

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа экономики и управления
Кафедра «Информационные технологии в экономике»

РАБОТА ПРОВЕРЕНА

Рецензент,

_____/Сапожников А.И./

«_____» _____ 2019 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.т.н., с.н.с

_____/Б.М.Суховилов/

«_____» _____ 2019 г.

Разработка конструктора алгоритмов для создания адаптивных тестов

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ–09.03.03.2019.032 ВКР

Руководитель, ст.преп.

_____/С.Ю.Нестеренко/

«_____» _____ 2019 г.

Автор

студент группы ЭУ-438

_____/К.А.Алаторцева/

«_____» _____ 2019 г.

Нормоконтролер, доцент

_____/Е.А.Конова/

«_____» _____ 2019 г.

Челябинск 2019

АННОТАЦИЯ

Алаторцева К.А. Разработка конструктора алгоритмов для создания адаптивных тестов. – Челябинск: ЮУрГУ, ЭУ-438,74 с.,13 ил., 11 табл., библиогр. список –10 наим., 5 прил.

Разработан конструктор алгоритмов для написания адаптивных тестов, который используется для контроля знаний в обучении на платформе Moodle.

Актуальностью работы является создание плагина, в котором пользователи смогут создавать алгоритмы для адаптивного тестирования.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1 ПОСТАНОВКА ЗАДАЧИ	5
1.1 Технические требования к системе	5
1.2 Адаптивное тестирование и его стратегии	5
1.3 Платформа Moodle	9
1.4 Анализ существующих сред программирования	9
1.5 Выбор инструментария для решения задачи	13
Вывод по разделу один	18
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА	19
2.1 Проектирование и применение базы данных	19
2.2 Описание таблиц базы данных.....	20
2.3 Построение DFD-модели системы.....	22
2.4 Описание работы приложения	24
Вывод по разделу два	29
3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ	30
3.1 Расчет затрат	30
3.2 Расчет цены продукта	31
ЗАКЛЮЧЕНИЕ	32
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	33
ПРИЛОЖЕНИЕ А Код страницы данных алгоритмов	34
ПРИЛОЖЕНИЕ Б Код страницы функции	44
ПРИЛОЖЕНИЕ В Код страницы аргументов.....	48
ПРИЛОЖЕНИЕ Г Код прописанных функций.....	53
ПРИЛОЖЕНИЕ Д Код страницы созданных алгоритмов	60

ВВЕДЕНИЕ

В настоящее время существует большое количество платформ дистанционного и смешанного образования. Большим спросом пользуются системы с открытым ключом, т.е. исходный код таких программ доступен для просмотра, изучения и изменения, что позволяет пользователю доработать программу, использовать код для создания новых программ и исправления в них ошибок. Среди всех систем самой популярной является система Moodle. Она на равных соперничает с мировыми флагманами рынка систем дистанционного образования.

Не все пользователи обладают навыками программирования, но, так как сейчас все находится в сети интернет, возможность создавать собственные алгоритмы тестирования просто необходима.

В соответствии с этим поставлены следующие задачи:

- проанализировать существующие модели адаптивного тестирования;
- выбрать среду программирования, для создания алгоритмов;
- разработать редактора алгоритмов;
- создать алгоритм адаптивного тестирования;
- отладить и протестировать функционал готового алгоритма.

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Технические требования к системе

Требуется разработать собственный конструктор, в котором непрограммирующие пользователи смогут самостоятельно создавать алгоритмы для адаптивных тестов.

Основные требования к программе:

- хранение информации с помощью единой базы данных для информационной системы;
- разработка удобного пользовательского интерфейса клиентского приложения для ввода данных алгоритмов, обеспечивающего легкое восприятие и обработку информации, а также построения алгоритмов;
- добавление, изменение алгоритмов;
- возможность загружать и сохранять алгоритмы в базу;

1.2 Адаптивное тестирование и его стратегии

Под адаптивным тестовым контролем понимают компьютеризованную систему научно обоснованной проверки и оценки результатов обучения, обладающую высокой эффективностью за счет оптимизации процедур генерации, предъявления и оценки результатов выполнения адаптивных тестов. Разложение проблемы на составляющие приведено в таблице 1.

Таблица 1 – Описание проблемы

Описание проблемы	Результаты стандартизированных тестов фиксированной длины не отражают полную картину знаний конкретного человека
Затрагивает	Испытуемые (студенты), сталкивающиеся с трудностью преодоления стандартизированных тестов; проверяющие (преподаватели), которым трудно находить индивидуальный подход к испытуемым
Ее следствием является	Затруднение оценивания знаний студентов
Успешное решение	Применение адаптивного тестирования. Смысл адаптивности в алгоритме тестирования в том, что тест адаптируется, настраивается на доступный испытуемому уровень сложности и тем самым быстрее находит его истинный тестовый балл.

На сегодняшний день существуют некоторые среды, которые помогают непрограммирующим пользователям и пользователям более опытным одинаково легко создавать алгоритмы, чтобы в дальнейшем с ними могли работать[11].

В последнее время вниманию педагогов предлагается достаточно большое количество разнообразных программных средств для разработки тестов и тестирования. Однако многие из них не могут реализовать современные требования к качеству педагогических контрольных материалов (ПКМ), т.к. сами не удовлетворяют требованиям, предъявляемым к системам компьютерного тестирования:

- возможность использования четырех форм заданий классического педагогического теста;
- корректировка и перекомпоновка тестовых заданий в зависимости от итогов статистической обработки результатов тестирования;
- защита результирующих матриц от несанкционированного доступа.

- Кроме того, с точки зрения педагога, в системах компьютерного тестирования хотелось бы иметь следующие возможности:

- применение мультимедийных технологий при тестировании.
- в большинстве тестовых оболочек задания представляются в виде текста (иногда с применением графики). Мультимедийные тестирующие системы объединяют текстовые, графические, анимационные и видеоматериалы в наиболее эффективных сочетаниях и используют одновременно все каналы общения для передачи информации: текст, изображение и звуковое сопровождение. Озвучивание вопросов и вариантов ответов позволяет исключить ошибки испытуемого при неверном прочтении задания; а в дисциплинах, связанных с изучением иностранных языков подача материала в аудио-форме является обязательной;

- использование псевдотестовых заданий, например, цепных, текстовых, ситуационных и даже не тестовых, например, кроссвордов, ребусов и т.п.;

- применение подготовленного теста не только для контроля, но и для самоконтроля знаний. В этом случае студент после выполнения такого теста получает сведения об успешности своих действий, а после окончания самоконтроля может вновь вернуться к заданиям, на которые давал неправильные ответы и попробовать снова ответить;

- использование алгоритмов адаптивного тестирования, обуславливающих выбор очередного задания в зависимости от ответов тестируемого на предыдущие вопросы;

- применение гипертекстовых ссылок в режимах самоконтроля и тренинга.
- Перечисленные выше дополнительные возможности расширили бы сферу применения систем компьютерного тестирования[11].

Одним из определяющих успех моментов создания тестов является правильный выбор технических и программных средств.

Ограниченные возможности работы не позволяют более детально и глубоко рассмотреть многие важные проблемы, связанные с тестированием. Краткая характеристика свойств компьютерного тестирования.

Адаптивность – способность системы приспосабливаться к изменяющимся условиям (техническим и программным средствам).

Открытость определяется способностью системы под воздействием квалифицированного пользователя подстраиваться под контроль конкретных учебных дисциплин.

Стандартность системы выражается использованием функций, дизайна и др., применяемых в программах общего пользования. Подготовленный пользователь чувствует себя более комфортно, а неподготовленный пользователь может использовать полученный опыт при работе с другими программами.

Возможность расширения и наращивания системы также является важной ее характеристикой. Ее обеспечение создает пользователю уверенность в дальнейшем, продолжительном использовании системы, в ее модификации, а также в применении различных решений по ее совершенствованию [10].

Если учитывать все перечисленные свойства, то в результате сформирована система, работая с которой студенты получают возможность в режиме самоконтроля проверить свои знания по каждой теме учебной дисциплины в удобном индивидуальном темпе, выявить пробелы и затем их устранить.

Большую роль при проверке знаний играет объективность, точность результатов и минимальная вероятность погрешности оценок, исключение влияния каких-либо субъективных факторов, а также практически одинаковость для всех студентов условий тестирования. Обеспечение глубины и полноты контроля также достигается предложением студенту ответить на несколько сотен вопросов. Это как минимум на порядок превышает подобные значения при традиционной проверке знаний. Контроль проводится непосредственно после завершения изучения каждого раздела учебной программы. Преподаватель получает оперативную и объективную информацию о результатах освоения студентами данного раздела [10]. Следовательно, полученные данные могут использоваться для внесения соответствующих корректив, в содержание и методику учебного процесса.

1.3 Платформа Moodle

Moodle – система управления курсами (электронное обучение), известная как система управления обучением или виртуальная обучающая среда. Является аббревиатурой от англ. Modular Object-Oriented Dynamic Learning Environment (модульная объектно-ориентированная динамическая обучающая среда)[10]. Представляет собой свободное веб-приложение (распространяющееся по лицензии GNU GPL), предоставляющее возможность создавать сайты для онлайн-обучения.

Система бесплатна, предоставляет множество различных обучающих функций, удобна и проста в использовании, что позволяет ей быть используемой более чем в 2000 странах мира[10].

Используя Moodle, преподаватель может создавать курсы, наполняя их содержимым в виде текстов, вспомогательных файлов, презентаций, опросников и т.п. Для использования Moodle достаточно иметь любой web-браузер, что делает использование этой учебной среды удобной как для преподавателя, так и для обучаемых. По результатам выполнения учениками заданий, преподаватель может выставлять оценки и давать комментарии.

1.4 Анализ существующих сред программирования

На сегодняшний день существуют некоторые среды, которые помогают непрограммирующим пользователям и пользователям более опытным одинаково легко создавать алгоритмы, чтобы в дальнейшем с ними могли работать. Ввиду важности и необходимости данной функции, существуют среды для программирования, которые решают данную задачу.

В настоящее время существует несколько систем, с помощью которых можно редактировать и создавать различные алгоритмы.

Среда программирования «Scratch»

Scratch – это объектно-ориентированная среда, в которой блоки программ собираются из разноцветных кирпичиков команд точно также, машины или другие объекты собираются из разноцветных кирпичиков в конструкторах Лего[1].

Scratch разрабатывался как новая учебная среда для обучения непрограммирующих пользователей. В то же время, в этой системе можно раскрыть творческие таланты, так как в ней легко можно создавать фильмы, игры, анимированные открытки; придумывать и реализовывать различные объекты, определять, как они выглядят в разных условиях, перемещать по экрану, устанавливать способы взаимодействия между объектами[1]. Такая программная среда, в основе которой лежит язык программирования Scratch, была разработана группой ученых из Массачусетского технологического института (MIT).

