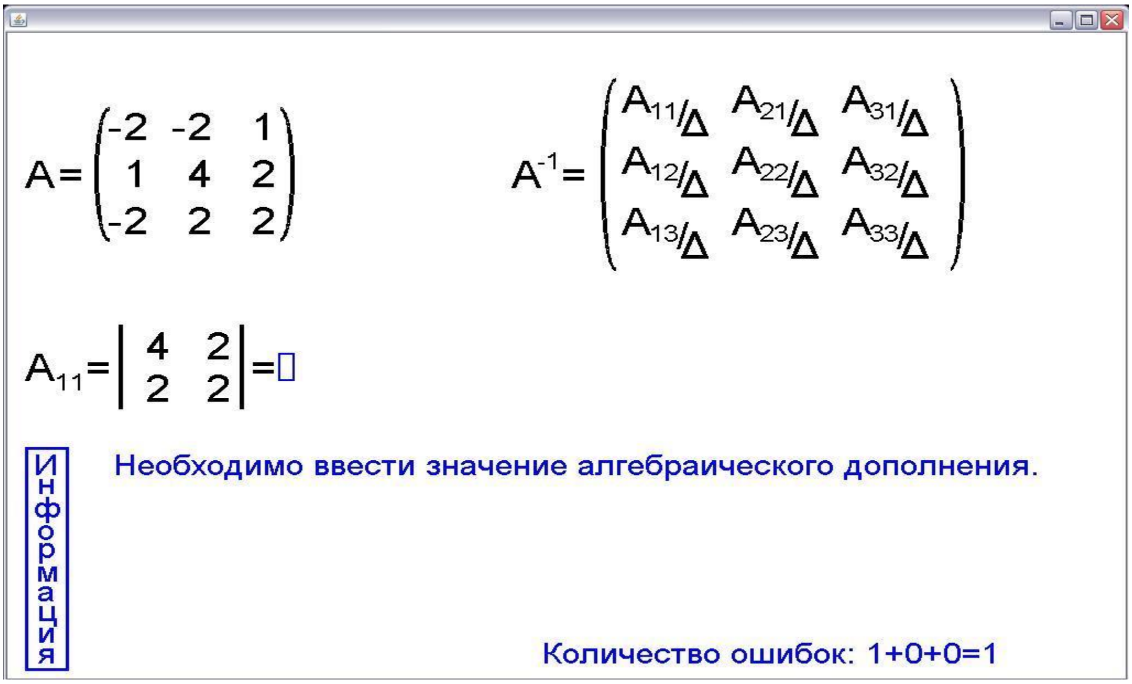

Рисунок 9 – Вычисление алгебраических дополнений

В ходе выполнения программы выводятся коментарии, исключающие возможность неправильного продолжения.

Для исключения возможной подсказки блок контроля реагирует на пропущенные символы, т.е. пропущенный знак перед определителем приводит к ошибке (рисунок 10).

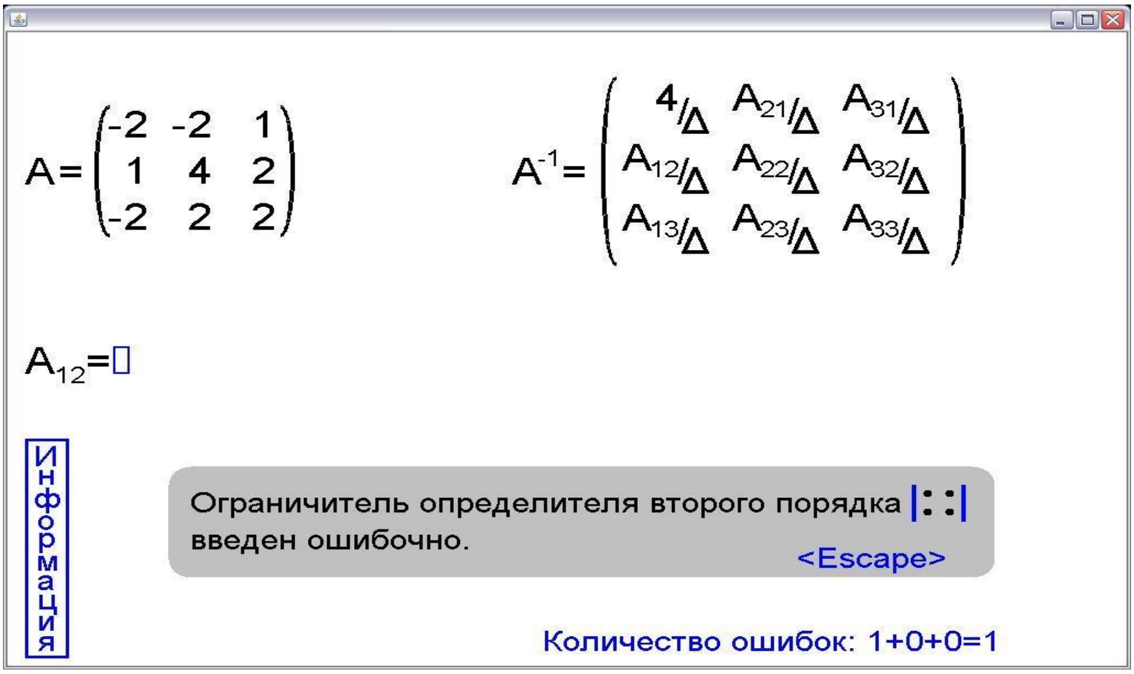

Рисунок 10 – Построение алгебраического дополнения в данном случае должно начинаться со знака ”-”

