

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

**РАЗРАБОТКА ЭКСПЕРТНЫХ СИСТЕМ ДЛЯ ПЛАТФОРМЫ
АНДРОИД**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Юдина Павла Витальевича

Научный руководитель
доцент, к. ф.-м. н.

И. А. Батраева

Заведующий кафедрой
к. ф.-м. н.

С. В. Миронов

Саратов 2016

ВВЕДЕНИЕ

Актуальность темы обусловлена тем, что любой человек сможет разрабатывать экспертную систему используя для этого всего лишь мобильное устройство на платформе андроид. В этой программе пользователь может реализовывать любую предметную область и в любой момент воспользоваться ей. В этой программе будет реализован алгоритм, с помощью которого можно получить конкретный ответ в определенной предметной области. Созданные экспертные системы смогут полностью взять на себя функции, выполнение которых обычно требует привлечения опыта человека-специалиста или стать помощником для человека принимающего решения. Ну и главное её достоинство будет то, что человек, работающий в сотрудничестве с такой программой, сможет добиться результатов более высокого качества.

Технология экспертных систем, получила наименование искусственного интеллекта. Исследования в этой области сконцентрированы на разработке и внедрении компьютерных программ, способных эмулировать (имитировать, воспроизводить) те области деятельности человека, которые требуют мышления, определенного мастерства и накопленного опыта. В программе будет реализовано принятие решения на основе фактов. Так же она будет реализована для платформы андроид, чтобы можно было использовать программу практически в любой момент времени.

В этой работе ставятся следующие задачи:

- рассмотреть принципы функционирования продукционных баз знаний;
- рассмотреть модель представления продукционных баз знаний на ЭВМ;
- рассмотреть методы и средства Android Studio;
- реализовать алгоритм, использующий прямую цепочку рассуждений;
- создать интерфейс для приложения, которое будет выполняться на платформе андроид;
- объединить алгоритм и интерфейс в единое приложение.

Настоящая работа представлена введением, пятью разделами, заключением, списком использованных источников и приложением (листингом программы).

Раздел 1. В данном разделе представлено описание принципов функционирования продукционных баз знаний.

Раздел 2. В данном разделе представлено рассмотрение модели пред-

ставления продукционных баз знаний на ЭВМ.

Раздел 3. В данном разделе представлено описание Android Studio и его средства разработки.

Раздел 4. В данном разделе представлено описание методов Android Studio, которые использовались при разработке приложения.

Раздел 5. В данном разделе представлено описание программы, реализующей алгоритм использующий прямую цепочку рассуждений для андроида. [1–10]

1 Основное содержание работы

Введение работы содержит в себе общие положения работы.

Глава 1 «Принципы функционирования продукционных баз знаний» содержит определение продукционной модели и принципы функционирования продукционных баз знаний.

Продукционная модель — это модель представления знаний, в основе которой лежит продукция, т.е. правила типа «если A то B », где A — условие применимости продукций, а B — заключение, которое делается, если установлена истинность условия применения продукции.

Каждое правило продукции можно представить графом с древовидной структурой (рисунок 1). Если существует множество правил, из которых вы-

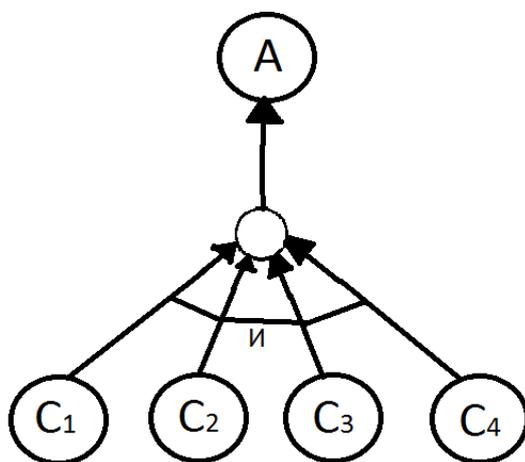


Рисунок 1 – Граф с древовидной структурой

водится одно и то же заключение, то такие правила можно изобразить графом с И и ИЛИ вершинами (рисунок 2).

В базе знаний хранятся не только набор продукционных правил для проведения логического вывода знания, но и дополнительная информация. Примером такой информацией являются вопросы и списки разрешенных значений. Если при диалоге с пользователем неудобно использовать стандартные вопросы типа «Каково значение объекта name?», можно описать вопрос, задаваемый пользователю при инициализации этого объекта. Например:

вопрос(возраст)=Сколько Вам лет?