Интерфейс программы спроектирован и сделан в расчете на неопытных пользователей, поэтому он максимально хорошо понимается интуитивно. В среде Scratch пользователи обучаются программированию поэтапно, от простого к сложному. Пример интерфейса данной среды представлен на рисунке 1.

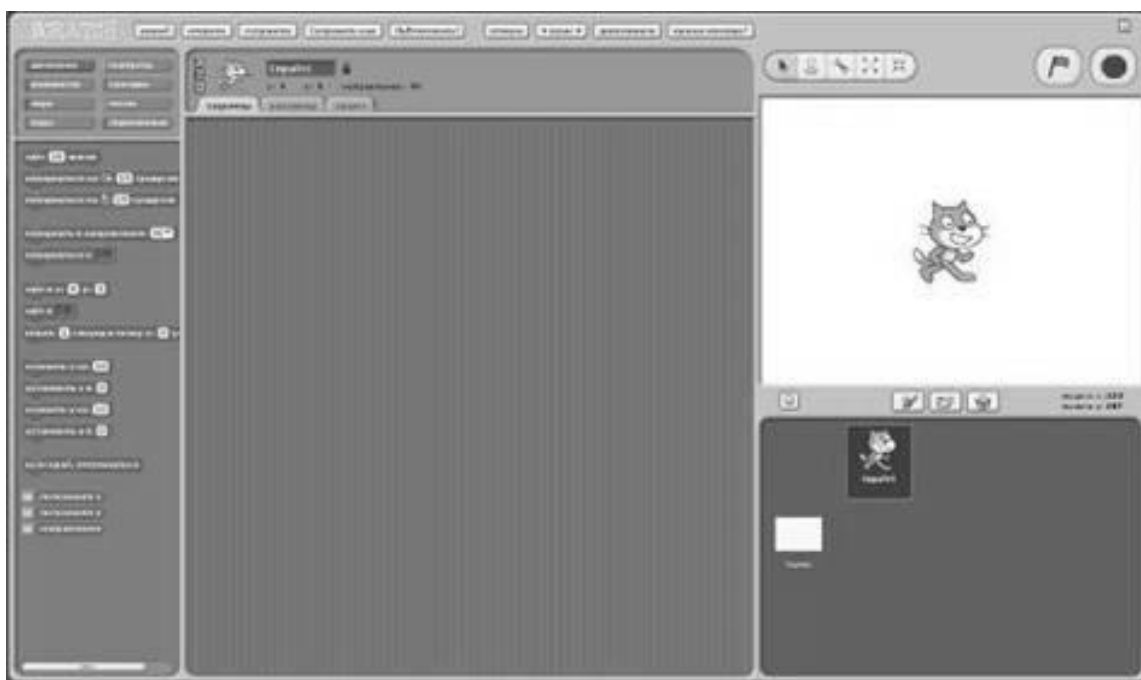


Рисунок 1– Интерфейс программы «Scratch»

На главной форме программы показано расположение готовых функций, которые нужно переместить в поле для создания алгоритмов.

Среда разработки приложений «HiAsm»

HiAsm – среда визуального программирования, программы в которой не пишутся, как это делается в классических языках, а проектируются из кубиков и

линий связей между ними. Каждый такой кубик представляет собой элемент, который выполняет какое-то достаточно простое и узкоспециализированное действие.

Интерфейс конструктора программ показан на рисунке 2.

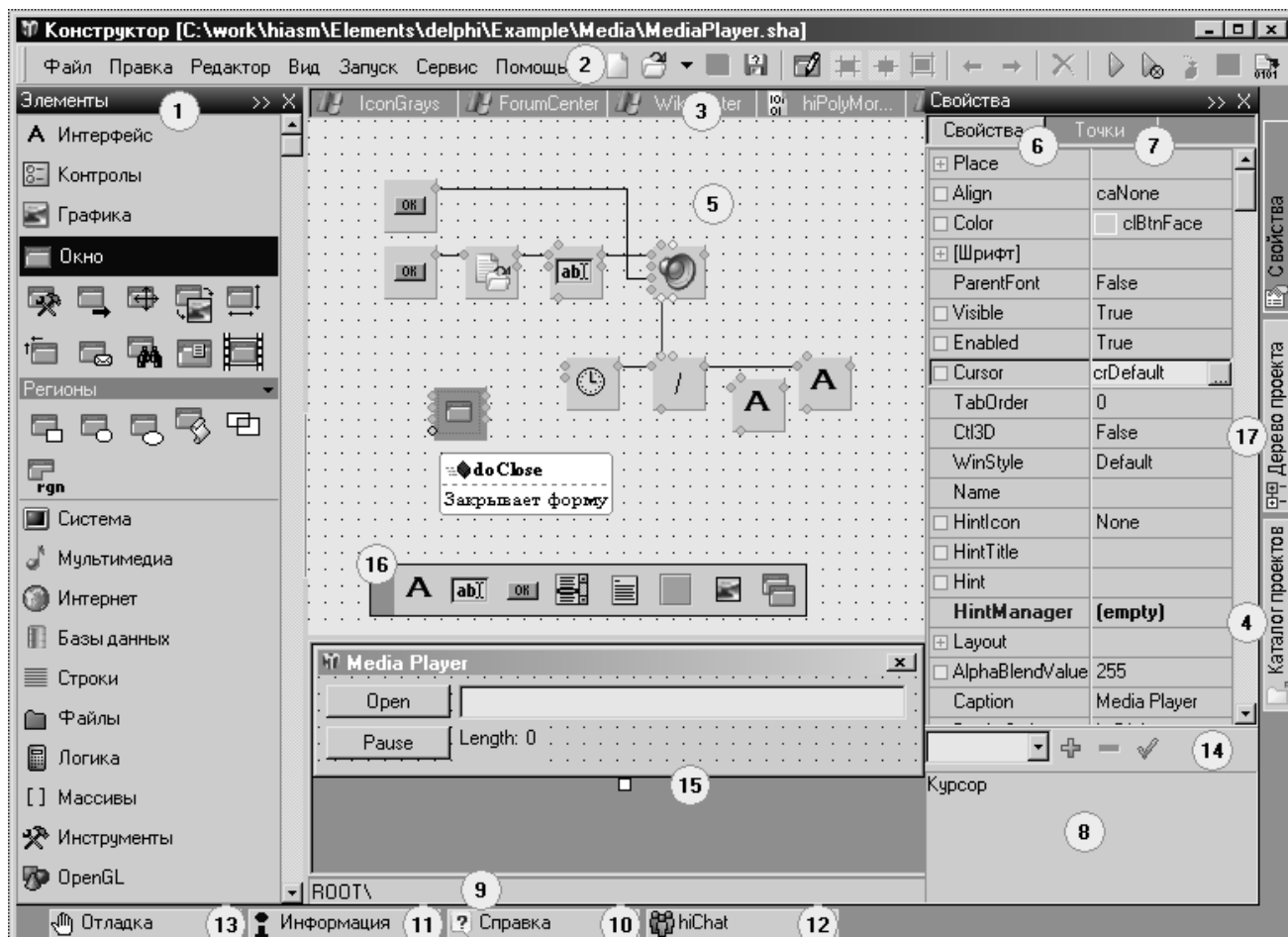


Рисунок 2– Интерфейс страницы создания алгоритмов «HiAsm»

HiAsm используется для создания программ для любой платформы: Windows, CNET, WEB, QT и другие. Установив дополнения, пользователи могут написать приложение даже для Android, IOS и других платформ[2].

Достоинства:

- возможность установки дополнений;
- кроссплатформенность;
- интуитивно понятный интерфейс;
- высокая скорость выполнения;

- официальная версия на русском языке.

Недостатки:

- не подходит для крупных проектов;
- большой объем исполняемых файлов.

Среда для создания программ на русском языке «Алгоритм»

«Алгоритм» – это среда для создания программ на русском языке, одна из немногих. Ее особенность в том, что она использует текстовое визуальное программирование. Это значит, что пользователь может создать программу, не зная языка. «Алгоритм» - конструктор, который обладает большим набором компонентов. Информацию о каждом компоненте можно найти в документации программы.

Основная страница среды «Алгоритм» показана на рисунке 3.

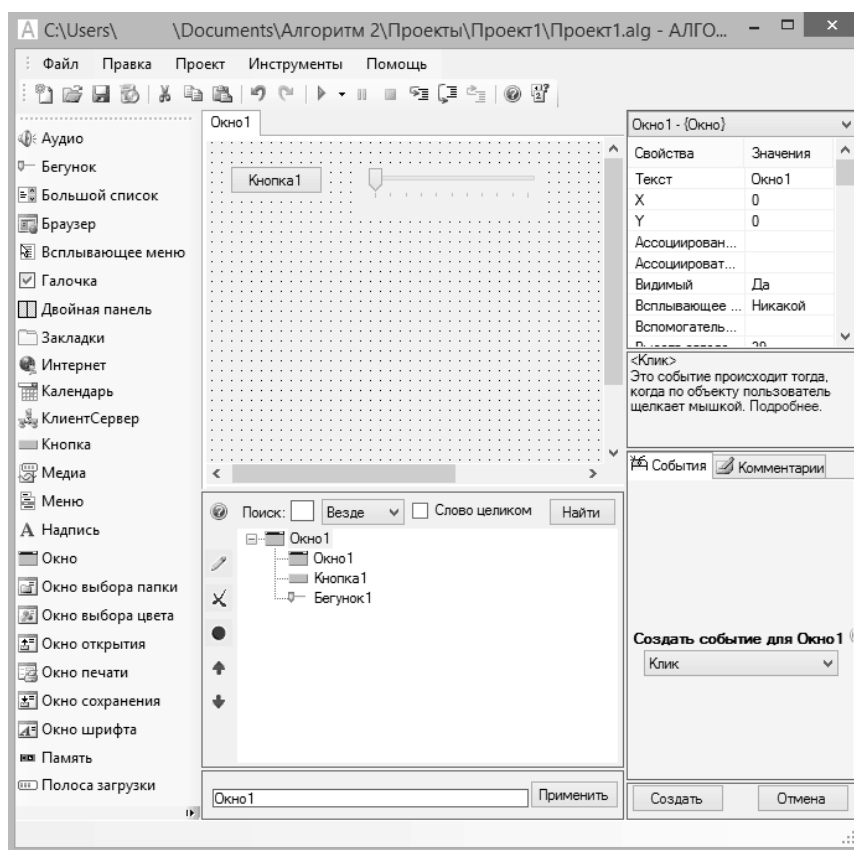


Рисунок 3 – Интерфейс среды программирования «Алгоритм»

«Алгоритм» позволяет работать с графическим модулем, но приложения с использованием графики будут выполняться довольно долго. На главной форме

находятся поле для создания графических изображений и функции, используемые для их создания.

Среде «Алгоритм», для серьезной разработки программ, не хватает возможностей, но для простых задач этот конструктор подойдет[2].

Исходя из полученного анализа, можно сравнить возможности приведенных сред программирования. Все системы удобны в использовании, как для программирующих пользователей, так и для менее опытных. Но, ни в одном решении невозможно программировать собственные алгоритмы тестирования (например, алгоритмы адаптивного тестирования).