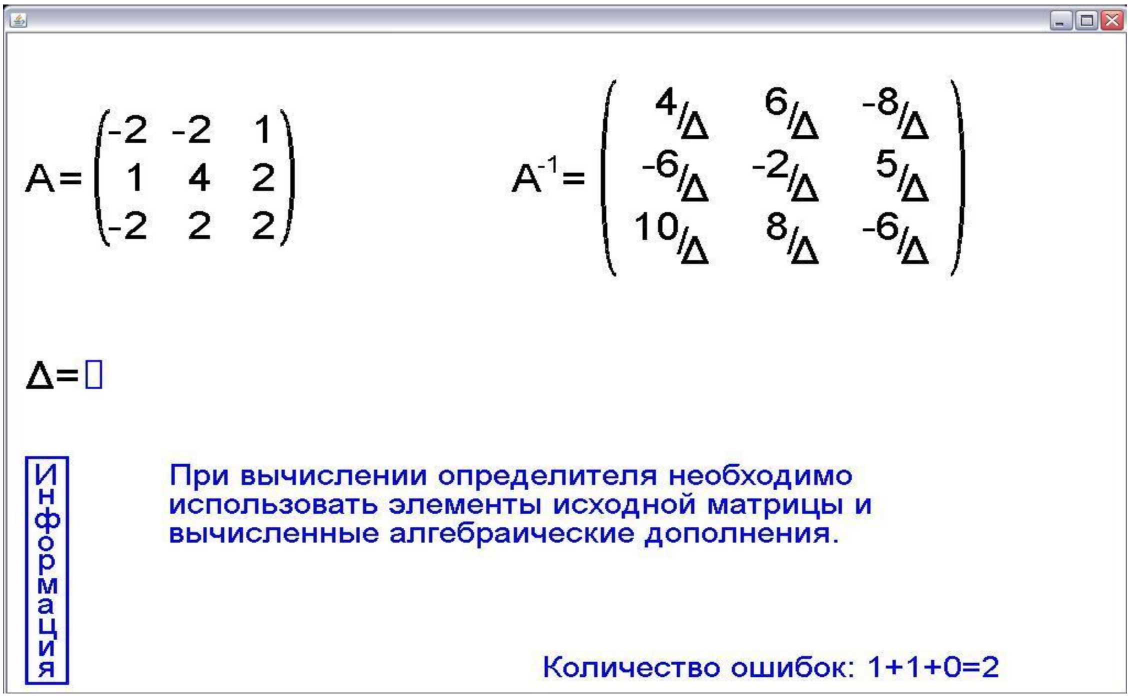

Рисунок 11 – Вычисление определителя

В шестом блоке вычислется определитель исходной матрицы (рисунок 11).

