

УДК 004.416.6

**Снегирева С.А., Тихомирова А.Н., Тахтеева П.М.**

Национальный исследовательский ядерный университет МИФИ, г. Москва, Россия

**ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ОБУЧАЮЩЕГО И КОНТРОЛЬНОГО РЕЖИМОВ ЭМУЛЯТОРА ДЛЯ СПРАВОЧНО-ОБУЧАЮЩЕЙ СИСТЕМЫ ПО КУРСУ «ТЕОРИЯ АЛГОРИТМОВ»****Аннотация**

*В данной статье рассматриваются проектирование и реализация обучающего и контрольного режимов эмулятора нормальных алгоритмов Маркова для справочно-обучающей системы по курсу «Теория алгоритмов». Представлен анализ и сравнительная характеристика существующих на данный момент аналогов. Выбран способ автоматизированной оценки качества написанных алгоритмов и приведен алгоритм оценивания.*

**Ключевые слова**

*Обучающие тренажеры; теория алгоритмов; обучающие системы; автоматизированный контроль знаний.*

**Snegireva S.A., Tikhomirova A.N., Takhteeva P.M.**

National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, Russia

**DESIGN AND IMPLEMENTATION OF THE TRAINING AND CONTROL MODES OF THE EMULATOR FOR THE REFERENCE-TRAINING SYSTEM ON THE COURSE «THEORY OF ALGORITHMS»****Abstract**

*This article deals with the design and implementation of the training and control modes of the emulator of Markov algorithms for the reference and training system for the course "Theory of algorithms". The analysis and comparative characteristics of existing analogues are presented. The method of automated estimation of the quality of the written algorithms is selected and the estimation algorithm is given.*

**Keywords**

*Training simulators; theory of algorithms; training systems; automated knowledge control.*

**Введение**

Разработка и применение обучающих систем в высшей школе – это одна из составляющих компонентов информатизации вуза. Главная цель информатизации в вузе – это повышение качества подготовки специалистов посредством внедрения в учебный и научный процессы новых информационных технологий, средств мультимедиа и телекоммуникаций. Технология обучения является информационной, если она существенно опирается на компьютерные средства и методы приема, обработки, передачи и отображения учебной информации [1].

Применение компьютеров в учебном процессе как средство контроля качества знаний получает все большее распространение. Как показывает практика, при должной научно-методической

подготовке автоматизация контроля позволяет заметно повысить, прежде всего, индивидуальность самого контроля, варьировать его в зависимости от способностей и образовательных ценностей студентов [2]. Кроме того, автоматизированный контроль повышает объективность самого контроля, позволяет оценивать качество знаний не только «в общем и целом», но обеспечивает количественную оценку качества усвоения студентами того или иного раздела учебного курса. Еще одним важным аспектом применения автоматизированного контроля, как показала практика, становится стимулирование и мотивирование студентов к самообразовательной деятельности. При условии соблюдения должных дидактических правил, методы автоматизированного контроля

предоставляют информацию о качестве знаний не только преподавателю, но и самим студентам [3].

При разработке подобных систем следует уделить должное внимание анализу существующих аналогов, выявить их достоинства, учесть недочеты таких систем, сформулировать требования, которые должны отражать потребности как преподавателей, так и самих студентов, а также удовлетворять педагогическим рекомендациям. Кроме того, необходимо грамотно подойти к выбору средств проектирования и реализации.

#### **Цель исследования**

В работе планируется обозначить необходимость автоматизации контроля знаний у студентов по курсу «Теория алгоритмов», спроектировать и реализовать эмулятор-тренажер нормальных алгоритмов Маркова, а также контрольный режим этого эмулятора, который позволит провести объективную и эффективную проверку знаний, позволит студентам отследить свои ошибки непосредственно после контрольных работ, что может служить дополнительным стимулом в процессе обучения

Кроме того, требуется представить анализ существующих систем, выбрать и обосновать алгоритм оценивания написанных студентами работ, рассмотреть процессы проектирования и реализации системы.

#### **1. Анализ и сравнительная характеристика существующих обучающих тренажеров НАМ**

Нормальные алгоритмы Маркова (далее – НАМ), введенные советским математиком А. А. Марковым, представляют собой класс алгоритмов, применимых к словам некоторого алфавита. Каждый НАМ определяется указанием алфавита, в котором он действует, и схемы НАМ [4].

На сегодняшний день в сети представлен достаточно широкий диапазон эмуляторов НАМ. Все они воплощают идею о тренажере, позволяющем отладить алгоритм. Рассмотрим некоторые из них подробнее.