1.5 Выбор инструментария для решения задачи

Для разработки собственной среды программирования используются технологии: HTML, CSS, JavaScript, PHP, XML, OpenServer.

Язык HTML применяется для разметки документов в Интернете. На сегодняшний день при проектировании множества страниц web-приложений применяют описание разметки на языке HTML. HTML интерпретируется браузерами, текст, который получается вследствие интерпретации, выводится на экран монитора компьютера или мобильного устройства.

HTML не является языком программирования, однако большинство страниц web-приложений включают в себя встроенные программы – скрипты на языках JavaScript.

В Интернете HTML-страницы, передаются браузерам от сервера по протоколам HTTP или HTTPS[3]. Передача может происходить в виде текста, однако, также может использоваться шифрование.

Документы, которые содержат разметку на языке HTML (обычно имеют расширение .html или .htm), обрабатываются специальными приложениями – браузерами, которые выводят документ в его форматированном виде. Они представляют для пользователя простой и интуитивно понятный интерфейс для запроса страниц web-приложений, а также их просмотра (вывод может осуществляться

на любое внешнее устройство) и, позволяют отправить данные, введенные пользователями, на сервер[6].

Язык HTML является теговым языком разметки документов. Документы, созданные на языке HTML, представляют собой набор элементов, начало и конец которых, обозначается особыми метками – тегами. Элементы так же могут не иметь никакого текста или каких-либо данных (например, тег перевода строки
). Если элемент пуст, закрывающий тег обычно не указывается. Атрибуты, которые могут иметь элементы, могут устанавливать какие-либо их свойства (например, размер шрифта для элемента font). Все атрибуты выполняются в открывающем теге[3]. CSS – формальный язык, который помогает описать внешний вид документа, написанного с помощью языков HTML и XHTML[4]. CSS применяется разработчиками страниц web-приложений для оформления документа, в частности применения цветов, задания шрифтов, расположения отдельных блоков на создаваемой странице и других вариантов оформления внешнего вида страниц web-приложений. Главной целью при создании CSS было разделение отображения логической структуры страницы web-приложения от отображения внешнего вида web-страницы (выполняется при помощи формального языка CSS). Это разделение помогает повысить доступность документа, обеспечить большую гибкость и возможность управления его представлением, и уменьшить сложность и повторяемость в структурном содержимом. CSS позволяет продемонстрировать один и тот же документ в разных стилях или методах вывода, например, печатное или экранное представление, либо чтение голосом (с помощью специального голосового браузера), или при выводе устройствами, которые используют шрифт Брайля[6].

JavaScript – прототипно-ориентированный сценарный язык программирования. JavaScript это диалект языка ECMAScript.

JavaScript обычно применяется как встраиваемый язык для программного доступа к объектам приложений. Широкое применение JavaScript находит в браузерах как язык сценариев, для того, чтобы придать страницам web-приложения интерактивность[5].

Благодаря JavaScriptразработчики могут управлять тем, как web-страница будет себя вести.

Основные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

JavaScript этообъектно-ориентированный язык, ноприменяемое в языке прототипирование обуславливает различия в работе с объектами в сравнении с традиционными класс-ориентированными языками. Также, у JavaScript есть свойства, которые обычно характеризуютфункциональные языки: функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания. Всё это придаёт языку дополнительную гибкость.

Несмотря на то, что JavaScript имеет схожий с Си синтаксис, у JavaScript есть значимые отличия: объекты, с возможностью интроспекции; функции как объекты первого класса, автоматическое приведение типов, автоматическая сборка мусора, анонимные функции[5].

Однако у JavaScript есть свои минусы. В языке отсутствуют следующие возможности:

- модульная система: JavaScript не предоставляет возможности управлять зависимостями и изоляцией областей видимости;
- стандартная библиотека: в частности, отсутствует интерфейс программирования приложений по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для бинарных данных;
- стандартные интерфейсы к web-серверам и базам данных;
- система управления пакетами, которая бы отслеживала зависимости и автоматически устанавливала их[5].

В качестве формата передачи данных могут использоваться фрагменты простого текста, HTML-кода, JSON или XML[7].

PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим

большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией, несовместимой с GNU GPL.

В области веб-программирования, в частности серверная часть, PHP – один из популярных сценарных языков (наряду с JSP, Perl и языками, используемыми в ASP.NET) благодаря своей простоте, скорости выполнения, богатой функциональности, кроссплатформенности и распространению исходных кодов на основе лицензии PHP.

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств, для разработки веб-приложений. Основные из них:

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;
- взаимодействие с большим количеством различных систем управления базами данных (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, MicrosoftSQLServer, Sybase, ODBC, mSQL, IBMDB2, Cloudscape и ApacheDerby, Informix, OvrimosSQL, LotusNotes, DB++, DBM, dBase, DBX, FrontBase, FilePro, IngresII, SESAM, Firebird / InterBase, ParadoxFileAccess, MaxDB, Интерфейс PDO);
- автоматизированная отправка HTTP-заголовков;
- работа с HTTP-авторизацией; работа с cookies и сессиями.

В настоящее время PHP используется сотнями тысяч разработчиков. Согласно рейтингу корпорации ТЮВЕ, базирующемуся на данных поисковых систем, в июне 2013 года PHP находился на 5 месте среди языков программирования. К крупнейшим сайтам, использующим PHP, относятся Facebook, Wikipedia и другие[9].

Входит в LAMP – распространённый набор программного обеспечения для создания и хостинга веб-сайтов (Linux, Apache, MySQL, PHP)[3].

Для хранения таблиц использовано приложение phpMyAdmin, которое написано на PHP и обеспечивает полноценную, в том числе удаленную, работу с базами данных MySQL через браузер. Так как phpMyAdmin позволяет во многих случаях обойтись без непосредственного ввода команд SQL, то работа с базами данных становится вполне посильной задачей даже для человека, весьма поверхностно знакомого с MySQL.

Активное использование MySQL в веб-программировании обусловило его актуальность, а интуитивно понятный интерфейс в совокупности с широкой функциональностью и поддержкой более 60 языков (в т.ч. и русского) обеспечило ему популярность среди веб-разработчиков.

В связи с поддержкой инициативы GoPHP5, для работы с phpMyAdmin 3.x.x требуются предустановленные MySQL 5 и PHP 5.2. Для использования более старых версий MySQL и PHP, разработчики до сих пор поддерживают 2.x.x ветку phpMyAdmin.

Данный ресурс ориентирован на тех, у кого в ходе работы с phpMyAdmin возникают затруднения, кто только начинает знакомиться с данным приложением или хочет узнать больше о возможностях этого приложения. Пользователями данного ресурса являются веб-мастера, администраторы баз данных и программисты.

XML – рекомендованный Консорциумом Всемирной паутины (W3C) язык разметки. Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Сочетание простого формального синтаксиса, удобства для человека, расширяемо-

сти, а также базирование на кодировках Юникод для представления содержания документов привело к широкому использованию как собственно XML, так и множества производных специализированных языков на базе XML в самых разнообразных программных средствах[9].

Вывод по разделу один

Представлены технические требованиями к разрабатываемой системе, описана система Moodle, приведен анализ существующих систем, определены преимущества и недостатки. Описаны используемые технологии для разработки программы.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

2.1 Проектирование и применение базы данных

Для хранения данных, используемых в приложении, применяется база данных MySQL в PHPMyAdmin.

MySQL – это реляционная система управления базами данных. То есть данные в ее базах хранятся в виде логически связанных между собой таблиц, доступ к которым осуществляется с помощью языка запросов SQL. MySQL – свободно распространяемая система. Кроме того, это достаточно быстрая, надежная и, главное, простая в использовании СУБД, вполне подходящая для не слишком глобальных проектов. Существует очень популярный визуальный интерфейс для работы с этой СУБД. Называется он PhpMyAdmin[8]. Этот интерфейс позволяет значительно упростить работу с базами данных в MySQL.

В текстовом режиме работа с базой данных выглядит просто как ввод команд в командную строку, а результаты выборок возвращаются в виде своеобразных таблиц, поля в которых налезают друг на друга, если данные не помещаются на экран.

PhpMyAdmin позволяет пользоваться всеми достоинствами браузера, включая прокрутку изображения, если оно не умещается на экран. Многие из базовых SQL-функций работы с данными в PhpMyAdmin сведены к интуитивно понятным интерфейсам и действиям, напоминающим переход по ссылкам в Internet.

Схема данных показана на рисунке 4.

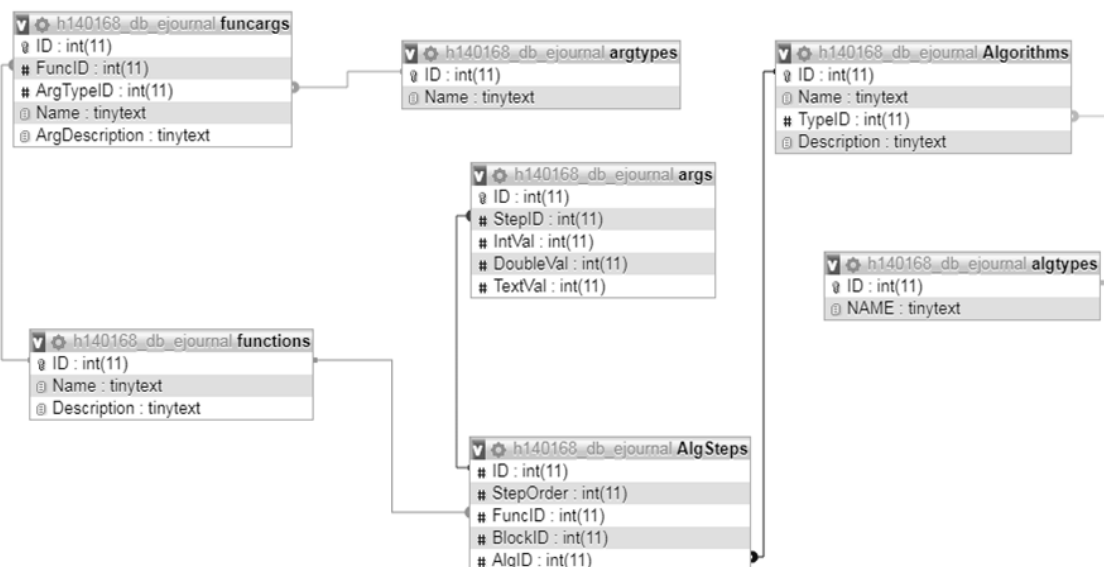


Рисунок 4– Схема базы данных

На схеме базы данных представлены таблицы, в которых хранится информация об алгоритмах и типах, о порядковых номерах алгоритмов и описание их аргументов, о функциях, а также об аргументах. Все таблицы приведены к третьей нормальной форме.

2.2 Описание таблиц базы данных

Таблица «Algorithms» содержит данные об алгоритмах. Список ее полей приведен в таблице 2.