Иногда лучше написать список разрешенных значений явно(неявно разрешенные значения уже присутствуют в правилах). Например:

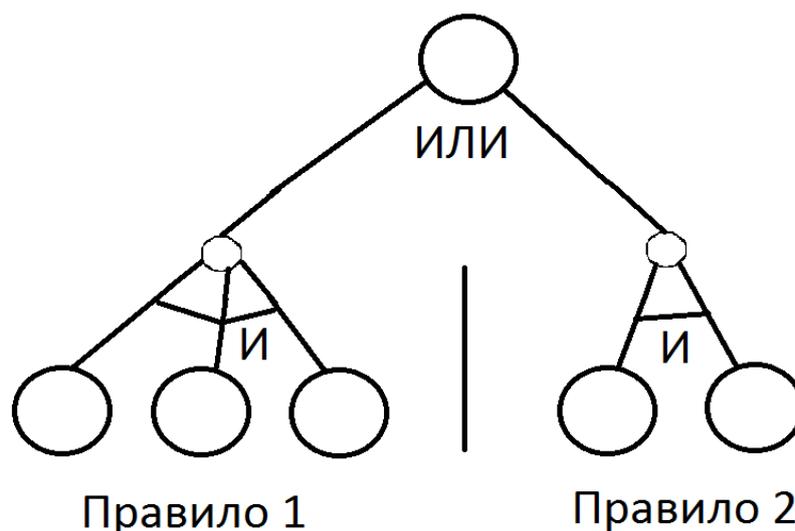


Рисунок 2 – Граф с И и ИЛИ вершинами

разрешен(возраст)=менее 18,18-30,31-60,более 60.

Во время проведения логического вывода продукционная система из исходных фактов с помощью дополнительно полученных от пользователя сведений выводит из баз знаний новые факты. Затем, используя старые и вновь полученные факты, получает новые знания, и так до тех пор, пока это возможно. Реальные базы знаний могут насчитывать сотни правил. Эти правила хранятся в базе в хаотичном порядке. Чтобы найти нужное правило, нужно просмотреть всю базу знаний, что снижает скорость работы продукционных систем.

Механизм логического вывода продукционной модели циклически выполняет четыре последовательных этапа (рисунок 3):

- выборка;
- сопоставление;
- разрешение конфликтов;
- действие или их совокупность.

На каждом из перечисленных этапов интерпретатор работает с базой знаний, рабочей памятью, памятью состояний интерпретатора.

Глава 2 «Модель представления продукционных баз знаний на ЭВМ» содержит в себе модель представления продукционных баз знаний и алгоритмы для проведения прямого логического вывода.

При проведении вывода у пользователя заранее запрашиваются все известные ему данные. Система проверяет части «если» всех правил в базе

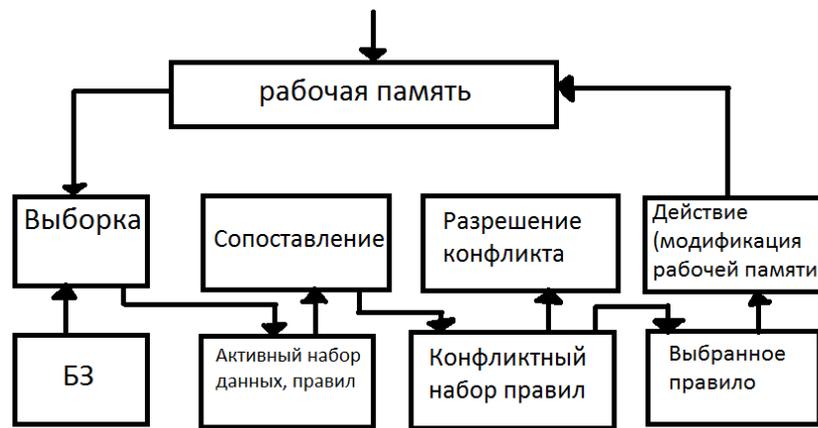


Рисунок 3 – Механизм логического вывода продукционной модели

знаний. При обнаружении наличия всех элементов части «если» некоторого правила в рабочей области, правило выполняется, после чего вносятся соответствующие изменения в рабочую область. Можно заметить, что и в этом случае многочисленные сравнения сильно замедляют работу алгоритма.

Воспользовался алгоритмом 1 для реализации программы.

Алгоритм 1. Проведение прямого логического вывода.