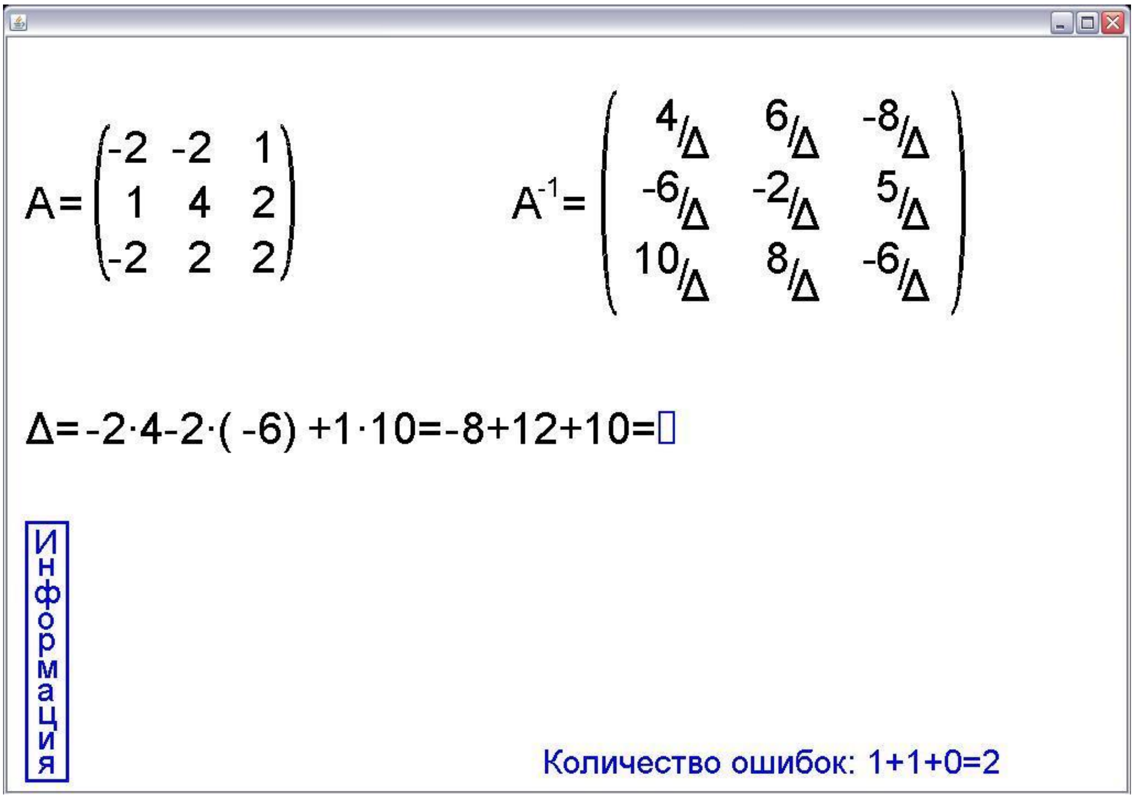

Рисунок 12 – Вычисление определителя

Значение определителя вычисляется как сумма парных произведений, в каждом из которых одним сомножителем является элемент исходной матрицы, другим – соответветствующее данному элементу алгебраическое дополнение. На рисунке 12 приведено разложение определителя по первой строке, но контролирующий блок допускает разложение определителя по любому ряду. Для ”оживления” процесса контроля после ввода пользователем окончательного значения определителя запускается поток, в котором последовательно после некоторой временной изменяются 3 параметра. Первый из них посимвольно уменьшает длину строки до тех пор, пока в ней не останется только окончательное значение определителя. Второй параметр обеспечивает изменение координат экземпляров значения определителя, в следствие чего организуется их