Таблица 2 – Поля таблицы «Algorithms»

Наименование	Описание	Тип
ID	Уникальный код	Идентификатор
Name	Название	Текстовое
TypeID	Код типа алгоритма	Целое
Description	Описание	Текстовое

В таблице «AlgSteps» хранится информация о присваиваемых порядковых номерах алгоритмов. Список ее полей приведен в таблице 3.

Таблица 3 – Поля таблицы «AlgSteps»

Наименование	Описание	Тип
ID	Уникальный код	Идентификатор
StepOrder	Порядковый номер	Целое
FuncID	Название класса	Целое
BlockID	Код	Целое
AlgID	Код алгоритма	Целое

Таблица «AlgTypes» содержит информацию о типах алгоритмов. Список ее полей приведен в таблице 4.

Таблица 4 – Поля таблицы «AlgTypes»

Наименование	Описание	Тип
ID	Уникальный код	Идентификатор
Name	Название	Текстовое

Таблица «Args» представляет собой данные о фактических аргументах. Все ее поля описаны в таблице 5.

Таблица 5 – Поля таблицы «Args»

Наименование	Описание	Тип
ID	Уникальный код	Идентификатор
ArgID	Код аргумента	Целое
StepID	Код порядкового номера	Целое
IntVal	Целое значение	Целое
DoubleVal	Символьное значение	Целое
TextVal	Текстовое значение	Целое

Таблица «ArgTypes» содержит названия аргументов. Поля описаны в таблице 6.

Таблица 6 – Поля таблицы «ArgTypes»

Наименование	Описание	Тип
ID	Уникальный код	Идентификатор
Name	Название аргумента	Текстовое

Таблица «FuncArgs» содержит данные о формальных аргументах функций. Поля описаны в таблице 7.

Таблица 7– Поля таблицы «FuncArgs»

Наименование	Описание	Тип
ID	Уникальный код	Идентификатор
FuncID	Код	Целое
ArgTypeID	Код	Целое
Name	Название функции	Текстовое
ArgDescription	Описание функции	Текстовое

Таблица «Functions» содержит данные о функциях. Данные таблицы «Functions» приведены в таблице 8.

Таблица 8– Поля таблицы «Functions»

Наименование	Описание	Тип
ID	Уникальный код	Идентификатор
Name	Название функции	Текстовое
Description	Описание	Текстовое

2.3 Построение DFD-модели системы

Диаграммы потоков данных (data flow diagramming, DFD), используются для моделирования работы информационной системы.

При помощи DFD-диаграмм, требования к проектируемой ИС разбиваются на процессы и представляются в виде сети, которая связана потоками данных.

Для представления разрабатываемой системы, составлена её DFD модель. Модель представлена в двух уровнях. На первом уровне находятся три внешние сущности: тестируемый, редактор тестов и редактор алгоритмов. Первый уровень системы изображен на рисунке 5.

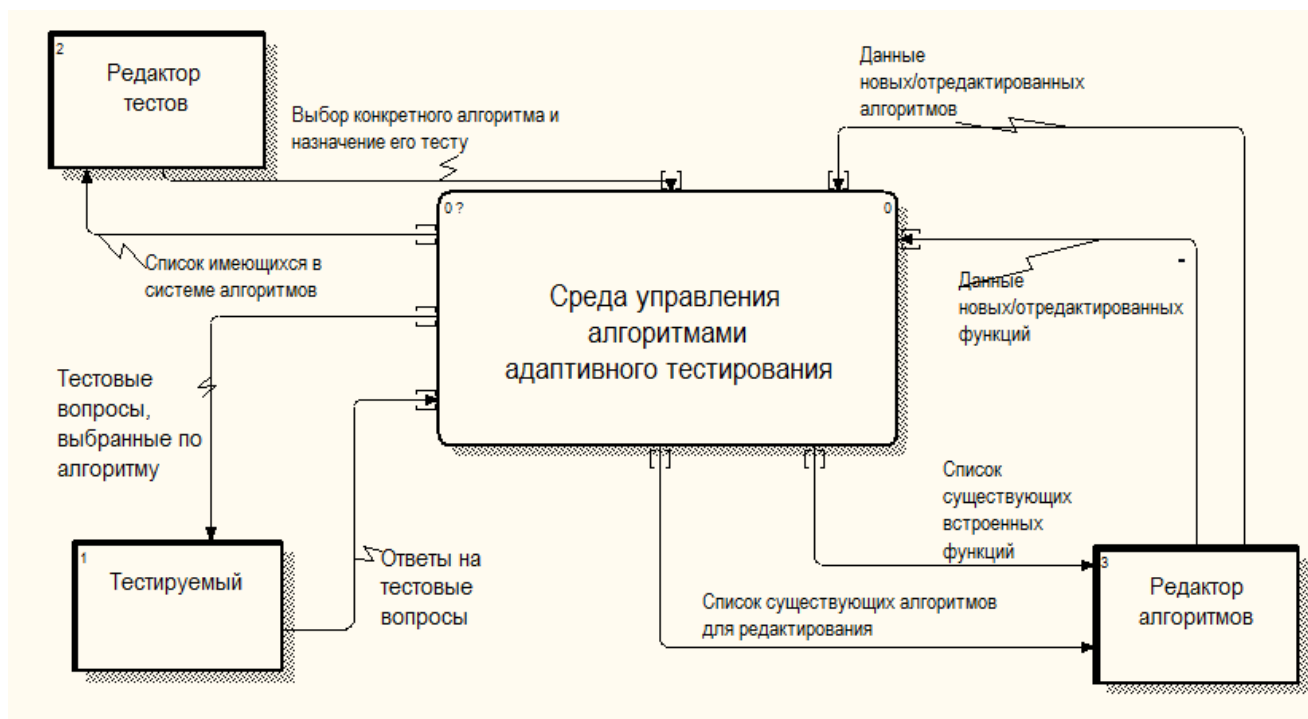


Рисунок 5 – Первый уровень DFD-модели

Среда управления алгоритмами адаптивного тестирования получает данные от сущностей. В таблице 9 представлены внешние сущности модели, у каждой из которых имеются входные или выходные данные.

Таблица 9– Внешние сущности DFD-модели

Внешняя сущность	Входные данные	Выходные данные
Тестируемый	Тестовые вопросы, выбранные по алгоритму	Ответы на опросы
Редактор тестов	Список имеющихся в системе алгоритмов	Выбор конкретного алгоритма и назначения его тесту
Редактор алгоритмов	Список существующих функций и алгоритмов для тестирования	Данные о новых или отредактируемых алгоритмах и функциях

Второй (или внутренний) уровень модели отражает заполнение хранилища данных, где содержатся сведения об интерфейсах редактирования алгоритмов тестирования и управления встроенными функциями, а также система проверки знаний и система управления тестовыми заданиями. Второй уровень DFD-модели представлен на рисунке 6.

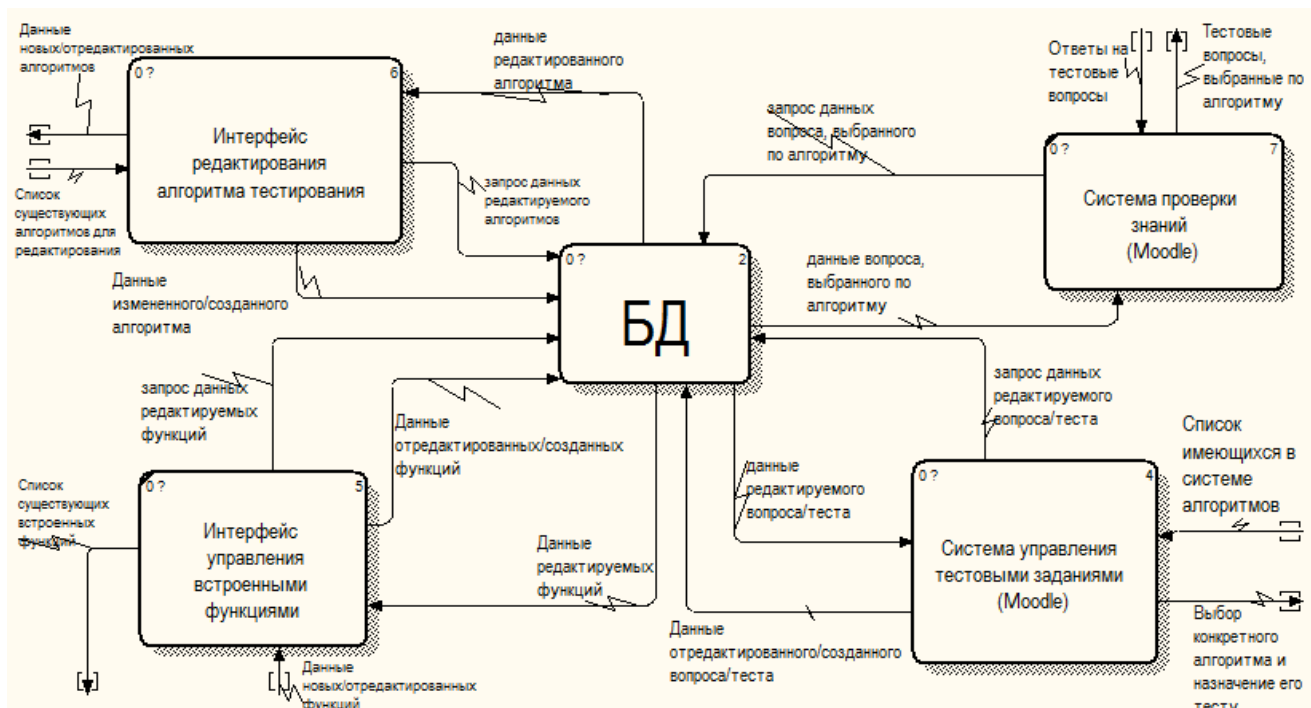


Рисунок 6 – Второй уровень DFD-модели

Заполнение базы данных происходит с помощью получения данных, которые приходят в систему управления тестовыми заданиями, в систему проверки знаний, на интерфейс редактирования алгоритма тестирования и интерфейс управления встроенными функциями.

2.4 Описание работы приложения

Для того чтобы непрограммирующий пользователь мог самостоятельно создавать алгоритмы для адаптивных тестов, нужно создать для него удобный и максимально простой интерфейс. Чтобы с помощью него просто заполнялись необходимые функции и аргументы для алгоритмов.

При создании функций (процедур) для удобства пользователей во внимание принимались следующие моменты:

- каждая функция (процедура) имеет имя, и оно является идентификатором и должно быть тем или иным способом объявлено;
- каждая функция (процедура) имеет свои коды, которые должны быть оформлены заданным языком программирования. Эти коды называются определением функции;

- для решения задачи функция (процедура) может потребовать набор аргументов (исходные данные), которые передаются ей в момент вызова;
- функция может возвращать результаты своих вычислений (возвращаемые данные), которые передаются ей в момент вызова.
- каждая функция (процедура) должна быть вызвана по имени. Если вызов отсутствует, то функция выполняться не будет.