1. Создать очередь установленных вершин-объектов, в которую заносятся объект и его значение, которое удалось установить. Запросить у пользователя значения всех вершин-объектов, значения которых ему известны, и поместить их в очередь. Перейти к шагу 2.
2. Если очередь пуста, перейти к шагу 5, иначе к шагу 3.
3. Взять из очереди объект o_t и его значение $l_{t,it}$. Поместить $l_{t,it}$ в $VL(o_t)$. Окрасить все дуги, ведущие из $l_{t,it}$. Перейти к шагу 4.
4. Для всех вершин-ветвлений, в которые ведут только что окрашенные дуги проверить, если для какой-то вершины-ветвления b все ведущие в нее дуги окрашены, то для всех дуг $((o_p, l_{p,ip}), b)$ пару $(o_p, l_{p,ip})$ занести в очередь установленных вершин. Перейти к шагу 2.
5. Вернуть значение множеств VL для всех вершин-объектов в качестве ответа и завершить работу алгоритма.

Глава 3 «Android Studio и его средства разработки» содержит в себе основные особенности Android Studio и его средства разработки.

Основные особенности — реализована возможность вёрстки в реальном времени, доступно множество вариантов размеров и разрешений экранов. Присутствует раздел справки. Встроены инструменты улучшения качества прило-

жений и монетизации. Имеются инструменты для отслеживания эффективности рекламных объявлений. Добавлено средство взаимодействия с бета-тестерами. И многое другое.

Android Development Tools — плагин для Eclipse Integrated development environment, позволяющий создавать приложения под Android с комфортом.

Менеджеры виртуальных устройств — инструменты для создания и управления виртуальными устройствами в Android (AVD) и управления ими в виде отдельных экземпляров эмулятора. AVD содержит эмулятор, внутри которого работает соответствующая версия Android. С его помощью можно указать версию SDK, разрешение экрана, емкость SD-карты, аппаратные возможности (сенсорный экран, GPS).

Виртуальная машина Android для тестирования и отладки ваших приложений прямо на компьютере, не имея реального устройства. Является реализацией виртуальной машины Dalvik, что делает его совместимым с любым Android-телефоном.

Используйте панель DDMS (Служба для отслеживания процесса отладки в Dalvik) для отслеживания и контроля за виртуальными машинами Dalvik. Это мощный отладочный инструмент, позволяющий изучать активные процессы, просматривать стек и кучу, мониторить и останавливать рабочие потоки, исследовать файловую систему подключенного устройства.

Утилита позволяет получить доступ к файлам базы данных SQLite, которые используются в Android.

Глава 4 «Методы Android Studio, которые использовались при разработке приложения» содержит в себе Методы Android Studio, которые были использованы в программе.

Методы о которых писалось в дипломной работе. Это ListView, AlertDialog, SQLite.

Глава 5 «Реализация алгоритма использующий прямую цепочку рассуждений для андроид» содержит в себе подробное описание разработанной программы. Программа реализована на языке программирования Java в среде Android Studio 1.4.

При запуске приложения в нём не содержится ни одной базы данных, нам предоставляется начальное меню (см. рисунок 4) и создается пустая база данных, которая называется «list» и в ней создается таблица под названием

«bases». В этой таблице будут храниться названия всех баз, которые создадим. Когда нажимаем на кнопку «НАЧАТЬ», то переходим на следующий экран (см. рисунок 5).

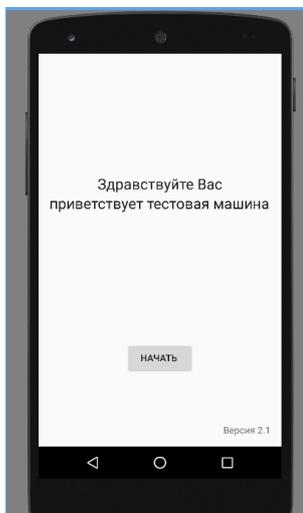


Рисунок 4 – Начальное меню



Рисунок 5 – Экран «выберите действие»

Здесь можно использовать уже существующую базу, либо создать новую, либо удалить ненужную базу. При нажатии на кнопку «СОЗДАТЬ НОВУЮ БАЗУ» переходим на следующий экран (см. рисунок 6). Здесь пишем название нашей новой базы и при нажатии на кнопку «СОЗДАТЬ» проверяем в базе данных «list», что название такой базы не существует.

Если такой базы нет, то заносим название в нашу базу и переходим на следующий экран (см. рисунок 7).

В противном случае выведем пользователю ошибку о том, что база с таким названием уже существует (см. рисунок 8).

