

Вывод. Очевидно, что рассмотренные итерационные алгоритмы позволяют эффективно бороться с краевыми эффектами и чрезмерным усилением шумов при восстановлении изображений. Итеративный процесс всегда можно остановить, если шум и осциллирующая помеха на изображении резко усиливаются. А использование операторов ограничения позволяет сократить количество вычислений и повысить точность восстановления.

СИСТЕМА ОНЛАЙН-ТЕСТИРОВАНИЯ SOTS

М. А. Мартыневский, А. А. Воробьёв

Белорусский национальный технический университет, г. Минск

Научный руководитель О. В. Дубровина

В данный момент в системе образования остается достаточно актуальным вопрос проверки знаний учащихся. В недавнем времени была введена новая форма оценки знаний – тестирование. Тестирование получило большое распространение в Беларуси: различные тесты успеваемости в школах и университетах, тестирование при поступлении в ВУЗ, огромное количество психологических тестов. Достаточно актуальным стоит вопрос о подготовке к централизованному тестированию. Однако в условиях глобальной компьютеризации общество ищет ещё более совершенные методы оценки знаний. Впервые с электронным тестированием мы столкнулись на примере софта-репетитора – справочная система, которая позволяет обобщить и впоследствии проверить свои знания. Подобные тесты были очень неудобны в использовании, обладали не самым лучшим интерфейсом. Однако, можно было сделать вывод, что электронное тестирование – очень удобный и универсальный способ проверки знаний, который может сгодиться не только для абитуриентов, но и для студентов.

На наш взгляд, электронное тестирование является наиболее универсальным и перспективным. Основные достоинства электронного тестирования:

- Возможность использовать один и тот же интерфейс для множества различных как по тематике, так и по сложности тестов. Это улучшает понимание и восприятие информации, а также повышает концентрацию внимания непосредственно на задании.
- Совершенно чёткие критерии оценки знаний (у компьютера нет симпатии и антипатии). Известно немало случаев, когда преподаватель выставляет оценку не только из соображений знаний, но и из личной оценки учащегося. Электронное тестирование полностью исключает подобные недоразумения.
- Легкость в использовании, доступность и универсальность. Сейчас почти в каждом доме имеется компьютер, в большинстве из них – доступ в Интернет. Следовательно, учащиеся имеют возможность готовиться к экзаменам, зачётным работам и т. п. дома, не утруждая себя поиском необходимых задачников.
- Облегчение рутинной работы преподавателя по проверке работ учеников.
- Возможность хранения результатов в отдельной базе данных с возможностью последующей печати.

Однако, на практике очень сложно найти систему, которая бы удовлетворяла интересам всех групп пользователей тестов: учащихся, студентов, преподавателей. Чаще всего нам встречается программное обеспечение, изготовленное только для одной из этих групп или непосредственно для подготовки и наоборот, проверки зна-

ний. Именно поэтому основной целью работы стало создание абсолютно универсального программного продукта, который смогут использовать не только студенты, но и преподаватели. Уже сейчас нарабатывается база тестов отдельно для тренировки и отдельно для оценки знаний. Разница между ними в том, что первые будут постоянно доступны студентам, вторые – преподавателям. Среды разработки PHP, HTML, MySQL были выбраны не случайно. Здесь мы опять делаем упор на универсальность и вседоступность, именно поэтому наша система – онлайн-приложение. Однако мы имеем возможность использовать систему локально. Для этого достаточно установить виртуальный сервер (Apache). Таким образом доступ к базе данных тестов будет открыт в локальной сети БНТУ.

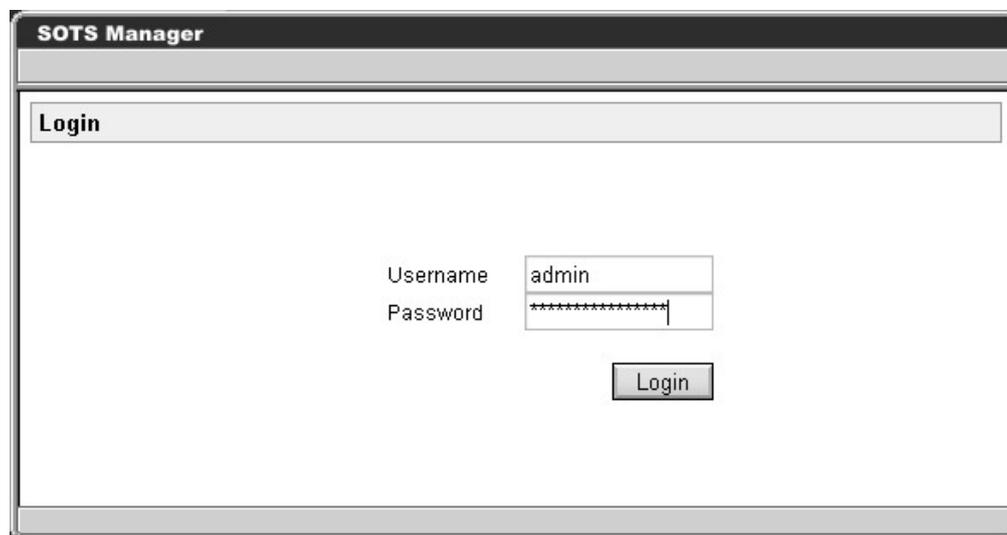
Итак, было принято решение о создании универсального программного продукта, который бы решал следующие намеченные задачи:

- Максимально удобный и простой интерфейс, доступный и понятный даже неопытному пользователю.
- Простой редактор для создания/добавления новых тестов.
- Возможность использования в качестве инструмента для проверки знаний студентов преподавателем.

Система добавления/редактирования максимально упрощена для того, чтобы любой преподаватель без помощи системного администратора мог добавлять тест по своему предмету в базу данных.

Итак, перейдем непосредственно к описанию самой онлайн-системы.

Система состоит из двух основных частей – редактора тестов и проигрывателя тестов.

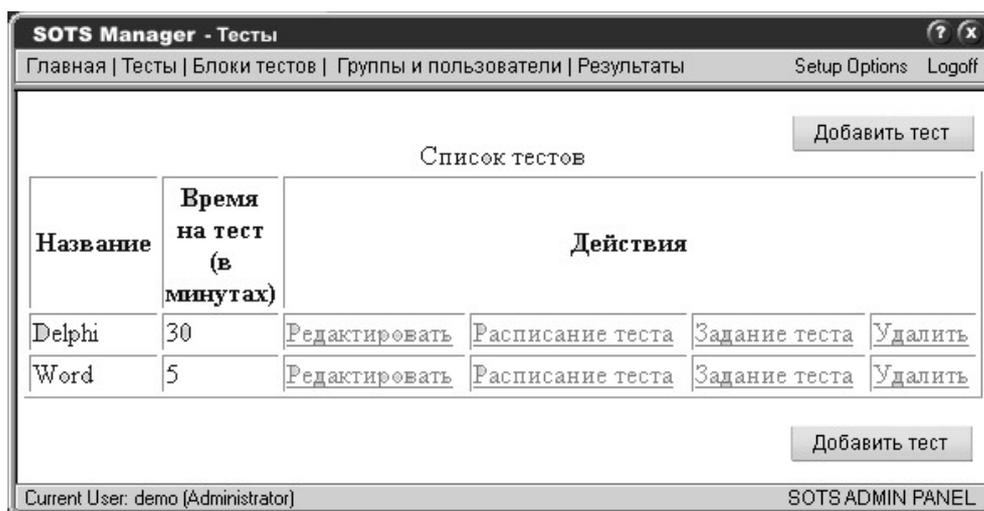


Вход в меню администрирования

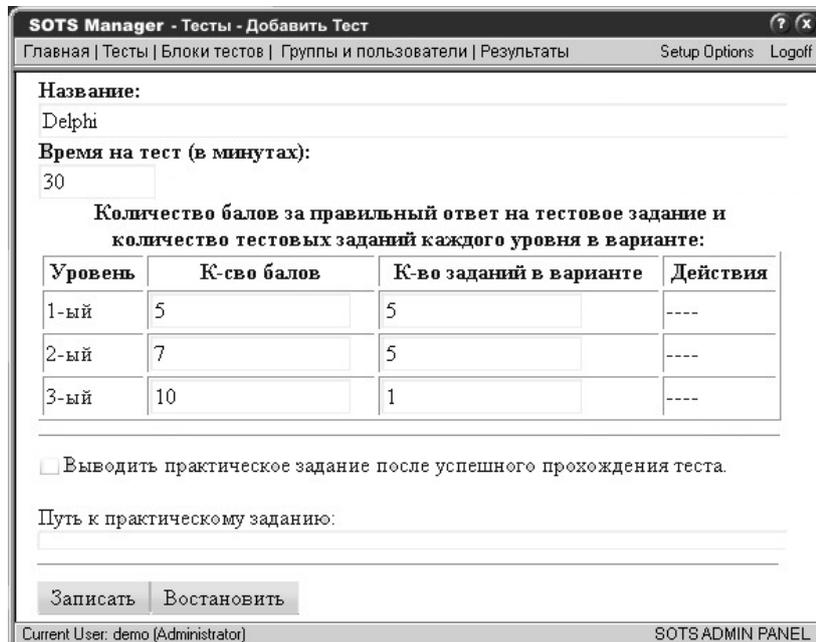
Главное меню. Возможность выбора режима работы: непосредственно начало тестирования и режим Создания/редактирования теста.