Имя функции (процедуры) рассматривается как ее идентификатор и составляется из правил составления идентификаторов конкретного языка программирования.

Далее, из всего этого получаются готовые формы, с помощью которых заполняется база данных.

Формадобавления функции представлена на рисунке 7.

The screenshot shows a web browser window with the address bar displaying 'algedit/functions/functions.php'. The page contains a form with two input fields: 'Название:' and 'Описание:'. Below the fields is a 'Сохранить' button. At the bottom of the page is a table with three rows, each representing a function entry. The table has two columns: 'Название' and actions 'Редактировать' and 'Удалить'.

Название	Редактировать	Удалить
Увеличить сложность	Редактировать	Удалить
Уменьшить сложность	Редактировать	Удалить
Ответ успешен	Редактировать	Удалить

Рисунок 7 – Форма добавления функции

Пользователь сам вводит значения, которые будут необходимы для построения алгоритмов, с помощью которых будет создаваться тестирование. Пользователь должен ввести название функции, ее описание, далее нажать на кнопку сохранить. Если же он не введет что-либо, то на форме появится ошибка, о том, что данные введены не полностью. Данные можно редактировать и удалять.

Ошибка о заполнении данных функции представлена на рисунке 8.

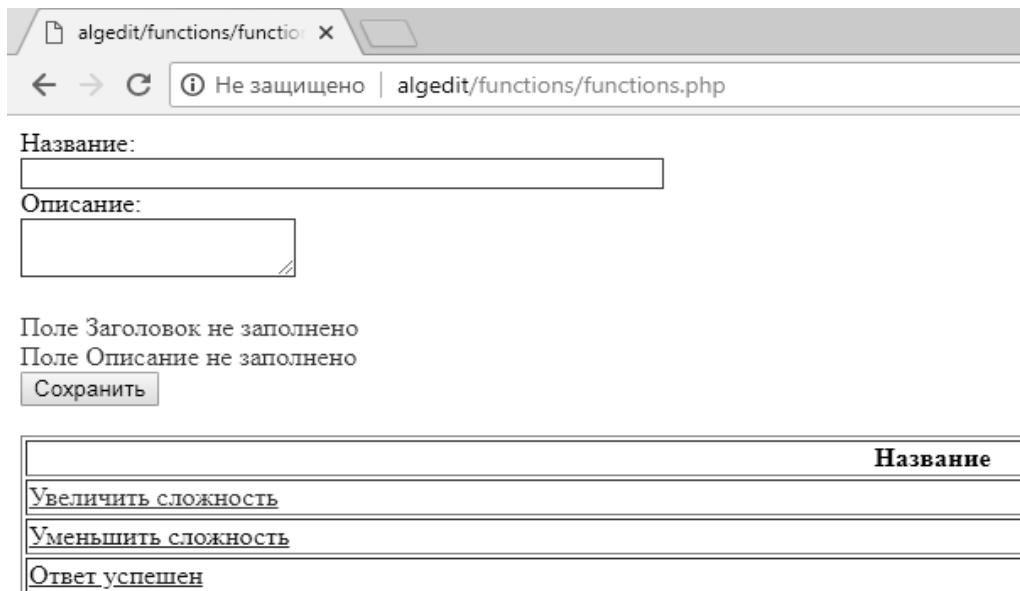


Рисунок 8 – Форма добавление функции с ошибкой заполнения

Чтобы перейти на форму подробного описания функций, нужно нажать на ее название. Откроется форма, в которой пользователь может ввести новые данные, отредактировать или удалить старые данные конкретно о выбранной функции. Для этого нужно заполнить поля название, описание и выбрать из выпадающего списка необходимый тип. Форма для добавления данных о функции показана на рисунке 9.

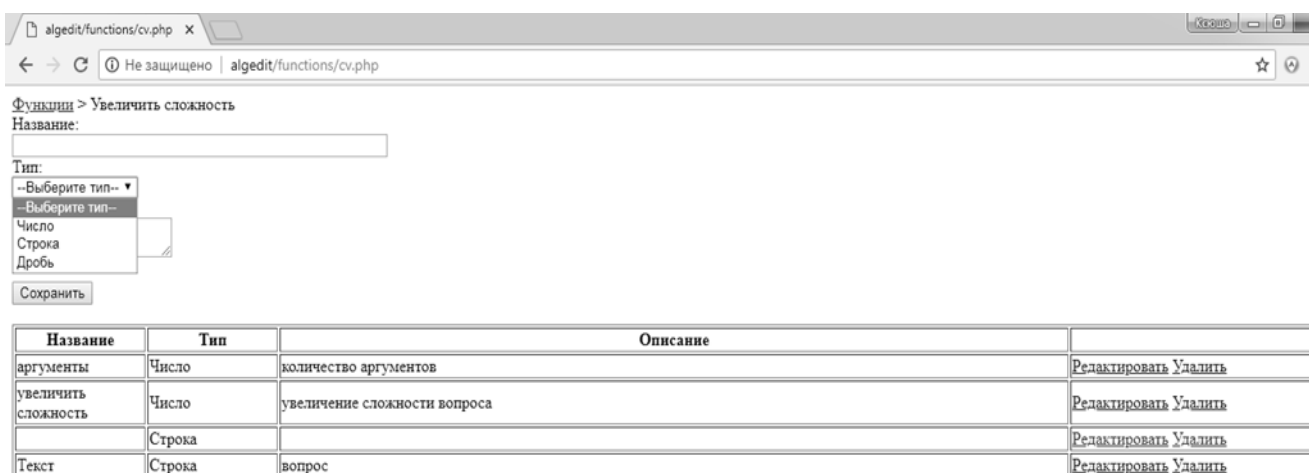


Рисунок 9 – Форма для добавления данных о функции

По аналогии с формой добавления функций, при некорректном вводе данных, будет появляться ошибка заполнения. Форма с неправильным заполнением представлена на рисунке 10.

Тип:

Описание:

Поле Секция не заполнено
 Поле <<Имя аргумента>> не заполнено

Название	Тип	Описание
аргументы	Число	количество аргументов
увеличить сложность	Число	увеличение сложности вопроса
	Строка	
Текст	Строка	вопрос

Рисунок 10 – Форма заполнения данных функции с ошибкой ввода

Пользователь должен указывать все данные, для дальнейшей работы.

На форме, представленной на рисунке 11, заполняются данные об алгоритмах. После того, как алгоритм для создания тестирования будет создан, он будет сохраняться в базу и открываться за счет данной формы, всего лишь нажав на название необходимого алгоритма.

Название:

Тип:

Описание:

Название	Тип	Описание	
Простой алгоритм	Подведение итогов тестирования	простой алгоритм с подведением итогов тестирования	Редактировать Удалить
Вывод	Вывод вопроса	Выводится вопрос определенной сложности	Редактировать Удалить
Мягкое тестирование	Подведение итогов тестирования	Тест проводится на простом уровне	Редактировать Удалить

Рисунок 11 – Форма данных об алгоритмах

Далее пользователь попадает на страницу создания самого алгоритма. Чтобы это сделать, необходимо всего лишь перенести нужные функции из панели инструментов в область алгоритма. Порядок и структура расположения функций будет зависеть от самого алгоритма. Пользователь сможет сам решать, какая функция будет первой, а какая последний. В любой момент этот порядок можно изменить, так как функции могут меняться местами. После сохранения алгоритма, он преобразуется в php код и загружается в систему Moodle для дальнейшего использования. Форма создания и готового алгоритма представлена на рисунке 12.

Рисунок 12 – Форма готового алгоритма

С помощью алгоритма мягкого тестирования пользователю выводится список вопросов в необходимом порядке.

Еще один вариант готового алгоритма представлен на рисунке 13.

Алгоритмы > Подведение итогов

область алгоритма	панель инструментов
<p>Сохранить</p> <p>ЗАПОМНИТЬ x</p> <p>Значение <input type="text" value="0"/> как <input type="text" value="sum"/></p> <p>ВЫПОЛНИТЬ ДЛЯ КАЖДОГО ОТВЕТА x</p> <p>УВЕЛИЧИТЬ ЗНАЧЕНИЕ x</p> <p>Переменной <input type="text" value="sum"/> на значение <input type="text" value="#СЛОЖН*#ПРАВИЛЬН#"/></p> <p>УСТАНОВИТЬ ИТОГОВЫЙ БАЛЛ x</p> <p>Значение <input type="text" value="#sum#/15"/></p>	<p>ВЫВОД ВОПРОСОВ</p> <p>ВЫВЕСТИ СЛУЧАЙНЫЙ ВОПРОС</p> <p>ВЫВЕСТИ ВОПРОС ЗАДАННОЙ СЛОЖНОСТИ</p> <p>ВЫВЕСТИ ВОПРОС ТЕКУЩЕЙ СЛОЖНОСТИ</p> <p>ВЫВЕСТИ ВОПРОС ПО КРИТЕРИЮ</p> <p>УВЕЛИЧИТЬ СЛОЖНОСТЬ</p> <p>УМЕНЬШИТЬ СЛОЖНОСТЬ</p> <p>МАТЕМАТИЧЕСКИЕ ФУНКЦИИ</p> <p>УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ</p> <p>МЕТАДАННЫЕ</p> <p>ВСПОМОГАТЕЛЬНЫЕ ИНСТРУМЕНТЫ</p>

Рисунок 13 – Форма готового алгоритма

Алгоритм подведения итогов рассчитывает оценку по пройденному тесту.

Вывод по разделу два

В разделе приведены выбранные средства для разработки. Создана база данных и описаны таблицы, используемые при разработке конструктора алгоритмов. Составлена DFD-модель двух уровней, в которой содержатся сущности, входные и выходные данные. Подробно описаны все страницы и все возможности программы.

3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

При принятии решения о создании и внедрении конструктора алгоритмов для создания адаптивного тестирования на платформе Moodle, рассчитана экономическая эффективность и целесообразность внедрения разработки.

Для определения эффекта окупаемости используется методика срока окупаемости, для этого рассчитываются затраты до внедрения и после внедрения проекта.

3.1 Расчет затрат

Затраты по покупке лицензионного программного обеспечения и ПК отражены в таблице 10.

Таблица 10 – Затраты на ПО и ПК

Наименование ПО	Стоимость, руб.
Антивирус Kaspersky Internet Security	1900,00
Базовые лицензии СДО Русский Moodle 3KL + техническая поддержка на 1 год	88000,00
ПК	25 000
Итого:	114900,00

Расчет заработной платы работников, разрабатывающих и внедряющих проект, приведен в таблице 11.

Таблица 11– Расчет заработной платы

Должность	Оклад, руб./мес.	Оплата, руб./день	Продолжительность работ, дни	Итого, руб.
Программист	25 000	1137	20	22 740
Итого:				22 740
Дополнительная заработная плата (20% от основной)				4548
Основная и дополнительная заработная плата				27 288

Затраты на разработку и внедрение программного продукта составляют 142188,00 руб.