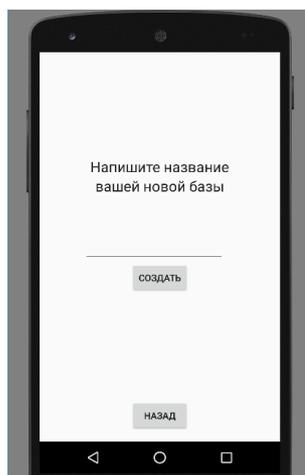


Рисунок 6 – Название базы

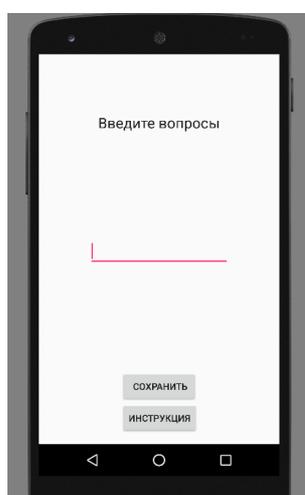


Рисунок 7 – Экран для записи вопросов в базу

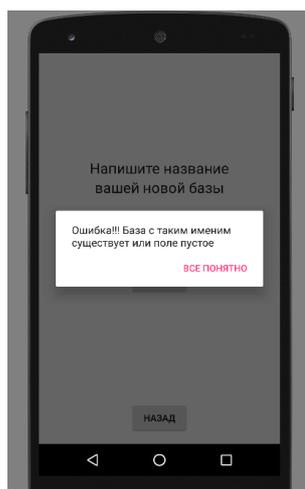


Рисунок 8 – Ошибка в название базы

На следующем экране можем посмотреть инструкцию, если заполняем базу впервые или вдруг забыли, как это делать. Инструкция выглядит следующим образом (см. рисунок 9). После того, как заполнили вопросы, переходим

на следующий экран, где заполняем исключения (см. рисунок 10).

Для исключений представлена собственная инструкция, которой нужно придерживаться для заполнения исключений. После этого переходим к последнему этапу заполнения базы, на этом этапе задаются правила. Для данной стадии также представлена инструкция заполнения. После того как заполнили базу, то можем ее использовать, перейдя на следующий экран. Изображения этих экранов можно посмотреть в дипломе.

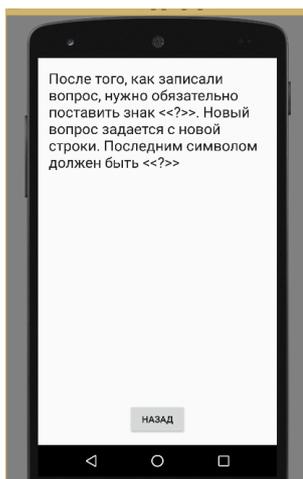


Рисунок 9 – Инструкция при записи вопросов в базу

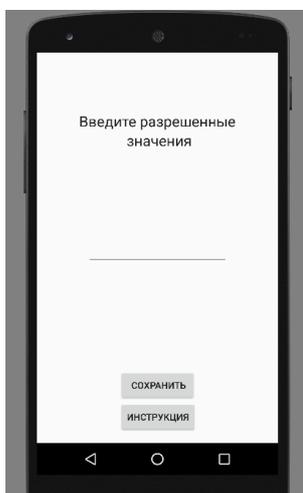


Рисунок 10 – Задаем исключения

Нам предлагают список всех баз данных, которые сейчас существуют, выберем одну из них (см. рисунок 11).

Если не выбрать базу, то выведется ошибка (см. рисунок 12). Дальше будут считываться вопросы из базы данных и будут выведены варианты ответов на вопросы (см. рисунок 13). После того, когда ответили на все вопросы, получим результат (см. рисунок 14).

Результат ищется при помощи Алгоритма 1, который был описан ранее. Нам будет предъявлено два вида результата. Полный ответ, все что нашла программа, когда проходила по всем правилам. И последний ответ, он предназначен для баз, в которых важен лишь последний результат поисков без лишней информации. Также мы можем удалить базу (см. рисунок 15).