движение, пока числа не займут места в структуре обратной матрицы. На рисунке 13 зафиксированы промежуточные состояния экземпляров значений определителей. После ввода значения определителя запускается поток, в котором каждый из 9 экземпляров числа ”14”, двигаясь от исходного положения до положения символа Δ, заменяет последний. Третий параметр обеспечивает перемещения знаков ”-”, если значения определителей отрицательны. Знаки “-”, подходя к соответстующим элементам обратной матрицы, инвертируют их знаки.

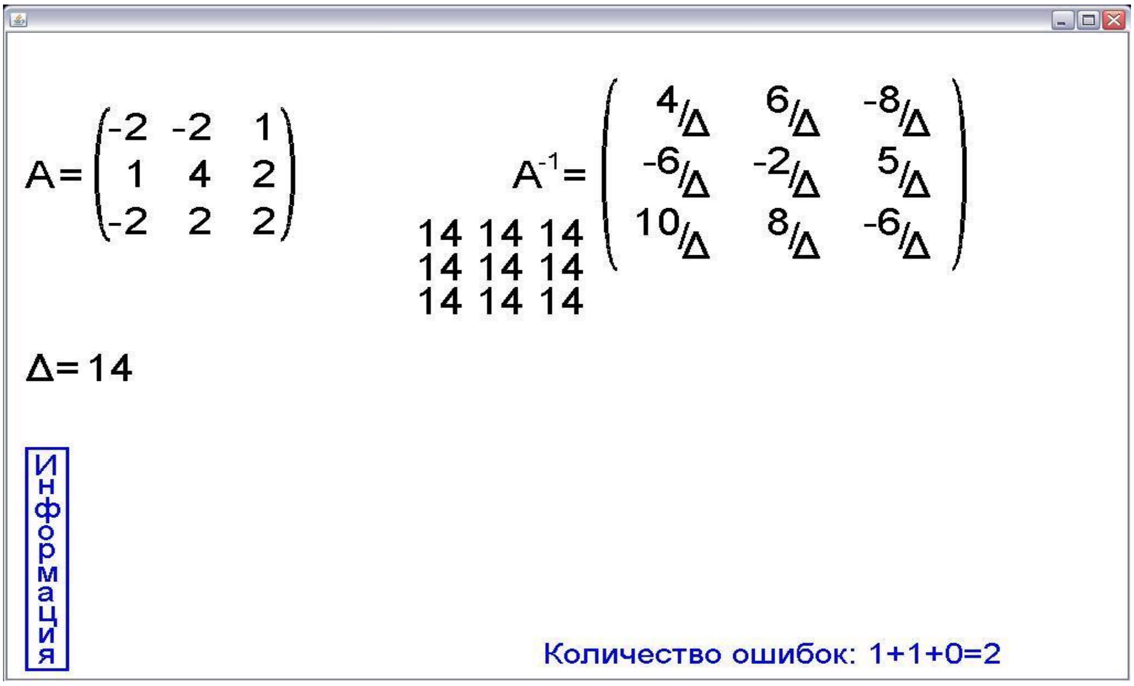

Рисунок 13 – Анимация после ввода определителя

В седьмом блоке призводится сокращение дробей, определяющих элементы обратной матрицы (рисунок 14). Рамка последовательно охватывает каждый ее элемент даже в том случае, когда элемент невозможно преобразовать. В таких вариантах наибольшим общим делителем является 1.

