

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА СИСТЕМЫ АДАПТИВНОГО ТЕСТИРОВАНИЯ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

Студентки 4 курса 411 группы
направления 02.03.02 – Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Пиратинской Юлии Дмитриевны

Научный руководитель _____ В. А. Векслер
к. п. н., доцент

Заведующий кафедрой _____ А. С. Иванов
к. ф.-м. н., доцент

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Основное содержание работы	4
ЗАКЛЮЧЕНИЕ	11
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	12

ВВЕДЕНИЕ

Традиционные методы тестирования представляют из себя заранее определенный набор вопросов, которые подаются по очереди испытуемому. Для прохождения такого теста обычно дается ограниченное количество времени. Такой подход не позволяет полноценно оценить знания тестируемого.

Для повышения точности проверки знаний была разработана модель адаптивного тестирования. Такая модель подразумевает, что для каждого испытуемого будет создан свой собственный маршрут тестирования, который будет зависеть от его уровня знаний.

Появление адаптивного тестирования было вызвано стремлением к повышению эффективности педагогических измерений, которая, как правило, связывалась с уменьшением числа заданий, времени, стоимости тестирования, а также с повышением точности оценок учащихся.

Актуальность темы бакалаврской работы определяется необходимостью разработки системы адаптивного тестирования, устраняющей недостатки традиционных систем, обеспечивающей реализацию адаптивных методов тестирования.

Целью настоящей работы является разработка приложения на языке программирования C# для реализации адаптивного тестирования. В приложении должны быть реализованы такие функции как:

- возможность создания адаптивного теста;
- добавление вопросов к тесту;
- прохождение адаптивного тестирования.

Подразумевается, что пользователи смогут создавать тесты, добавлять к ним вопросы, а так же проходить тестирование, для проверки своих знаний.

Исходя из обозначенной цели, были определены следующие задачи:

- изучить научную литературу по теме исследования;
- ознакомится с алгоритмами адаптивного тестирования;
- разработать приложение для проведения адаптивного тестирования.

1 Основное содержание работы

Бакалаврская работа состоит из обозначений, введения, 2 разделов, заключения, списка использованных источников и 3 приложений. Первый раздел содержит четыре параграфа, второй раздел — восемь параграфов. Общий объем работы — 51 страница, из них 41 страница — основное содержание, включая 23 рисунка и списка использованных источников из 20 наименований.

Первый раздел «Адаптивное тестирование» рассматривает теоретическую часть адаптивного тестирования. Компьютерное адаптивное тестирование имеет три важных составляющих компонента: начальную позицию, подбор вопросов и оценку способностей, критерий остановки. Главный принцип системы такого тестирования заключается в подборе вопроса. В начале тестирования каждому испытуемому выводится вопрос среднего уровня сложности. После оценки ответа будет подобран следующий вопрос, соответствующий способности испытуемого. Если ответ на вопрос верный, то следующий вопрос будет сложнее, иначе — легче. Тестирование будет продолжаться до тех пор, пока оценка способности испытуемого не станет стабильной, или не будет достигнут критерий остановки (тогда тестирование завершится).

В параграфе 1.1 «Преимущества адаптивного тестирования» рассказывается о преимуществах адаптивного тестирования в современном мире. За счет своего индивидуального подхода, адаптивное тестирование является одним из наиболее эффективных методов проверки усвоения материала обучающимся. В современном мире уже давно используют компьютерные системы для обучения и в том числе для проведения тестирования. В отличие от традиционных методов тестирования, адаптивный подход может быть реализован, только с помощью компьютерной системы. Такой подход требует определенных алгоритмом вычислений для выдачи следующего вопроса испытуемому [1]. К числу важных преимуществ компьютерного адаптивного тестирования можно отнести индивидуализацию темпа выполнения теста и высокий уровень мотивации к тестированию у наиболее слабых обучаемых за счет исключения слишком трудных заданий [2].

В параграфе 1.2 «Алгоритмы и стратегии адаптивного тестирования» рассматриваются различные методы и подходы адаптивного тестирования. В работе адаптивное тестирование делится на проведение тестирования в два этапа и множество этапов, на основе которых выстраиваются различные стра-

тегии и разрабатываются алгоритмы.

При первом варианте проведения адаптивного тестирования предполагается наличие двух этапов. На первом этапе прохождения тестирования всей группе испытуемых дается один и тот же вариант входного теста. Такой тест дается для расчета предварительной оценки знаний учащихся. На основе его прохождения во втором этапе каждому участнику выдается индивидуальный адаптивный тест [3].

Стратегии адаптивного тестирования, которые проходят в множество этапов представляет собой такую модель проведения теста, где каждый испытуемый движется по маршруту тестирования, который индивидуален для него. Данный вид тестирования проводится на основе алгоритмов с полной зависимостью, где переход к следующему вопросу, возможен только после оценки ответа на предыдущий вопрос [4].