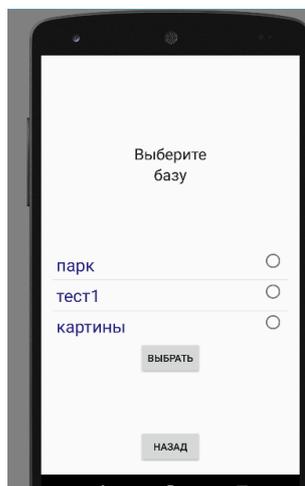


Рисунок 11 – Выбор базы

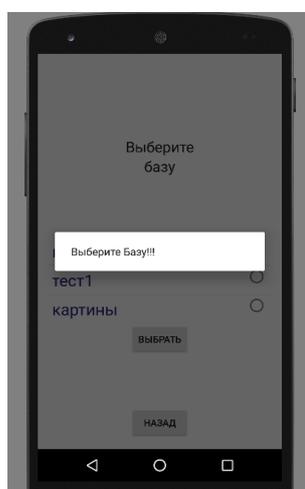


Рисунок 12 – Ошибка при выборе базы

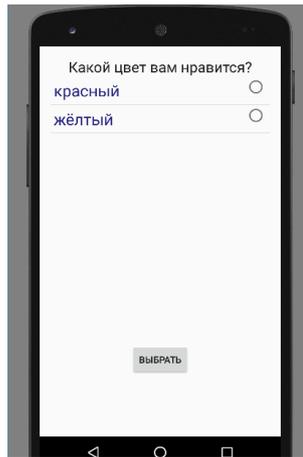


Рисунок 13 – Отвечаем на вопросы

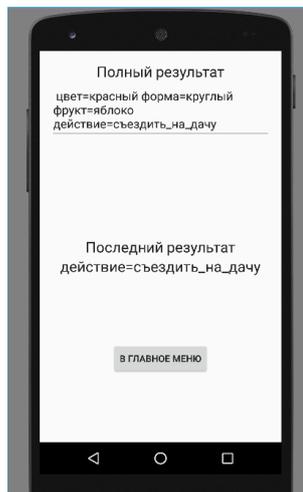


Рисунок 14 – Результат

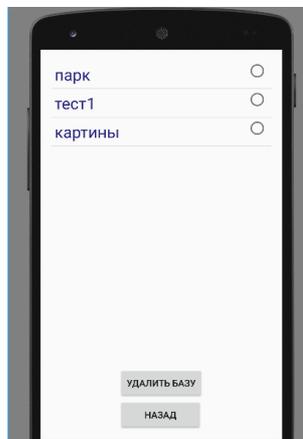


Рисунок 15 – Удаляем базу

ЗАКЛЮЧЕНИЕ

В настоящей работе рассматривались вопросы затрагивающие модель представления продукционной базы знаний на ЭВМ, адаптированная для реализации прямой цепочки рассуждений и реализации ее для андроид: рассмотрены принципы функционирования продукционных баз знаний, рассмотрена модель представления продукционных баз знаний на ЭВМ. Так же исследованы методы и средства Android Studio. Так же была написана программа для платформы андроид в которой можно создавать любые предметные области, за счет этого программа стала более функциональной. И реализован алгоритм, использующий прямую цепочку рассуждений. Так же были реализованы: удаление предметной области, инструкции для создания предметной области и хранения их на телефоне при помощи баз данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Яхно, Т. М.* Средства представления и методы обработки знаний в интеллектуальных системах: Автореф. дис. д-ра физ. мат. наук / Т. М. Яхно. — Новосибирск, 1997.
- 2 *Song, Z.* Research efforts to improve performance of production systems / Z. Song. — J. Harbin Inst. Techn., 2001.
- 3 *Головчинер, М. Н.* Введение в системы знаний / М. Н. Головчинер. — Томск, 2011.
- 4 *Бек, Л.* Введение в системное программирование / Л. Бек. — «Мир», 1988.
- 5 *Климов, А.* Средства разработки для Android [электронный ресурс]. / А. Климов. — 2015. — URL: <http://developer.alexanderklimov.ru/android/theory/tools.php/> (дата обращения 23.05.2016) Загл. с экр. Яз. рус.
- 6 *Alviere, D.* Диалоговые окна в Android. Часть 1 [электронный ресурс]. / D. Alviere. — 2013. — URL: [https://habrahabr.ru/post/166469//](https://habrahabr.ru/post/166469/) (дата обращения 23.05.2016) Загл. с экр. Яз. рус.
- 7 *Иванов, А. С.* Математические модели и алгоритмы функционирования продукционных баз знаний / А. С. Иванов. — Саратов, 2008.
- 8 *Вагин, В. Н.* Параллелизм в продукционных моделях представления знаний / В. Н. Вагин. — 1994.
- 9 *Попов, Э. В.* Экспертные системы / Э. В. Попов. — «Наука», 1987.
- 10 *Форсайта, Р.* Экспертные системы. Принципы работы и примеры / Р. Форсайта. — «Радио и связь», 1987.