3.2 Расчет цены продукта

Подсчитано, что в затраты на создание и внедрение программы в общей сумме составят 142188,00 рублей.

Вывод по разделу три

Рассчитаны статьи затраты на разработку и внедрение, вычислена цена продукта.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы были поставлены цели и задачи, обоснована актуальность и необходимость выбранной темы. Также были проанализированы существующие информационные системы.

В основном разделе работы, прежде всего, были рассмотрены выбранные средства разработки. Кроме того, были приведены схемы базы данных и описаны используемые в программе таблицы базы данных.

Была разработана структура приложения и описана работа отдельных частей программы, а именно, добавление алгоритмов, функций и аргументов с подробным описанием.

В экономической части данной работы были подсчитаны затраты на разработку.

Таким образом, цель работы – разработка конструктора алгоритмов для создания адаптивных тестов– достигнута. В полной мере решены поставленные задачи, а именно:

- анализ существующих моделей адаптивного тестирования;
- выбор среды программирования, для создания алгоритмов;
- разработка редактора алгоритмов;
- создание алгоритма адаптивного тестирования;
- отладка и тестирование функционала готового алгоритма.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Среда программирования «Scratch» – <http://progopedia.ru/language/scratch/>.
2. Среда разработки «Алгоритм» – <http://www.getchip.net/posts/sreda-programmirovaniya-algorithm-builder/>.
3. Лабберс, П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений – ProHTML5 Programming: Powerful APIs for Richer Internet Application Development / П. Лабберс, Б. Олберс, Ф. Салим. – М.: Вильямс, 2011. 272 с.
4. Ливингстон, Д. CSS и DHTML: Web-профессионалам / Д. Ливингстон, М. Браун. – Киев: BHV, 2012. 264 с.
5. Дмитриева, М.В. JavaScript / М.В. Дмитриева. – СПб.: БХВ-Петербург, 2016. 328 с.
6. Вулстон, Д. Ajax, и платформа NET 2.0 для профессионалов = ProAjax and the .NET 2.0 Platform / Д. Вулстон. – М.: Вильямс, 2007. 464 с.
7. Крейн, Д. AJAX в действии: технология – Asynchronous JavaScript and XML – Ajax in Action / Д. Крейн, Э. Паскарелло, Д. Джеймс. – М.: Вильямс, 2006. 640 с.
8. Горяка А.А. Основы ASP.NET 2.0: Учебное пособие / А.А. Горяка. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2017. 296 с.
9. Маршал, Б. XML в действии: Практика программирования / Б. Маршал. – М.: Триумф, 2012. 364 с.
10. Использование платформы Moodle – <http://moodlebook.ru/>.
11. Адаптивное тестирование: учеб.-метод. пособие / Н. М. Опарина [и др]. – Хабаровск : Изд-во ДВГУПС, 2007. – 95. : ил.

ПРИЛОЖЕНИЕ А

Код страницы данных алгоритмов

```
<? session_start();
if(isset($_GET["alg_id"])) {
$_SESSION["alg_id"]=(int)$_GET["alg_id"];
$_SESSION["alg_name"]=$_GET["alg_name"];
header("Location: $_SERVER[PHP_SELF]");
}
?>
<?require_once("$_SERVER[DOCUMENT_ROOT]/../db/dal.inc.php");?>
<?
if(isset($_POST["go"])) {
foreach($_POST["step_order"] As $k=>$step_order) {
if(isset($_POST["is_removed"][$k])) {
if(isset($_POST["step_id"][$k]))
DBDelete_AlgorithmStep((int)$_POST["step_id"][$k]);
}
else {
if(!isset($_POST["step_id"][$k]))
DBAdd_AlgorithmStep((int)$step_order,
(int)$_POST["func_id"][$k],0,(int)$_SESSION["alg_id"]);
else
DBUpdate_AlgorithmStep(
(int)$_POST["step_id"][$k],
(int)$step_order,(int)$_POST["func_id"][$k],0,
(int)$_SESSION["alg_id"]);
foreach($_POST["arg_val"][$step_order] As $fa_id=>$fa_val) {
if(!isset($_POST["arg_val_id"][$step_order][$fa_id]))DBAddArgs(
$fa_id,(int)$_POST["step_id"][$k],$_POST["arg_type_mnemo"][$fa_id],
```

```
$_POST["arg_val"][$step_order][$fa_id]);
else
DBUpdateArgs($_POST["arg_val_id"][$step_order][$fa_id],
(int)$_POST["step_id"][$k],$_POST["arg_type_mnemo"][$fa_id],$_POST["a
rg_val"][$step_order][$fa_id]);
}
}
}

header("Location: $_SERVER[PHP_SELF]");}
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>HTML-таблица</title>
<script src="/libs/jqueryui/js/jquery-1.12.4.js"></script>
<script type="text/javascript" src="/libs/jqueryui/js/jquery-
ui.js"></script>
<link rel="stylesheet" href="/libs/jqueryui/css/jquery-ui.css">
<link rel="stylesheet" href="/libs/jqueryui/css/style.css">
<style>
table {
width: 100%; /* Ширина таблицы */
background: black; /* Цвет фона таблицы */
color: black; /* Цвет текста */
border-spacing: 1px; /* Расстояние между ячейками */
}
td, th {
background: white; /* Цвет фона ячеек */
padding: 5px; /* Поля вокруг текста */
```

```

}
</style>
<script>
function onstepremove() {
var item = $(this).parent().parent().parent();
item.find(".step_order").attr("class","rem_order");
item.fadeOut(1300,function(){
$('<input name="is_removed[]" type="hidden" value="1"/>').appendTo($(item).find("span.step_info"));
refresh_order();
});
return false;}
function refresh_order() {
var s="";
$(".remove").unbind("click");
$(".remove").click(onstepremove);
$(".remove").each(function() {$(this).unbind("click");
$(this).click(onstepremove);
$("title").text(s); $("textarea").text("");
$(".step_order").each(function(n) {
$(this).parent().parent().parent().parent().find("h3 b").html(n);
s+=" "+n;
$("title").text(s);
$(this).val(n);
var first = true;
$(this).parent().find("input").each(function(){if(first)
$("textarea").append("[ "+$(this).parent().parent().parent().parent().
.find("h3").html()+"]\n");

```

```

first=false;
str=$(this).attr("name").replace(/\[.*\]/gi,"["+n+"]");
$(this).attr("name",str);});});
$("#rem_order").each(function(n) {
$(this).val(n);
$(this).parent().find("input").each(function(){
str=$(this).attr("name").replace(/\[.*\]/gi,"[rm"+n+"]");
$(this).attr("name",str);
});
});
}
$( function() {
refresh_order();
$("#sortable2" )
.accordion({header:'h3'})
.sortable({
helper: function (e,li){
this.copyHelper = li.clone().insertAfter(li);
$(this).data('copied',false);
return li.clone();},
connectWith: ".connectedSortable",
stop: function(e,li) {var copied = $(this).data('copied');
if (!copied) {this.copyHelper.remove();}
this.copyHelper = null;
$( ".connectedSortable" ).accordion( "refresh" );
}
}).disableSelection();
$("#sortable1")
.accordion({header:'h3'})

```

Продолжение приложения А

```

.sortable({
receive: function (e,ui) {
ui.sender.data('copied', true);
var func_name = $(ui.item).find(".func_name").val();
var func_id = $(ui.item).find(".func_id").val();
var arg_names =
[]$(ui.item).find(".func_argname").each(function(k,v){arg_names.push(
$(v).val());});
$("#func_template table").html("");
$("#func_template h3 span").html(func_name);
$("#func_template input.func_id").val(func_id);
var htm="";
$.each(arg_names,function(k,v){
htm+=''<tr><td>'+v+'</td><td><input type="text" size="2"/></td></tr>';
});
$("#func_template table").html(htm);
$(ui.item).html($("#func_template").html());
if($("#sortable1 li").length>1) {
$("#sortable1 li#hint").hide();
} else {
$("#sortable1 li#hint").show();
}
},
stop: function(e,li) {
refresh_order();
},
update: function() {
refresh_order();}}});
$( "#sortable1" ).on( "sortstop", function( event, ui ) {

```

Продолжение приложения А

```

var line_num=$(ui.item).prevAll().length;
$(ui.item).find(".line_num").val(line_num);
});
"#sortable2" ).on( "sortremove", function( event, ui ) {
var func_name = $(ui.item).find(".func_name").val();
var arg_names = []
$(ui.item).find(".func_argname").each(function(k,v){
arg_names.push($(v).val());
});
$("#func_template table").html("");
$("#func_template h3").html(func_name);
$.each(arg_names,function(k,v){
$("#func_template table").append(
'<tr><td>'+v+'</td><td><input type="text" size="2"/></td></tr>'
);
});
} );

</script>
</head>
<body>
<!--xmp>
<!--?print_r($_SESSION);?-->
<!--?print_r($_POST);?>
</xmp-->
<a
href="index.php">Алгоритмы</a>&nbsp;&nbsp;&nbsp;<span><?=$_SESSION["alg
_name"]?></span>
<div id="func_template" style="display:none" >

```

Продолжение приложения А

```

<div class="alg_step">
<h3><span>Имя функции</span><input style="float: right;"
class="remove" type="button" value="x"/><b></b></h3>
<div>
<p>
<table border="0" cellspacing="0" cellpadding="0">
</table>
<span class="step_info" style="display: none;">
<input class="func_id" name="func_id[]" type="hidden" value=""/>
<input class="step_order" name="step_order[]" type="hidden" val-
ue=""/>
</span>
</p>
</div>
</div>
</div>
</div>
<table>
<tr><th>область алгоритма </th><th>панель инструментов</th></tr>
<tr><td>
<!--li class="ui-state-default">
<h3>Имя функции <?=$i?></h3>
<div>
<p>
<table border="0" cellspacing="0" cellpadding="0">
<tr>
<td>Аргумент 1</td>
<td><input type="text" size="2"/></td>
</tr>
<tr>
<td>Аргумент 2</td>

```

Продолжение приложения А


```

<td><input type="text" size="2"/></td>
</tr>
<tr>
<td>Аргумент N</td>
<td><input type="text" size="2"/></td>
</tr>
</table>
</p>
</div>
</li>
<?}?><?*/?>
<form action="" method="POST">
<input name="go" type="submit" value="УСЛОВИЕ"/>

<ul id="sortable1" class="connectedSortable">
<?while($item=DBFetch_AlgStep($_SESSION["alg_id"])):?>
<?$arg_vals=DBListArgs($item["ID"]);?>
<li class="ui-state-default">
<div class="alg_step">
<h3><span><?=$item["FuncName"]?></span><input style="float: right;"
class="remove" type="button" value="x"/><b></b></h3>
<div>
<p>
<table border="0" cellspacing="0" cellpadding="0">
<?while($arg=DBFetchFuncArgs($item["FuncID"])):?>
<tr>
<td>
<?=$arg["Name"]?>
<input name="arg_type_mnemo[<?=$arg["ID"]?>]" type="hidden" val-
ue="<?=$arg["Mnemo"]?>"/>