В подпункте 1.2.1 «Общий алгоритм адаптивного тестирования» параграфа 1.2 приведен обобщенный алгоритм адаптивного тестирования. В общем случае, адаптивное тестирование проводится по следующему алгоритму:

- Для первого вопроса выбирается либо заранее заданное значение уровня вопроса, обычно это средний уровень, либо берется в соответствии с уровнем испытуемого;
- Следующие вопросы выдаются в зависимости от того, как тестируемый ответил на предыдущий вопрос;
- Если ответ был верным, то следующий вопрос будет выше по уровню сложности. В противном случае, уровень сложности вопроса понизится [5].

В подпункте 1.2.2 «Пирамидальная стратегия» рассмотрен алгоритм пирамидальной стратегии и приведен пример прохождения тестирования. В данной стратегии тестирование для всех начинается с вопроса среднего уровня. Как при типовом алгоритме тестирования, следующий вопрос подбирается в зависимости от верности ответа тестируемого. Если верно уровень повышается, иначе понижается. Тест выполняется, пока испытуемый не даст ответы, на определенное заранее количество вопросов. Для осуществления тестирования с данной стратегией необходимо при имеющемся количестве уровней сложности, обозначить сколько вопросов будет для каждого отдельного уровня.

В подпункте 1.2.3 «Метод половинного деления» рассмотрен алгоритм

адаптивного тестирования методом половинного деления, который базируется на Items Response Theory, переводится как теория тестовых заданий. IRT это совокупность методов, с помощью которых можно произвести оценку вероятности верного ответа тестируемого, при ответах на вопросы различного уровня сложности.

В подпункте 1.2.4 «Нечеткие множества» рассмотрен алгоритм пирамидальной стратегии и приведен пример прохождения тестирования. Применение методов нечеткой математики позволяет подробно описывать все составляющие вопроса и самого теста. Оценка результата прохождения теста при таком подходе, будет выполнена более качественно.

В подпункте 1.2.5 «Варьирующая ветвящаяся модель алгоритма» рассмотрен принцип варьирующей ветвящейся стратегии. Данный принцип является в предъявлении вопросов по алгоритму, который прогнозирует оптимальную трудность последующего задания по результатам выполнения испытуемым предыдущего тестового вопроса. Алгоритм, реализующий варьирующую стратегию адаптивного тестирования, носит циклический характер.

Сопоставление различных видов адаптивного тестирования с позиции их эффективности для решения проблем контроля в дистанционном образовании приводит к выбору компьютерного адаптивного тестирования, основанного на многошаговых варьирующих стратегиях, в качестве основного подхода [6].

Наиболее важное преимущество варьирующих стратегий связано с возможностью оперативного реагирования на результаты выполнения учебных заданий путем переоценки уровня подготовленности обучаемого после выполнения каждого очередного задания адаптивного теста. Последнее обстоятельство порождает уникальные возможности в решении проблем индивидуализации при обучении и контроле в дистанционном образовании.

В параграфе 1.3 «Начальный и завершающий этап тестирования» рассматривается начало и завершение адаптивного тестирования. Определение уровня сложности вопроса при входе в тестирование может быть рассчитан разными способами. Существует несколько основных подходов для выбора начального уровня вопроса.

- чтобы выявить оценку уровня подготовки, ученики проходят предварительное общее тестирование с одинаковыми вопросами и уже после начинают прохождение адаптивного теста с уровня вопроса, приближен-

ного к их уровню знаний;

- оценка уровня знаний ученика проводится постоянно, обновляясь после каждого прохождения теста;
- если у тестируемого по какой либо причине отсутствует предварительная оценка уровня знаний, то по умолчанию первый вопрос выдается среднего уровня сложности [7].

Адаптивное тестирование может завершится по нескольким определенным критериям. Если испытуемый хорошо справляется с вопросами трудного уровня сложности, либо плохо с вопросами из легкого уровня сложности. Так же тестирование может завершится, если закончились вопросы или истекло время тестирования [8].

В параграфе 1.4 «Существующие системы тестирования» перечислены и кратко описаны наиболее известные компьютерные системы тестирования: ACT–Тест, eLearning Office, М–Тест, Гефест, IRT.

Во втором разделе «Разработка веб-приложения» бакалаврской работы описывается процесс разработки веб–приложения. Описаны основные функции приложения, изложенные в требованиях.