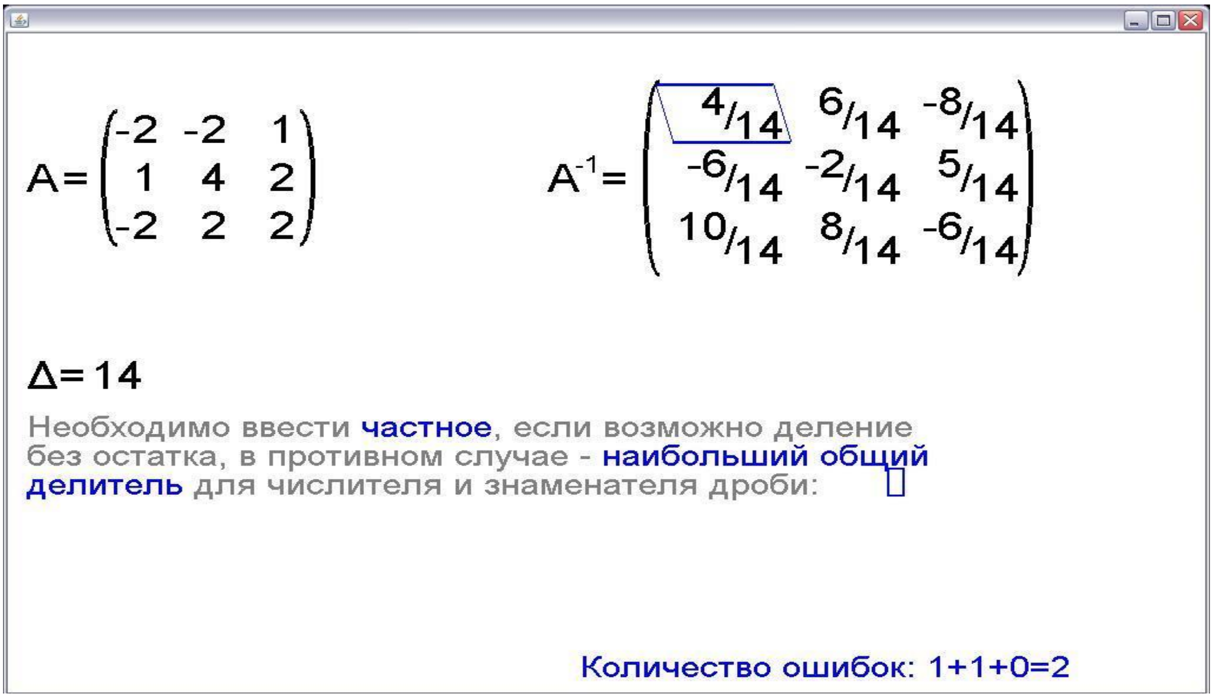

Рисунок 14 – Сокращение чисел, расположенных в числителе и знаменателе дробей

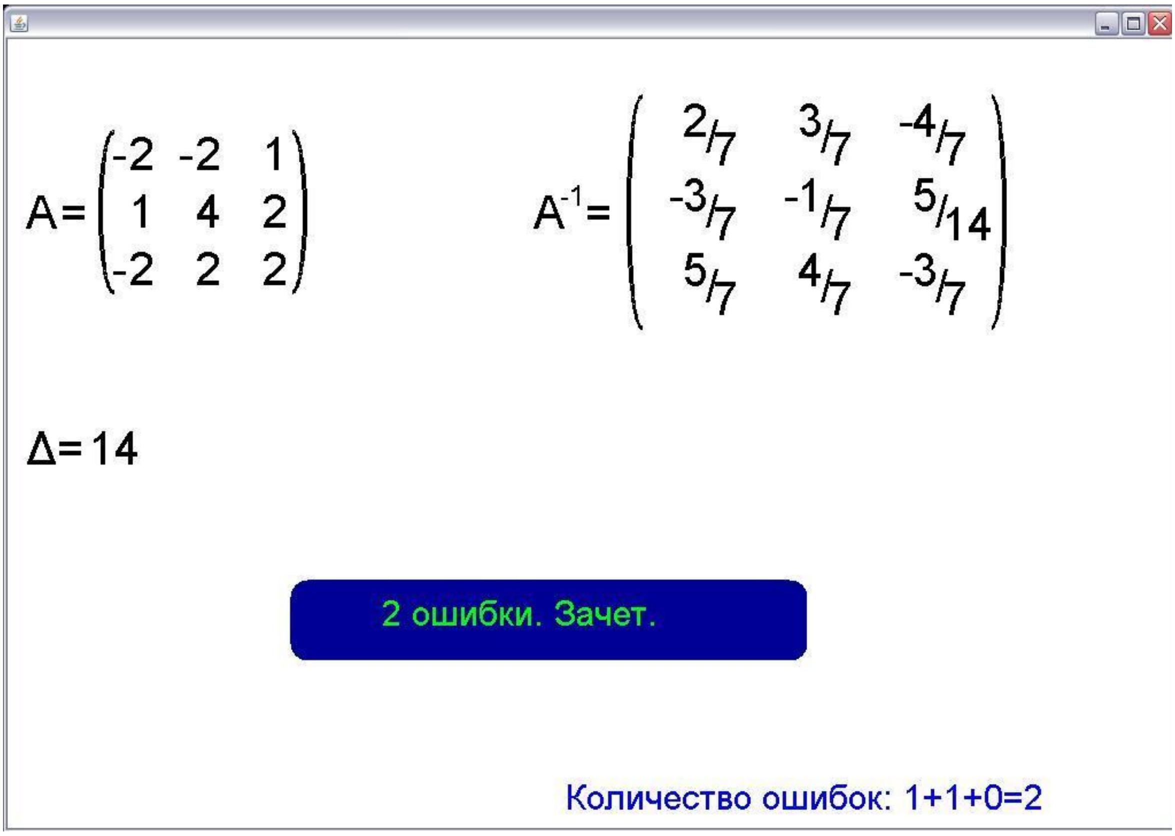

Рисунок 15 – Вывод заключительного комментария

Когда выполнены сокращения для последнего элемента обратной матрицы тренажер завершает работу с выводом заключительного комментария (рисунок 15).

Результат можно впоследствии увидеть, если открыть текстовый файл Result.txt, где отображается фамилия и количество верных ответов.

2.2 Апробация тренажера

В работе были продиагностированы шесть человек в возрасте от 20 до 24 лет. Каждый реципиент запускал данный тренажер на компьютере и приступал к построению обратных матриц, начиная с малой степени сложности. После того, как количество ошибок становилось меньше или равному допустимому значению, человек переходил на новый уровень сложности. Так, для получения «зачета» при построении обратной к матрице второго порядка необходимо допустить не более одной ошибки. Для матрицы третьего порядка – не более двух, а для четвертого – не более трех ошибок. Таким образом, многократно повторяя одни и те же действия, человек обучался решению данной задачи. В ходе исследования были получены следующие результаты:

Рисунок 16 – Первый реципиент

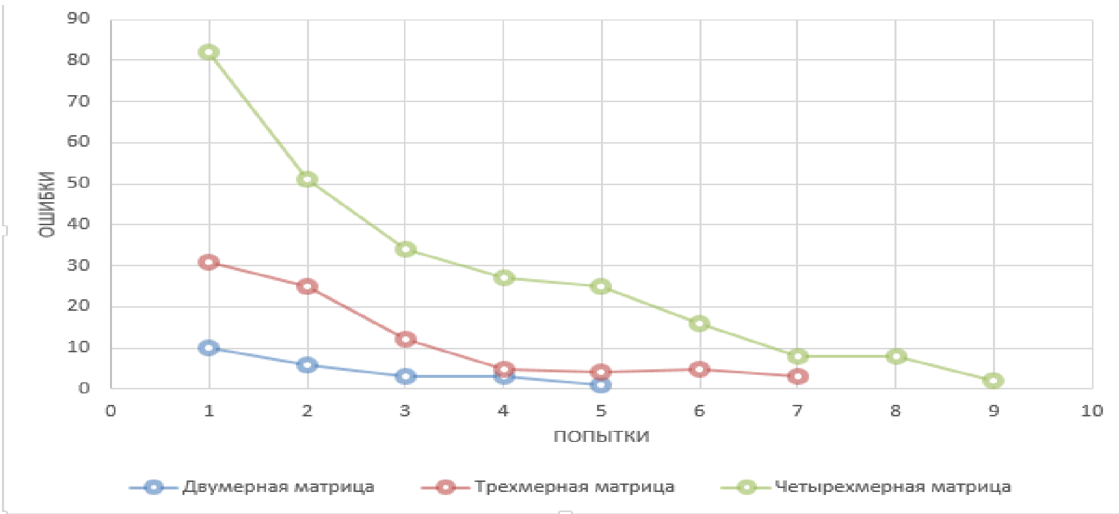

Рисунок 17 – Второй реципиент

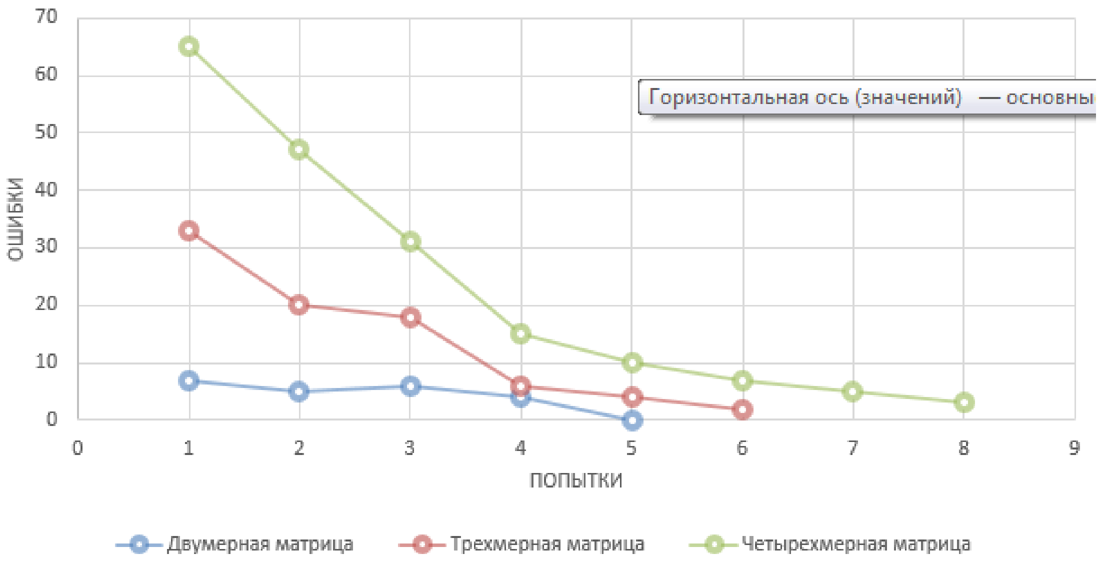

Рисунок 18 – Третий реципиент

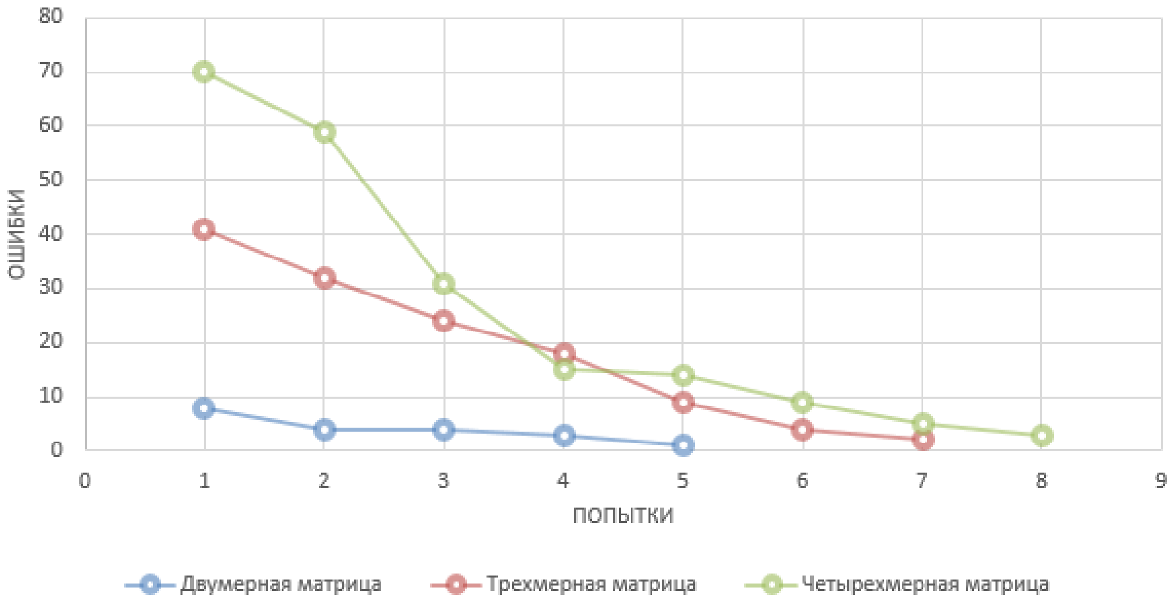

Рисунок 19 – Четвертый реципиент

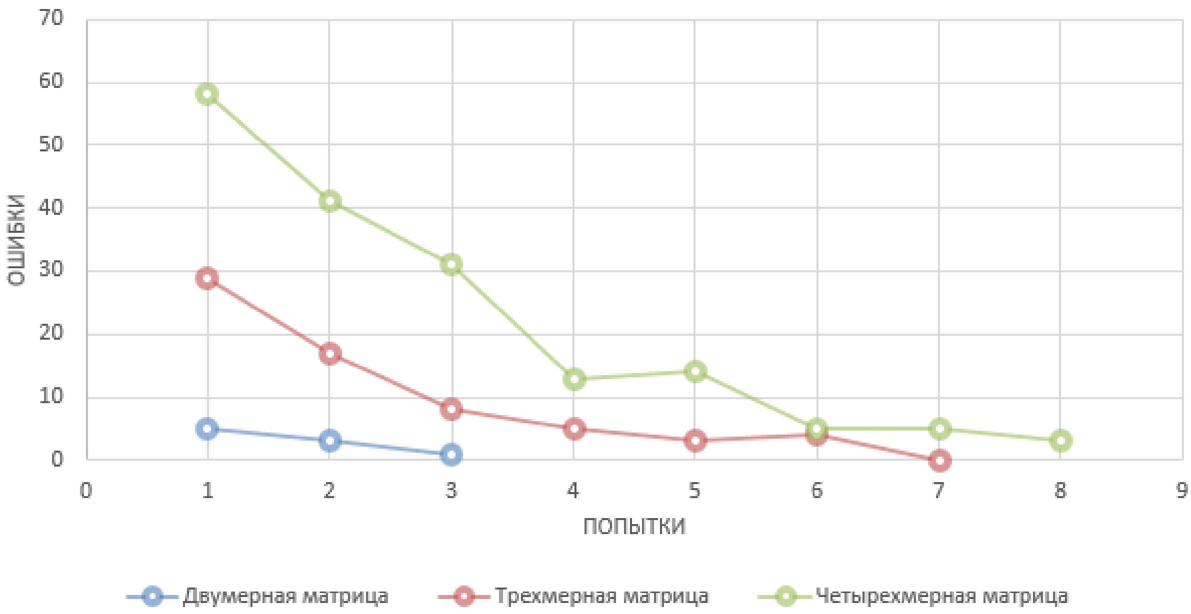

Рисунок 20 – Пятый реципиент

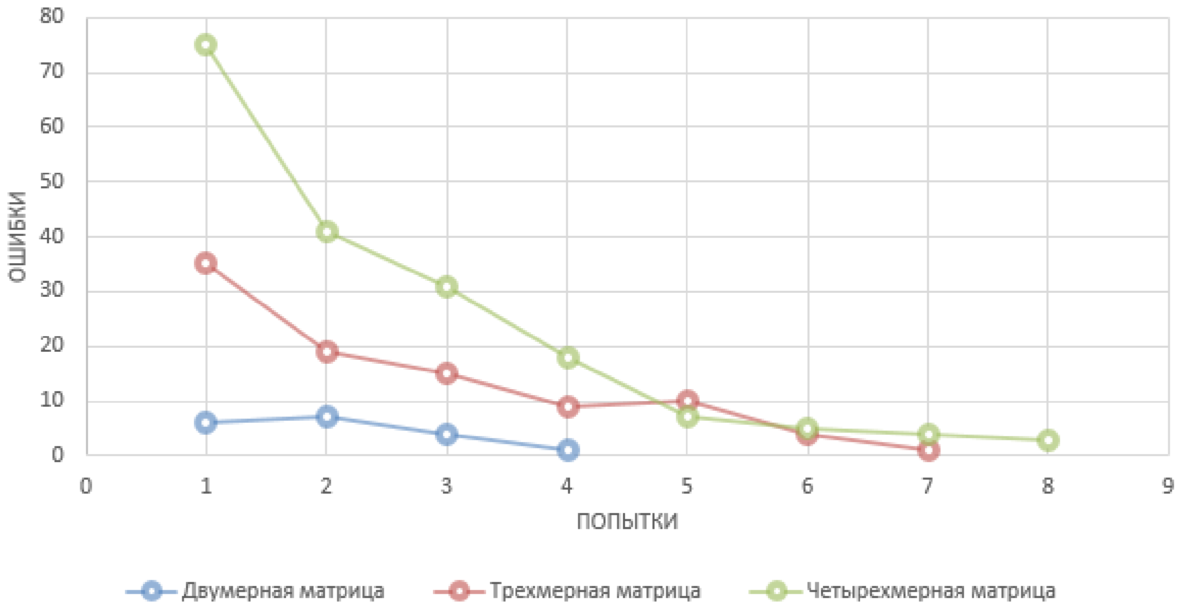

Рисунок 21 – Шестой реципиент

- среднее число попыток для освоения принципов построения обратной матрицы второго порядка – 5,4;

- среднее число попыток для освоения принципов построения обратной матрицы третьего порядка – 6,6;

- среднее число попыток для освоения принципов построения обратной матрицы четвертого порядка – 8,1.