Онлайн-эмулятор, доступный по ссылке <http://cmcmsu.no-ip.info/1course/alg.schema.nam.htm>, представляет собой приложение, включающее в себя минимальный набор функций, но выполняющее главную из них – применение алгоритма к любому входному слову и просмотр результата. Основным функционалом данного тренажера заключаются в следующих возможностях:

- отладка введенной последовательности;
- подсветка выполняемого действия;
- возможность пошагового выполнения алгоритма.
- Как было замечено выше, возможности этого эмулятора ограничены. К недостаткам можно отнести отсутствие следующих возможностей:

- сохранение программы и считывания из файла;
- добавление комментария к строке.

Другим примером реализации эмулятора НАМ является приложение, скачиваемое по ссылке: <http://yad-studio.github.io/>. Эмулятор обладает следующим функционалом:

- отладка введенной последовательности;
- подсветка выполняемого действия;
- возможность пошагового выполнения алгоритма;
- сохранение введенной программы;
- ведение истории вводимых строк.

Несмотря на широкие возможности и продуманный удобный интерфейс, данный эмулятор не является веб-приложением и не дает возможности осуществлять контроль за обучением в режиме реального времени.

Таким образом, задача создания обучающего онлайн-эмулятора, адаптированного под нужды обучающегося и преподавателя и содержащего необходимый и полный для образовательного модуля функционал является уместной. На основании проведенного анализа был составлен список требований к эмулятору, в котором учтены достоинства и недостатки рассматриваемых аналогов. Кроме того, принято решение о введении возможности работы эмулятора в двух режимах: обучающем и контрольном. Работа в режиме самопроверки предоставит возможность проводить обучение студентов автономно и в более наглядной, интерактивной форме, а также отработать навыки по написанию алгоритмов. Контрольный режим позволит провести объективную оценку полученных студентами навыков по написанию алгоритмов.

#### **2. Выбор метода оценивания знаний для контрольного режима работы эмулятора**

Контроль знаний, умений, навыков студентов – один из важнейших элементов учебного процесса. Выбор способа оценки качества знаний всегда является ключевым при реализации образовательной программы. Исходя из требований к проверке и характера задач, предлагаемых обучающемуся в качестве контрольного материала, была выбрана оптимальная стратегия оценивания: оно должно производиться по корректной работе алгоритма и верному преобразованию различных входных данных; кроме того, полезно учитывать такие показатели, как скорость выполнения задания, а также количество итераций, потребовавшихся для преобразования входного слова. Для этого сформированы тестовые последовательности для контрольных заданий, представляющие собой входные слова и результаты верного преобразования. Результат должен быть сопоставлен с ответом, полученным после применения к соответствующему входному слову

алгоритма, составленным обучающимся.

Для применения критерия, позволяющего учесть быстроту выполнения контрольной задачи, предполагается введение коэффициента  $T$ , значения которого варьируются в диапазоне от 0,5 до 1,5. Таким образом, вводится система «штрафов» и «поощрений», позволяющая стимулировать студентов к более быстрому выполнению работы.

$$T(t) = \begin{cases} -0,0125t + 1,5, t \in ; \\ 1, t \in [40; 50]; \\ -0,0125t + 1,625, t \in . \end{cases}$$

Итоговый балл также должен иметь зависимость от количества итераций, которые потребовались для преобразования, так как длина алгоритма не обязательно коррелирует с его сложностью (при небольшой разнице).

Для достижения этой цели решено ввести коэффициент  $E$ , характеризующий эффективность алгоритма, который должен учитывать количество изменений на различных входных словах одинаковой длины (10 символов). Этот коэффициент должен быть рассчитан по следующей формуле:

$$E = \max \left( \frac{E_{\min_i}}{E'_i} \right) i \in \{1, 2, 3, 4, 5\},$$

где  $E'_i$  – число итераций, затраченное на преобразование  $i$ -ого слова алгоритмом, написанным студентом;  $E_{\min_i}$  – минимальное число итераций, потребовавшихся для преобразований над  $i$ -ым словом, это число сохраняется для каждой последовательности, причем его значение в базе обновляется (становится равным  $E'_i$ ) в случае, когда коэффициент  $E$  становится больше единицы. Так как параметр является косвенной характеристикой алгоритма, то его влияние на итоговую оценку должно выражаться в относительно небольших вычетах и надбавках: если студенту удалось создать алгоритм, позволяющий провести операции над словом быстрее, чем были созданы до этого, то оценка будет увеличена на 10%, в противном случае снижена на 5%.