В нашем веб-приложении для тестирования необходимо создать модели для хранения данных об объектах и представления. Для взаимодействия моделей и представлений будут добавлены контроллеры. Будет реализована авторизация и добавлена возможность разграничения по ролям. Присваивать пользователям определенную роль, может только администратор. Также необходимо добавить возможность просмотра отчетности. Для того, чтобы преподаватель имел возможность посмотреть, как с тестированием справилась группа или отдельный ученик.

Для самого прохождения и подсчета результатов теста необходимо реализовать алгоритм адаптивного тестирования, для более точной оценки усвоения материала учениками. У каждого уровня имеется список вопросов, где каждому вопросу присвоено числовое значение, которое означает насколько сложный вопрос для определенного уровня. Определять уровень и сложность вопроса будет преподаватель опираясь на свое субъективное мнение.

Тестирование будет проходить следующим образом. Первый вопрос выдается среднего уровня с максимальной сложностью. Если студент ответит верно, то следующий вопрос будет выдан сложного уровня. В случае невер-

ногого ответа, студенту выдается вопрос легкого уровня с максимальным имеющимся значением и так далее. Тестирование заканчивается при повторном верном ответе на вопросы сложного уровня, либо повторном неверном ответе на легком уровне, либо пока все вопросы не закончатся.

Расчет оценки тестируемого будет вычисляться в процентном соотношении набранных баллов ко всем возможным.

В параграфе 2.1 «Выбор среды разработки» описывается как лучше реализовать систему адаптивного тестирования. При выборе среды разработки приложения на языке программирования C#, нельзя не обратить внимание на ASP.NET – фреймворк для создания веб приложений и сервисов с использованием .NET и C# [9].

Так как адаптивное тестирование разработано в целях обучения для более точной проверки знаний, то могу предположить, что веб–приложение подойдет для такой системы лучше всего. Веб–приложение можно открыть практически на любом устройстве и оно не требует предварительной установки на устройство, с которого будет запущено приложение.

В параграфе 2.2 «ASP.NET MVC» рассматривается концепция паттерна MVC. Для реализации веб-приложения было решено остановится на ASP.NET Core MVC. Платформа ASP.NET Core MVC представляет собой фреймворк для создания сайтов и веб-приложений с помощью реализации паттерна MVC.

Концепция паттерна (шаблона) MVC (model - view - controller) предполагает разделение приложения на три компонента:

- Контроллер (controller) представляет класс, обеспечивающий связь между пользователем и системой, представлением и хранилищем данных.
- Представление (view) - это собственно визуальная часть или пользовательский интерфейс приложения.
- Модель (model) представляет класс, описывающий логику используемых данных.

Такое распределение обязанностей позволяет масштабировать приложение в контексте сложности, так как проще писать код, выполнять отладку и тестирование компонента (модели, представления или контроллера) с одним заданием.

ASP.NET Core MVC предоставляет основанный на шаблонах способ создания динамических веб-сайтов с четким разделением задач. Она обеспечивает

ет полный контроль разметки, поддерживает согласованную с TDD разработку и использует новейшие веб-стандарты.

В параграфе 2.3 «Регистрация и вход в приложение» описывается каким образом реализована регистрация и вход в приложение. Для того чтобы воспользоваться ресурсом, необходимо зарегистрироваться и войти в приложение. В данном приложении подключены сервисы Identity. Поэтому в контроллере через конструктор мы можем ими воспользоваться. В данном случае мы получаем сервис по управлению пользователями - UserManager и сервис SignInManager, который позволяет аутентифицировать пользователя и устанавливать или удалять его куки.

В параграфе 2.4 «Распределение ролей» рассматривается управление ролями в приложении. Управление распределения ролей принадлежит администратору

веб-приложения. В его полномочиях находится редактирования списка ролей. Администратор может добавлять новые роли и удалять существующие. Для того чтобы присвоить определенную роль или несколько ролей пользователю, нужно нажать на кнопку «Права доступа» напротив пользователя, которому хотим присвоить роль. В post-версии изменения ролей пользователя мы ожидаем получить id пользователя и список выбранных ролей. С помощью операции разности множеств получаем добавленные и удаленные роли и далее применяем к ним методы userManager.AddToRolesAsync() и userManager.RemoveFromRolesAsync().

В параграфе 2.5 «Описание возможностей работы с моделью Question» рассматривается детально класс Question и взаимодействие с ним контроллера и представления. Класс Question содержит:

- поле Id, которое является обязательным для первичного ключа базы данных;
- поле Title, для хранения вопроса;
- поля FirstOption, SecondOption, ThirdOption, FourthOption для хранения вариантов ответа на вопрос;
- поле Level хранит значение уровня вопроса;
- поле SLevel хранит значение сложности вопроса;
- CorrectAns хранит номер правильного варианта ответа;
- поле Test для привязки вопроса к тесту.