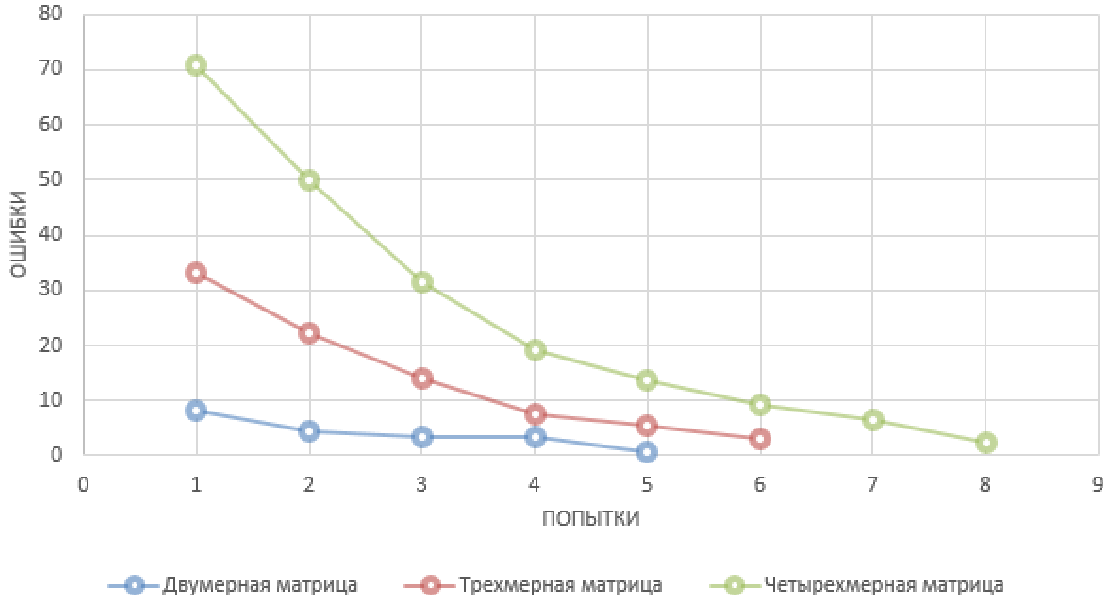

Рисунок 22 – Зависимость количества ошибок от количества попыток и уровня сложности

На рисунке 22 представлена средняя зависимость количества ошибок от количества попыток, в качестве исходных данных было взято среднее количество ошибок в зависимости от уровня сложности.

Очевидно, что при увеличении количества попыток количество ошибок уменьшается, что и является положительным результатом работы тренажера. Другими словами, студент А, путем неоднократного повторения определенных действий с увеличением уровня сложности научился решать данную задачу.

ЗАКЛЮЧЕНИЕ

Предложенная математическая модель и реализованный комплект инструментальных средств представляют собой практическую базу для создания автоматизированных систем обучения по математическим дисциплинам.

В ходе исследования получены следующие результаты:

1. обоснована необходимость создания компьютерных тренажеров по решению задач;
2. разработана модель тренажера на основе когнитивного подхода;
3. создана программная среда для реализации модели;
4. проведен анализ полезности тренажера.

Из всего вышесказанного можно сделать вывод: контроль знаний в системе компьютерного обучения должен осуществляться в формах, наиболее приближенных к формам контроля знаний традиционного обучения. Наиболее подходящими средствами для осуществления такого контроля являются компьютерные тренажеры, которые выявляют компетентность студента на уровне применения теоретических знаний на практике.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Яриков В.В. Иллюстрирующая программа по нахождению первообразной функции для интеграла вида произведения двух непрерывных гладких функций / В.В. Яриков // Образовательные технологии и общество. – 2011. – № 1. С. 337-346.

2 Яриков В.В. Тренажер по нахождению первообразной сложной функции для интеграла вида P(x)Q(x) / В.В. Яриков // Образовательные технологии и общество. – 2011. – № 4. С. 368-376.

3 Ноутон Н. JavaTM 2 / Н. Ноутон, Г. Шилдт; Пер. с англ. – СПБ: БХВ-Петербург, 2003. – 1072 с;

4 Пак Н.И. О модели мышления и ментальных схемах / Н. И. Пак // Практико-ориентированное обучение в профессиональном образовании: проблемы и пути развития: материалы научно-практической конференции в рамках XVIII Международной научной конференции «Решетневские чтения». – Красноярск, 2014. – С. 306–310.

ПРИЛОЖЕНИЕ А

Функция выбора шаблона алгебраического дополнения

```
public void ShablonA(int p, int n, int x, Graphics g)
  {
    int sm = 0;
    int dx = 0;
    String T = null;
    switch(n)
    {
    case 2: // '\002'
      sm = 40;
      break;

    case 3: // '\003'
      sm = 66;
      break;

    case 4: // '\004'
      sm = 90;
      break;
    }
    if(p == -1)
    {
      dx = 22;
      g.setFont(f2);
      g.drawString("-1", x + 25, 175);
    }
    g.setColor(Color.black);
    g.setFont(f1);
    g.drawString("A", x, 195);
    g.drawString("=", x + 30 + dx, 195);
    Bra1(n, x + dx, g);
    for(int i = 0; i < n; i++)
    {
      for(int j = 0; j < n; j++)
      {
        g.setColor(Color.gray);
        g.fillRect(x + 85 + 58 * j + dx, (180 - sm) + 50 * i, 20, 30);
        g.setColor(Color.white);
        g.fillRect(x + 85 + 58 * j + 3 + dx, (180 - sm) + 50 * i + 3, 14, 24);
      }

    }

    Ket1(n, x + dx, g);
  }
```

ПРИЛОЖЕНИЕ Б

Функция вычисления определителя

```
public int Det(int n, int A[][])
  {
    int B[][] = new int[4][4];
    int S;
    if(n == 1)
      S = A[0][0];
    else
    if(n == 2)
    {
      S = A[0][0] * A[1][1] - A[0][1] * A[1][0];
    } else
    {
      int Sign = 1;
      S = 0;
      for(int k = 0; k < n; k++)
      {
        for(int i = 1; i < n; i++)
        {
          for(int j = 0; j < n; j++)
            B[i - 1][j] = A[i][j];

        }

        for(int i = 0; i < n - 1; i++)
        {
          for(int j = k + 1; j < n; j++)
            B[i][j - 1] = B[i][j];

        }

        S += Sign * A[0][k] * Det(n - 1, B);
        Sign = -Sign;
      }

    }
    return S;
  }
```