Исходя из представленных данных, формируется оценка. Таким образом, оценка за задачу имеет следующий вид:

$$O_j = M_j * K_j * T_j * E_j, j \in \{1, 2\},$$

где  $M_j$  – максимальный балл за  $j$ -ую задачу,  $T_j$  – временной коэффициент выполнения  $j$ -ой задачи,  $E_j$  – коэффициент эффективности алгоритма,  $K_j$  – коэффициент, определяющийся количеством верно отлаженных последовательностей.

Такой метод оценивания является наиболее эффективным и оптимальным для задач данного типа и удовлетворяет предъявляемым к нему требованиям.

### 3. Проектирование эмулятора НАМ

#### 3.1. Проектирование интерфейса

Проектирование пользовательского интерфейса – важная часть разработки обучающего тренажера. Основными в рамках поставленной задачи являются следующие требования к пользовательскому интерфейсу:

- функциональность (соответствие задачам пользователя);
- понятность и логичность;
- обеспечение высокой скорости работы пользователя;
- обеспечение защиты от человеческих ошибок;
- быстрое обучение пользователя.

Для комфортной работы в обучающем режиме предусмотрено поле ввода условия задачи (Рис. 2). Введенную информацию можно удалить, воспользовавшись кнопкой, расположенной в правой части поля («крестик»). Входное слово также вводится пользователем в отведенное для этого поле. В блоке *Алгоритм* вводится текст программы – последовательность строк, которые будут применены к входному слову.

Следует обратить особое внимание на то, в правильном ли порядке находятся строки алгоритма. Это один из важнейших аспектов работы алгоритма и источник большинства ошибок. Как правило, схема такая:

- строки, содержащие условия принудительного окончания алгоритма;
- основное тело программы (алгоритма);
- начальные условия работы алгоритма (добавление флага, поиск начала слова и т.д.).

В правой части окна отображается пошаговое применение алгоритма к слову и конечный результат его применения. Для редактирования формул используются элементы, расположенные внизу поля с программой. Слева каждой строки находится кнопка ее удаления. Выбранные строки можно очистить, удалить или добавить новую до или после выделенной. Вся справочная информация для работы доступна при нажатии кнопки в верхнем правом углу окна.

В режиме контрольной работы интерфейс должен выглядеть несколько иначе, так как он должен быть приспособлен для проверки знаний студентов.

Для написания контрольной работы студенту необходимо решить две задачи, при этом сложность заданий выбирается им самостоятельно. В контрольном режиме эмулятора нормальных алгоритмов Маркова предусмотрена возможность ознакомления с заданиями сложного и легкого уровней, после чего обучающийся может выбрать подходящий ему вариант и приступить к решению. После завершения ввода правил в отведенное поле и отправки решения происходит проверка корректной работы алгоритма, а также

оценивание его по критериям. Свой результат можно будет увидеть в выделенном для этого поле, в котором появится балл за выполненное задание, исходя из выбранного уровня сложности и количества допущенных ошибок.

Для учета требования по предоставлению студентам информации о допущенных ошибках планируется рядом с выставленным баллом поместить таблицу, в которой будут сопоставлены входные слова, по которым велась проверка введенного алгоритма, с теми, что получились в ходе его выполнения. Таким образом, студенты смогут ознакомиться со своими ошибками, что в педагогическом плане играет важную роль в процессе обучения.

Для удобной работы на странице одновременно отображаются две задачи, решать которые допускается в произвольном порядке. Общая оценка за контрольную работу складывается из суммы полученных баллов.

### 3.2. Проектирование модуля работы с контрольным материалом

Для реализации функциональной части контрольного режима необходимо создание базы данных, обращение к которой должно осуществляться для получения контрольных заданий, информации и о сложности, а также тестовых последовательностей для каждого типа задач, необходимых для оценки корректной работы введенного пользователем алгоритма. Кроме того, в системе должна быть реализована возможность интерфейсного взаимодействия с базой данных для изменения или удаления задач, а также добавления нового контрольного материала. Для реализации базы данных была выбрана СУБД (система управления базами данных) MySQL.

Для проектирования БД использован Erwin – средство для проектирования и документирования баз данных, которое позволяет создавать, документировать и сопровождать базы данных. Схема реляционной базы данных в виде диаграммы физического уровня системы ERwin приведена на рисунке 1.

Таким образом, спроектирован модуль управления данными, позволяющий преподавателю контролировать заполнение базы данных контрольными задачами и тестовыми последовательностями, а также удаление и редактирование материала.