Режим создания/редактирования тестов:

Создание теста. Необходимые атрибуты: Название теста; время, отведенное на решение теста; Количество баллов за правильный ответ на тестовое задание; количество тестовых заданий каждого уровня в варианте.



Вопросы подразделяются на 3 уровня: 1, 2, 3. В вопросах 1-го уровня есть несколько вариантов ответа и только один из них может быть верным. Следовательно, нам необходимо указать как минимум 2 варианта ответа. Такие типы ответов называются открытыми. Вопросы 2-го уровня также открытые, однако правильных вариантов ответа может быть несколько. 3-ий уровень – вопросы закрытого типа, абитуриент записывает ответ в специальное поле. Для каждого уровня вопросов имеется возможность задать количество баллов, которое будет начисляться при правильном ответе.



После того как мы создали тест и внесли в него необходимые вопросы тест готов к использованию. Имеется возможность задать временной интервал, в течение которого тест будет запускаться. Это сделано для того, чтобы студенты не могли готовиться к намеченному контролю до его непосредственного наступления. Это поле можно оставить пустым. Имеется возможность объединить необходимые тесты в блоки тестов. При этом при открытии блока генерируется тест, состоящий из вопросов тестов, входящих в данный блок, т. е. если каждый тест составлен на определённый

ную тему, то с помощью блоков тестов можно создать итоговый тест, содержащий в себе вопросы на все темы, что очень удобно для подготовки контроля для студентов.

Создание групп и пользователей. Для упрощения оценки знаний студентов в университете создана возможность создания групп пользователей. В группы заносится имена студентов; необходимые атрибуты для нового студента (имя студента, пароль).

Вывод результатов для каждой группы. Имеется возможность удаления плохих результатов (например <40 баллов) одной кнопкой и создание версии для печати.

Для полноценной работы системы необходимы: веб-сервер с поддержкой PHP, MySQL на стороне сервера и веб-браузер с поддержкой JavaScript на стороне клиента.

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ РЕФЛЕКСИИ В MICROSOFT.NET ДЛЯ ОПТИМИЗАЦИИ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ ПАМЯТИ НА ОСНОВЕ ПРИМЕНЕНИЯ МЕТАДААННЫХ

В. В. Бухвальд

Белорусский национальный технический университет, г. Минск

Научный руководитель А. Б. Севрук

Метаданные являются основным фундаментом разработки платформы Net Framework. Метаданные описывают поля и методы типов, с помощью метадаанных сборщик мусора определяет достижимые объекты, т. к. метаданные указывают объекты, на которые ссылается данный объект. Применение метадаанных с целью простой сериализации и десериализации объекта, чтобы, например, передать его через сеть, позволяет легко создавать Web-сервисы в .Net Framework. Процесс изучения метадаанных называется отражением, иначе говоря, пользователь видит метаданные модуля или сборки, отраженные тем или иным инструментом. Отражение позволяет создать динамически расширяемые приложения. Тем самым обеспечивается создание типа и упаковка его в сборку. Такая глубина интеграции и простота её достижения не имеет аналогов среди более ранних технологий вроде Win32 и COM. Отражение позволяет методу изучать другой код и корректировать свою работу в зависимости от полученных сведений. На самом деле все специализированные атрибуты основаны на отражении. Кроме того, отражение позволяет методу корректировать свою работу в зависимости от того, кем он вызван. Так, можно реализовать метод, который будет выполнять одно действие, если вызван код этой же сборки, и совсем другое, если вызывающий код находится в другой сборке.

Метаданные – это набор таблиц. При компоновке сборки или модуля компилятор создает таблицы определений типов, полей, методов и т. д. В пространстве имен System.Reflection.FCL есть несколько типов, позволяющих писать код для отражения (т. е. для синтаксического разбора) этих таблиц. На самом деле типы из этого пространства имен предлагают модель объектов для отражения метадаанных сборки или модуля.

Типы, составляющие эту модель объектов, позволяют легко перечислить все типы из таблицы определений типов, а также получить для каждого из них базовый тип, интерфейсы и ассоциированные с ним флаги. Остальные типы из пространства имен System.Reflection позволяют запрашивать поля, методы, свойства и события типа путем синтаксического разбора соответствующих таблиц метадаанных. Reflection позволяют запрашивать таблицы определений в метадаанных, однако типов, запрашивающих таблицы ссылок, нет. Типы отражения также не поддерживают чте-