```

Продолжение приложения А

```

</td>
<td>
<input name="arg_val[<?=$item["StepOrder"]?>][<?=$arg["ID"]?>]"
type="text" size="2" value="<?=$arg_vals[$arg["ID"]]["Val"]?>"/>
<?if($arg_vals[$arg["ID"]]["ID"]):?>
<input name="arg_val_id[<?=$item["StepOrder"]?>][<?=$arg["ID"]?>]"
type="hidden" value="<?=$arg_vals[$arg["ID"]]["ID"]?>"/>
<?endif;?>
</td>
</tr>
<?endwhile;?>
</table>
<span class="step_info" style="display: none;">
<input class="step_id" name="step_id[]" type="hidden" val-
ue="<?=$item["ID"]?>"/>
<input class="func_id" name="func_id[]" type="hidden" val-
ue="<?=$item["FuncID"]?>"/>
<input class="step_order" name="step_order[]" type="hidden" val-
ue="<?=$item["StepOrder"]?>"/>
</span>
</p>
</div>
</div>
</li>
<?endwhile?>
<li id="hint" clas="ui-state-default">
Перетаскивайте функции с панели инструментов
</li>
</ul>
</form>

```

Продолжение приложения А

```

</td>
<td width="30%">
<ul id="sortable2" class="connectedSortable">
<?while($func=DBFetchFunction()) {?>
<li class="ui-state-default">
<input class="func_id" type="hidden" value="<?=$func["ID"]?>"/>
<input class="func_name" type="hidden" value="<?=$func["Name"]?>"/>
<input class="func_description" type="hidden" val-
ue="<?=$func["Description"]?>"/>
<h3><?=$func["Name"]?><?=$i?><b></b></h3>
<div>
<p>
<?=$func["Description"]?><br/>
<!--Аргументы (формальные):-->
<ul>
<?while($arg=DBFetchFuncArgs($func["ID"])) {?>
<li>
<b><?=$arg["Name"]?></b> - <?=$arg["Description"]?>
<input class="func_argname" type="hidden" value="<?=$arg["Name"]?>"/>
<input class="func_argdescription" type="hidden" val-
ue="<?=$arg["Description"]?>"/>
</li><?}&?>
</ul></p></div></li><?}&?>
</ul></td>
</tr>
</table>
</body>
</html></html>

```

ПРИЛОЖЕНИЕБ

Код страницы функции

```
<?php
//Подключение слоя доступа к данным
require_once("$_SERVER[DOCUMENT_ROOT]/../db/dal.inc.php");
//ВАЛИДАЦИЯВВОДАНА PHP
$errmsg=""; $isvalid=true; $selectors="";
function set_error($message, $input_selector="") {
global $errmsg, $isvalid, $selectors;
static $comma;
$errmsg.=$message."<br/>";
if(trim($input_selector)!="")
$selectors.="$comma $input_selector";
$comma=",";
$isvalid=false;
}
if(isset($_GET["editid"])) {
$item=DBGetFunction((int)$_GET["editid"]);
$form_fields["f_ID"]=$item["ID"];
$form_fields["f_title"]=$item["Name"];
$form_fields["f_content"]=$item["Description"];
}
if(isset($_GET["delid"])) {
DBDeleteFunction((int)$_GET["delid"]);
header("Location: $_SERVER[PHP_SELF]");
}
if(isset($_POST["Go"])) {
//Фильтрация
```

```
$form_fields["f_title"]=_DBEscString($_POST["f_title"]);
$form_fields["f_content"]=_DBEscString($_POST["f_content"]);
if(isset($_POST["f_ID"]))
$form_fields["f_ID"]=(int)$_POST["f_ID"];
//Валидация
if(trim($form_fields["f_title"])=="")
set_error("Поле Заголовков не заполнено","#f_title");
if(trim($form_fields["f_content"])=="")
set_error("Поле Описание не заполнено","#f_content");
if($isvalid) {
try {
//Сохраним вакансию в базе данных
if(!isset($_POST["f_ID"])) {
DBAddFunction(
$form_fields["f_title"],
$form_fields["f_content"]
);
}
else
DBUpdateFunction(
$form_fields["f_ID"],
$form_fields["f_title"],
$form_fields["f_content"]
);
//и перейдём на эту же страницу
header("Location: $_SERVER[PHP_SELF]");
}
catch(Exception $ex) {
//Если при сохранении возникла ошибка
```

```

//выведемееописание
set_error($ex->getMessage());
}
}
}
?>
<!DOCTYPE html>
<html>
<head>
<?if(!$isvalid):?>
<!--В случае ошибки выделим неверно заполненные поля красным цветом -->
<style>
<?=$selectors?> {
border-color: #F00;
border-style: solid;
border-width: 1px;
}
</style>
<?endif;?>
</head>
<body>

<form action="" method="POST">
Название:<br/>
<input id="f_title" name="f_title" type="text" size="50" value="<?=$form_fields["f_title"]?>"/><br/>
Описание:<br/>
<textarea id="f_content"
name="f_content"><?=$form_fields["f_content"]?></textarea><br/>

```

```

<?if(isset($form_fields["f_ID"])):?>
<input name="f_ID" type="hidden" value="<?=$form_fields["f_ID"]?>" />
<?endif;?>
</div><br/>
<div style="color: #F00;"><?=$errmsg?></div>
<input name="Go" type="Submit" value="Сохранить" />
</form>

<br/>
<table width="100%" border="1">
<tr>
<th>Название</th>
<th width="20%"></th>
</tr>
<?_DBFetchQuery(NULL, Array("reset"=>1));?>

<?while($item=DBFetchFunction($user_id)):?>
<tr>
<td><a
href="cv.php?func_id=<?=$item["ID"]?>&func_name=<?=$item["Name"]?>"><
?=$item["Name"]?></a></td>
<td>
<a href="?editid=<?=$item["ID"]?>">Редактировать</a>
<a href="?delid=<?=$item["ID"]?>" onclick="return
confirm(' Действительноудалить? ');">Удалить</a>
</td>
</tr>
<?endwhile;?></table>
</body></html>

```

ПРИЛОЖЕНИЕ В

Код страницы аргументов

```
<?php
//Подключение слоя доступа к данным
require_once("$_SERVER[DOCUMENT_ROOT]/../db/dal.inc.php");
session_start();
if(isset($_GET["func_id"])) {
$_SESSION["func_id"]=(int)$_GET["func_id"];
$_SESSION["func_name"]=(int)$_GET["func_name"];
header("Location: $_SERVER[PHP_SELF]");
}
//ВАЛИДАЦИЯВВОДАНА PHP
$errmsg=""; $isvalid=true; $selectors="";
function set_error($message, $input_selector="") {
global $errmsg, $isvalid, $selectors;
static $comma;
$errmsg.=$message."<br/>";
if(trim($input_selector)!="")
$selectors.="$comma $input_selector";
$comma=",";
$isvalid=false;
}
if(isset($_GET["editid"])) {
$item=DBGetFunction((int)$_GET["editid"]);
$form_fields["f_title"]=$item["Name"];
$form_fields["f_content"]=$item["Description"];
}
if(isset($_GET["delid"])) {
```



```
DBDeleteFunction((int)$_GET["delid"]);
header("Location: $_SERVER[PHP_SELF]");
}
if(isset($_POST["Go"])) {
//Фильтрация
$form_fields["f_title"]=_DBEscString($_POST["f_title"]);
$form_fields["f_content"]=_DBEscString($_POST["f_content"]);
if(isset($_POST["f_ID"]))
$form_fields["f_ID"]=(int)$_POST["f_ID"];
//Валидация
if(trim($form_fields["f_title"])=="")
set_error("Поле Заголовков не заполнено","#f_title");
if(trim($form_fields["f_content"])=="")
set_error("Поле Описание не заполнено","#f_content");
if($isvalid) {
try {
//Сохраним вакансию в базе данных
if(!isset($_POST["f_ID"])) {
DBAddFunction(
$form_fields["f_title"],
$form_fields["f_content"]
);
}
else
DBUpdateFunction(
$form_fields["f_ID"],
$form_fields["f_title"],
$form_fields["f_content"]
);
}
```

```
//и перейдём на эту же страницу
//для сброса полей формы и
//предотвращения дублирования данных
header("Location: $_SERVER[PHP_SELF]");
}
catch(Exception $ex) {
//Если при сохранении возникла ошибка
//выведем её описание
set_error($ex->getMessage());
}
}
}
?>
<!DOCTYPE html>
<html>
<head>
<?if(!$isvalid):?>
<!--В случае ошибки выделим неверно заполненные поля красным цветом -
->
<style>
<?=$selectors?> {
border-color: #F00;
border-style: solid;
border-width: 1px;
}
</style>
<?endif;?>
</head>
<body>
<h1>Аргументы функции<?=$_SESSION["func_name"]?></h1>
```

```

<form action="" method="POST">
Название:<br/>
<input id="f_title" name="f_title" type="text" size="50" value="<?=$form_fields["f_title"]?>"/><br/>
Описание:<br/>
<textarea id="f_content"
name="f_content"><?=$form_fields["f_content"]?></textarea><br/>
<?if(isset($form_fields["f_ID"])):?>
<input name="f_ID" type="hidden" value="<?=$form_fields["f_ID"]?>"/>
<?endif;?>
</div><br/>
<div style="color: #F00;"><?=$errmsg?></div>
<input name="Go" type="Submit" value="Сохранить"/>
</form>
<br/>
<table width="100%" border="1">
<tr>
<th>Название</th>
<th width="20%"></th>
</tr>
<?_DBFetchQuery(NULL,Array("reset"=>1));?>
<?while($item=DBFetchFunction($user_id)):?>
<tr>
<td><?=$item["Name"]?></td>
<td>
<a href="?editid=<?=$item["ID"]?>">Редактировать</a>
<a href="?delid=<?=$item["ID"]?>" onclick="return
confirm('Действительноудалить?');">Удалить</a>
</td>