### 4. Реализация эмулятора НАМ

Работа с эмуляторами в обычном режиме ведется через веб-интерфейс, поэтому для взаимодействия в контрольном режиме преподавателем и обучающимися был использован стандартный язык разметки HTML для реализации интерфейсной части. Для обработки введенных студентами алгоритмов реализован PHP-скрипт, который запускается на

сервере после выполнения текущего задания и нажатия кнопки «Отправить». При этом информация со страницы преобразуется в JSON-строку. Скрипт обрабатывает ее и передает алгоритм на проверку с входными словами, подгружаемыми из базы данных.

Все ответы сохраняются на сервере, на котором и происходит сравнение полученных отработанных слов с «эталонными» последовательностями, которые также берутся из базы данных. После этого происходит формирование оценки, основанное на вышеописанных критериях. Для учета времени выполнения задания в базу данных поступает информация о начале сессии и ее завершении. Балл передается с сервера на веб-страницу и записывается в базу данных, затем передается на страницу, видимую пользователем.

Работа обучающего режима отличается тем, что на вход алгоритму передается слово, которое задает пользователь. После обработки выходное слово, полученное в результате применения к нему алгоритма, передается на страницу в специально отведенное для этого поле.

Для реализации функциональной части контрольного режима необходимо было подготовить базу данных, обращение к которой должно осуществляться для использования, добавления, удаления контрольных заданий, получения информации и о сложности, а также тестовых последовательностей для каждого типа задач, необходимых для оценки корректной работы введенного пользователем алгоритма.

Основной язык, которым описывается графический интерфейс веб-приложения — это HTML. Этот язык описывает структуру веб-страницы. Художественное оформление веб-страниц описывается таблицами стилей — CSS. Для реализации дополнительных необходимых функций графического интерфейса используются дополнительные технологии: скрипты JavaScript и JQuery. Также был использован скриптовый язык PHP.

Реализована возможность просмотра всех имеющихся в базе задач и тестовых последовательностей, также организованная с помощью запроса к базе данных в скрипте *alltasks.php*. Кроме того, преподаватель может удалить по своему усмотрению задачу со всеми тестовыми последовательностями, либо отдельные тестовые последовательности. Для исполнения запроса на удаление нужной строки выводимой таблицы используется метод GET, позволяющий передать по ссылке значение ID выбранной задачи или последовательности, которое используется для удаления в одной или двух связанных таблицах. На рисунке 2 представлен скриншот интерфейса эмулятора в обучающем режиме.

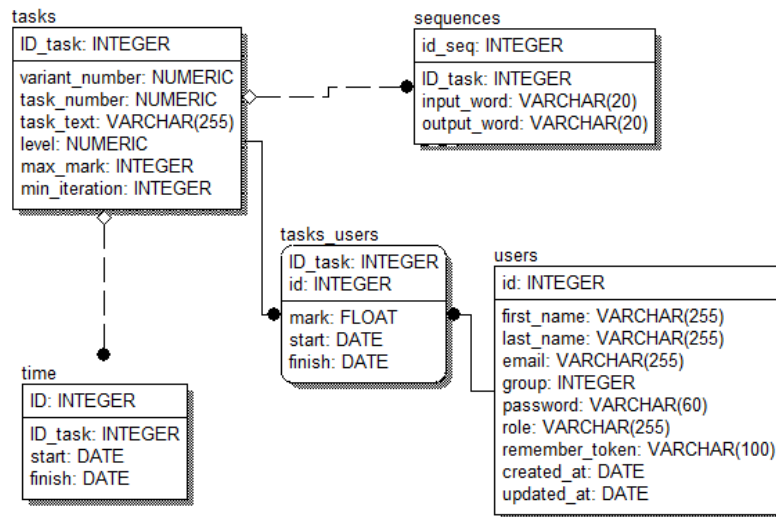


Рисунок 3.2.1 – Диаграмма в системе ERwin физического уровня

Контрольная работа по теме "Нормальные алгоритмы Маркова"

ЗАДАЧА №1

Задача №1

Ваш алгоритм:

0 → 1

Специфические символы: # 0 1

Отладка: Входное слово: [ ]

ОТПРАВИТЬ

Ваш балл: 0

ЗАПУСК

Результат:

#	Входное слово	Верное преобразование	Ваш результат
1	01100101001010	2	11111111111111
2	010010010010	3	111111111111
3	01000001	1	11111111
4	1111111111	0	11111111
5	00000000	0	11111111

Рисунок 4.1 – Скриншот интерфейса контрольного режима эмулятора НАМ

### Заключение

В данной работе рассматривался процесс проектирования и реализации эмулятора нормальных алгоритмов Маркова в обучающем и контрольном режимах. Приложение является составной частью справочно-обучающей системы для студентов по курсу «Теория алгоритмов». Тренажер позволит проводить обучение студентов автономно и в более наглядной, интерактивной

форме, а также отработать навыки по написанию алгоритмов. Для преподавателя контрольный режим позволит избежать неудобств с субъективным выставлением оценок, ввести оперативную и удобную систему проверки. Таким образом, программа предназначена для обучения и закрепления пройденного материала студентами, а также призвана помочь преподавателю в проведении занятий и контрольных мероприятий.