Для данной модели создано представление и контроллер. В основном в контроллере вопроса описаны методы создания, удаления, редактирования. В представлении описано в каком виде пользователь увидит данные на веб-странице. То есть, если преподаватель, захочет создать вопрос, то он перейдет на страницу создания и редактирования вопроса.

В параграфе 2.6 «Описание возможностей работы с моделью Test» рассматривается класс Test. Класс Test содержит:

- поле Id, которое является обязательным для первичного ключа базы данных;
- поле Title, для хранения названия теста;
- список вопросов для теста.

В обычном представлении на веб-странице для создания и редактирования тестов мы выводим списком все тесты. Данная страница доступна только преподавателям. Находясь на этой странице мы можем создать новый тест. Тогда мы перейдем в форму заполнения поля названия для теста.

В параграфе 2.7 «Прохождение теста» описывается реализация тестирования. Прохождение тестирования доступно только ученикам. Чтобы пройти тест нужно войти в систему с домашней страницы под пользователем с ролью ученик. При выборе варианта ответа и нажатии кнопки «Отправить», данные о тесте и вопросе передаются в контроллер NextQuestion. В нем происходит сравнение верного ответа и ответа студента, если верно то уровень повышается до возможного предела, либо понижается. Дальше, опять же с помощью асинхронных методов ищем последний удовлетворяющий вариант из списка вопросов, и отправляем выбранный вопрос в представление тестирования. После прохождения тестирования, нас переносит на страницу где представлен полученный результат.

В параграфе 2.8 «Формирование отчетов» описывается каким образом формируются отчеты о прохождении тестирования. Для записи данных студента о прохождении тестирования существует модель результата ответа на вопрос, которая описана в классе ResultInfo. После того, как студент пройдет тестирование, информация о его прохождении теста, будет записана и сохранена в виде отчета . Подобным образом формируются отчеты по студентам, принадлежащих к выбранной группе.

ЗАКЛЮЧЕНИЕ

В ходе данной работы была поставлена и достигнута задача создания веб-приложения системы адаптивного тестирования. Изучены различные алгоритмы адаптивного тестирования. На основе выявленных требований были спроектированы модели данных и определены взаимодействия между ними. В качестве одной из задач был изучен ASP.NET Core для C#, средства которого помогли разработать удобную онлайн систему тестирования.

Таким образом, была достигнута цель работы и выполнены все поставленные задачи. Результатом работы стало веб-приложение для системы адаптивного тестирования.

В дальнейшем планируется расширение функционала системы. Добавление различных форм тестовых заданий. Улучшения алгоритма адаптивного тестирования, для более точной оценки результатов. Также планируется внедрение полноценной системы анализа статистических данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Николаева, Е. Н. Адаптивная система тестирования / Е. Н. Николаева // *Традиции и инновации в строительстве и архитектуре. Естественные науки и техносферная безопасность.* — 2016. — Т. 4. — С. 50–53.
- 2 Литовкин, Д. В. Модели и алгоритмы для программного обеспечения системы адаптивного тестирования / Д. В. Литовкин, А. И. Потапов // *Известия Волгоградского Государственного Технического Университета.* — 2012. — Т. 4. — С. 132–138.
- 3 Яворский, В. В. Модели адаптивного компьютерного тестирования / В. В. Яворский, Ш. А. Ашkenова, А. В. Баширов // *Международный журнал экспериментального образования.* — 2016. — С. 39–41.
- 4 Малыгин, А. А. Стратегии и алгоритмы реализации адаптивных технологий педагогических измерений / А. А. Малыгин // *Вестник университета.* — 2013. — Т. 9. — С. 393–402.
- 5 Калинин, А. Г. Алгоритм реализации адаптивного компьютерного тестирования // Наука сегодня: проблемы и пути решения. — Vol. 1. — Вологда, Россия: 2019. — Рп. 80–82.
- 6 Глебов, В. А. Адаптивное тестирование как современное средство контроля результатов обучения / В. А. Глебов // *Актуальные вопросы современной науки.* — 2012. — С. 75–87.
- 7 Павлова, Р. С. Принципы адаптивного тестирования в оценивании достижений обучающихся / Р. С. Павлова, М. Е. Захарова, В. В. Загорский // *Теоретические и прикладные аспекты современной науки.* — 2008. — Т. 4. — С. 63–67.
- 8 Колдаев, В. Технология применения адаптивного тестирования it-специалистов коммерческой фирмы. / В. Колдаев, А. Волков, Р. Павлов // *Научный журнал.* — 2017. — Т. 5. — С. 83–86.
- 9 ASP.NET Free. Cross-platform. Open source. [Электронный ресурс]. — URL: <https://dotnet.microsoft.com/apps/aspnet> (Дата обращения 03.05.2020). Загл. с экрана. Яз. рус.