```

```
</tr>  
<?endwhile;?>  
</table>  
</body>  
</html>
```

ПРИЛОЖЕНИЕГ

Код прописанных функций

```
<?php re-
quire_once("$_SERVER[DOCUMENT_ROOT]/../db/common.dal.inc.php");

function DBFetchFunction() {
return _DBFetchQuery("
SELECT * FROM functions
",Array("fetcher_id"=>"functions"));
}

function DBAddFunction($name,$description) {
_DBQuery("
INSERT INTO functions(Name,Description)
VALUES('$name','$description')");
}

function DBUpdateFunction($ID,$name,$description) {
_DBQuery("
UPDATE functions SET Name='$name',Description='$description' WHERE
ID=$ID");
}

function DBGetFunction($FuncID) {
return _DBGetQuery("SELECT * FROM functions WHERE ID=$FuncID");
}

function DBDeleteFunction($ID) {
_DBQuery("DELETE FROM functions WHERE ID=$ID");
}
```

```

function DBAddFuncArgs($funcid,$argtypeid,$name,$argdescription) {
  _DBQuery("INSERT INTO funcargs(FuncID,ArgTypeID,Name,Argdescription)
VALUES('$funcid','$argtypeid','$name','$argdescription')");
}
function
DBUpdateFuncArgs($id,$funcid,$argtypeid,$name,$argdescription) {
  _DBQuery("
UPDATE funcargs
SET FuncID='$funcid',
ArgTypeID='$argtypeid',
Name='$name',
ArgDescription='$argdescription'
WHERE ID=$id
");
}
function DBDeleteFuncArg($ID) {
  _DBQuery("DELETE FROM funcargs WHERE ID=$ID");
}
function DBGetFuncArg($ID) {
  return _DBGetQuery("SELECT * FROM funcargs WHERE ID=$ID");
}
function DBFetchFuncArgs($funcid) {
  return _DBFetchQuery("
SELECT
funcargs.IDAsID,
funcargs.Name As Name,
argtypes.Name As TypeName,
funcargs.ArgDescription As Description,
argtypes.Mnemo As Mnemo
FROM funcargs,argtypes

```

```

WHERE
funcargs.ArgTypeID=argtypes.ID AND
funcargs.FuncID=$funcid
",Array("fetcher_id"=>"args"));
}
//-----
function DBAddAlgorithms($name,$typeid,$description) {
  _DBQuery("INSERT INTO Algorithms(Name,TypeID,Description)
VALUES('$name','$typeid','$description')");
}
function DBUpdateAlgorithms($id,$name,$typeid,$description) {
  _DBQuery("
UPDATE Algorithms
SET NAME='$name',
TypeID='$typeid',
Description='$description'
WHERE ID=$id
");
}
function DBGetAlgorithms($ID) {
return _DBGetQuery("SELECT * FROM Algorithms WHERE ID=$ID");
}
function DBDeleteAlgorithms($ID) {
  _DBQuery("DELETE FROM Algorithms WHERE ID=$ID");
}
function DBFetchAlgorithms() {
return _DBFetchQuery("
SELECT
Algorithms.ID As ID,

```

```

Algorithms.Name As Name,
algtypes.Name As TypeName,
Algorithms.Description As Description
FROM Algorithms,algtypes
WHERE
Algorithms.TypeID=algtypes.ID
",Array("fetcher_id"=>"algs"));
}

```

//-----

```

function DBFetchArgTypes() {
return _DBFetchQuery("
SELECT
*
FROM argtypes
");
}

```

//-----

```

function DBFetchAlgTypes() {
return _DBFetchQuery("
SELECT
*
FROM algtypes
");
}

```

//-----

Реализовать DAL для таблицы AlgSteps

DBFetch_AlgStep

DBAdd_AlgSteps


```
DBUpdate_AlgSteps
DBDelete_AlgStep
function DBFetch_AlgStep($alg_id) {
return _DBFetchQuery("
SELECT
AlgSteps.ID As ID,
AlgSteps.StepOrder As StepOrder,
functions.ID As FuncID,
functions.Name As FuncName,
functions.Description As Description
FROM AlgSteps,functions
WHERE
AlgSteps.FuncID=functions.ID AND
AlgSteps.AlgID = $alg_id
ORDER BY
StepOrder ASC
",Array("fetcher_id"=>"step"));
}
function DBAdd_AlgStep($steporder,$funcid,$blockid,$algid) {
_DBQuery("INSERT INTO AlgSteps(StepOrder,FuncID,BlockID, AlgID)
VALUES('$steporder','$funcid','$blockid','$algid)");
}
function DBUpdate_AlgStep($id,$steporder,$funcid,$blockid,$algid) {
_DBQuery("
UPDATE AlgSteps
SET StepOrder='$steporder',
FuncID='$funcid',
BlockID='$blockid',
AlgID='$algid'
```

```

WHERE ID=$id
");
}
function DBGet_AlgsSteps($ID) {
return _DBGetQuery("SELECT * FROM AlgsSteps WHERE ID=$ID");
}
function DBDelete_AlgsStep($ID) {
_DBQuery("DELETE FROM AlgsSteps WHERE ID=$ID");
}
//-----
$val_f=Array(
"INT"=>"IntVal",
"DBL"=>"DoubleVal",
"STR"=>"TextVal",
);
function DBAddArgs($argid,$stepid,$mnemo,$val) {
global $val_f;
//echo "mnemo=$mnemo";
_DBQuery("INSERT INTO args(FuncArgID,StepID,$val_f[$mnemo])
VALUES('$argid','$stepid','$val')");
}
function DBUpdateArgs($id,$stepid,$mnemo,$val) {
global $val_f;
_DBQuery("
UPDATE args
SET
StepID='$stepid',
$val_f[$mnemo]='$val'
WHERE ID=$id

```

```

");}
function DBDeleteArg($StepID) {
  _DBQuery("DELETE FROM args WHERE StepID=$StepID");}
function DBGetArg($ID) {
  return _DBGetQuery("SELECT * FROM args WHERE ID=$ID");}
function DBListArgs($stepid) {
  //$arg_vals=Array();
  while($item=_DBFetchQuery("
  SELECT
  args.ID As ID,
  args.FuncArgID,
  IF(args.IntVal IS NOT NULL, args.IntVal,
  IF(args.DoubleVal IS NOT NULL, args.DoubleVal,
  IF(args.TextVal IS NOT NULL, args.TextVal, \"\"))
  ) AsVal,
  argtypes.Mnemo As Mnemo
  FROM args,funcargs,argtypes
  WHERE
  args.FuncArgID = funcargs.ID AND
  funcargs.ArgTypeID=argtypes.ID AND
  args.StepID='$stepid'

  ",Array("fetcher_id"=>"args"))) {
  $arg_vals[$item["FuncArgID"]]=$item;
  }
  return $arg_vals;
}

```

ПРИЛОЖЕНИЕ Д

Код страницы созданных алгоритмов

```
<?require_once("$_SERVER[DOCUMENT_ROOT]/../db/dal.inc.php");?>
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>HTML-таблица</title>
<script src="/libs/jqueryui/js/jquery-1.12.4.js"></script>
<script type="text/javascript" src="/libs/jqueryui/js/jquery-
ui.js"></script>
<link rel="stylesheet" href="/libs/jqueryui/css/jquery-ui.css">
<link rel="stylesheet" href="/libs/jqueryui/css/style.css">
<style>
table {
width: 100%; /* Ширинатаблицы */
background: black; /* Цветфонатаблицы */
color: black; /* Цветтекста */
border-spacing: 1px; /* Расстояниемеждуячейками */
}
td, th {
background: white; /* Цветфонаячеек */
padding: 5px; /* Поля вокруг текста */
}
</style>
<script>
/*$( function() {
$( "#accord1,#accord2" )
.accordion({
header: ">div>h3"})
```

```
.sortable({
axis: "y",
handle: "h3",
connectWith: ".connectedSortable",
stop: function( event, ui ) {
ui.item.children( "h3" ).triggerHandler( "focusout" );

$(".connectedSortable").accordion( "refresh" );
}
});
} );*/
$( function() {
$( "#sortable1, #sortable2" )
.accordion({header:'h3'})
.sortable({
helper: "clone",
connectWith: ".connectedSortable",
stop: function( event, ui ) {
// IE doesn't register the blur when sorting
// so trigger focusout handlers to remove .ui-state-focus
ui.item.children( "h3" ).triggerHandler( "focusout" );
// Refresh accordion to handle new order
$( ".connectedSortable" ).accordion( "refresh" );
//alert($( this ).attr("id"));
change: function(event,ui)
$( ".connectedSortable" ).accordion( "refresh" );
$("#hint").css("display","none");
}).disableSelection();
```

```

//$( "#sortable2" ).sortable( "option", "helper", "clone" );
$( "#sortable2" ).on( "sortremove", function( event, ui ) {
//$(ui.item).html($("#func_template").html());
alert("before");
var func_name = $(ui.item).find(".func_name").val();
var arg_names = []
$(ui.item).find(".func_argname").each(function(k,v){
arg_names.push($(v).val());
});
$("#func_template table").html("");
$("#func_template h3").html(func_name);
$.each(arg_names,function(k,v){
$("#func_template table").append(
'<tr><td>'+v+'</td><td><input type="text" size="2"/></td></tr>'
);
});
alert("after");
//alert(func_name);
ui.item.clone().html($("#func_template").html()).appendTo('#sortable1
');
$(this).sortable('cancel');
} );
} );
</script>
</head>
<body>
<div id="func_template" style="display:none">
<h3>Имя функции</h3>
<div>

```

```

<p>
<table border="0" cellspacing="0" cellpadding="0">
</table>
<input id="line_number" name="line_number" type="hidden" value=""/>
</p>
</div>
</div>
<table>
<tr><th>область алгоритма</th><th>панель инструментов</th></tr>
<tr><td>
<ul id="sortable1" class="connectedSortable">
<li id="hint" class="ui-state-
default">Перетаскивайте функции панели инструментов</li>
<!--li class="ui-state-default">
<h3>Имя функции <?=$i?></h3>
<div>
<p></p>
</div>
</li-->
</ul>
<?/*for($i=0;$i<5;$i++) {?>
<li class="ui-state-default">
<h3>Имя функции <?=$i?></h3>
<div>
<p>
<table border="0" cellspacing="0" cellpadding="0">
<tr>
<td>Аргумент 1</td>

```

```

<td><input type="text" size="2"/></td>
</tr>
<tr>
<td>Аргумент 2</td>
<td><input type="text" size="2"/></td>
</tr>
<tr>
<td>Аргумент N</td>
<td><input type="text" size="2"/></td>
</tr>
</table>
</p>
</div>
</li>
<?}?><?*/?>
</ul>
</td>
<tdwidth="30%">
<ul id="sortable2" class="connectedSortable">
<?while($func=DBFetchFunction()) {?>
<li class="ui-state-default">
<input class="func_name" type="hidden" value="<?=$func["Name"]?>"/>
<input class="func_description" type="hidden" val-
ue="<?=$func["Description"]?>"/>
<h3><?=$func["Name"]?><?=$i?></h3>
<div>
<p>
<?=$func["Description"]?><br/>
Аргументы (формальные):

```



```

<ul>
<?while($arg=DBFetchFuncArgs($func["ID"])) {?>
<li>
<b><?=$arg["Name"]?></b> - <?=$arg["Description"]?>
<input class="func_argname" type="hidden" value="<?=$arg["Name"]?>"/>
<input class="func_argdescription" type="hidden" value="<?=$arg["Description"]?>"/>
</li>
<?}?>
</ul>
</p>
</div>
</li>
<?}?>
</ul>
</td>
</tr>
</table>
</body>
</html>

```