## Литература

1. Нигматуллина Г.В., Варфоломеева Т.Н. Автоматизация контроля результатов обучения учащихся // Материалы VII Международной студенческой электронной научной конференции «Студенческий научный форум» [Электронный ресурс] URL: <http://www.scienceforum.ru/2015/976/7989> (дата обращения: 15.06.2017).
2. Дорошенко С. И., Никольская В. С. Особенности использования автоматизированного контроля знаний студентов //Статья [Электронный ресурс], сайт «Ispu», URL: [http://ispu.ru/files/SovetRector\\_10-2008% 20c. – 2016. – С. 79-82.](http://ispu.ru/files/SovetRector_10-2008%20c.-2016.-S.79-82) (дата обращения: 15.06.2017).
3. Карбелашвили Н.О. Разработка и применение интерактивных информационных обучающих систем на основе современных программных средств. Сетевой электронный научный журнал "СИСТЕМОТЕХНИКА", № 1, 2003 г. Стр. 23
4. Тихомирова А.Н. Теория алгоритмов: Учебное пособие. М.: МИФИ, 2008. Стр. 47.

## References

1. Nigmatullina G.V., Varfolomeeva T.N. AVTOMATIZACIJA KONTROLJA REZUL'TATOV OBUChENIJA UChASHhIHShJa // Materialy VII Mezhdunarodnoj studencheskoj jelektronnoj nauchnoj konferencii «Studencheskij nauchnyj forum» [Jelektronnyj resurs] URL: <http://www.scienceforum.ru/2015/976/7989> (data obrashhenija: 15.06.2017).
2. Doroshenko S. I., Nikol'skaja V. S. Osobennosti ispol'zovanija avtomatizirovannogo kontrolja znanij studentov //Stat'ja [Jelektronnyj resurs], sajt «Ispu», URL: [http://ispu.ru/files/SovetRector\\_10-2008% 20c. – 2016. – S. 79-82.](http://ispu.ru/files/SovetRector_10-2008%20c.-2016.-S.79-82) (data obrashhenija: 15.06.2017).
3. Karbelashvili N.O. Razrabotka i primenenie interaktivnyh informacionnyh obuchajushhih sistem na osnove sovremennyh programnyh sredstv. Setevoj jelektronnyj nauchnyj zhurnal "SISTEMOTEHNIKA", № 1, 2003 g. Str. 23
4. Tihomirova A.N. Teorija algoritmov: Uchebnoe posobie. M.: MIFI, 2008. Str. 47.

Поступила: 17.07.2017

### Об авторах:

**Снегирева София Александровна**, студентка, Национальный исследовательский ядерный университет МИФИ, [sofiko\\_sneg@mail.ru](mailto:sofiko_sneg@mail.ru);

**Тихомирова Анна Николаевна**, кандидат технических наук, доцент кафедры кибернетики, факультет кибернетики, Национальный исследовательский ядерный университет МИФИ, [anna7909966@ya.ru](mailto:anna7909966@ya.ru);

**Тахтеева Полина Михайловна**, студентка, Национальный исследовательский ядерный университет МИФИ, [t-polya@mail.ru](mailto:t-polya@mail.ru).

### Note on the authors:

**Snegireva Sofia A.**, student, National Research Nuclear University MEPHI (Moscow Engineering Physics Institute), [sofiko\\_sneg@mail.ru](mailto:sofiko_sneg@mail.ru);

**Tikhomirova Anna N.**, Candidate of Technical Sciences, Associate Professor, Department of Cybernetics, Faculty of Cybernetics, National Research Nuclear University MEPHI (Moscow Engineering Physics Institute), [anna7909966@ya.ru](mailto:anna7909966@ya.ru);

**Takhteeva Polina M.**, student, National Research Nuclear University MEPHI (Moscow Engineering Physics Institute), [t-polya@mail.ru](mailto:t-polya@mail.ru).