

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ
(ФИЛИАЛ) ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО АВТОНОМНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Кафедра «Информационные системы»

Утверждаю
Заведующий кафедрой ИС

_____ Р.А.Валиев

_____ г.

КУРСОВОЙ ПРОЕКТ

по дисциплине:

«Корпоративные информационные системы»

на тему:

«Разработка автоматизированной информационной системы на
технологической платформе 1С:Предприятие 8.3»

Вариант:

**Разработка АИС для тестирования учащихся в
общеобразовательной школе**

Автор:
студент группы 2211122

_____ А. В. Ибашаев

Оценка: _____

Руководитель:
Ст. преподаватель кафедры ИС

_____ Л. Б. Хузятова

Дата защиты: _____ г.

Набережные Челны
2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Кафедра «Информационные системы»

Утверждаю
Заведующий кафедрой ИС

_____ Р.А.Валиев

_____ г.

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Студент:

Ибашаев А. В.

1 Тема

«Разработка АИС для тестирования учащихся в общеобразовательной школе»

2 Срок представления к защите

_____ г.

3 Исходные данные

Информационная система

4 Перечень подлежащих разработке вопросов

- Описание бизнес-требований;
- Разработка модели данных;
- Моделирование бизнес-процессов;
- Разработка пользовательских требований;
- Разработка функциональных требований;
- Разработка нефункциональных требований;
- Проектирование АИС;
- Разработка функциональной модели системы;
- Разработка алгоритмов системы.
- Программная реализация проекта

Задание выдано _____ г. _____ Л. Б. Хузятова

Задание принято _____ г. _____ А. В. Ибашаев

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	3
ВВЕДЕНИЕ	4
1 Анализ предметной области	6
1.1 Обзор предметной области	6
1.2 Описание структуры школы	7
1.3 Описание уровня автоматизации школы	8
1.4 Выявление узких мест текущего состояния объекта анализа	10
1.5 Обзор существующих методов решения проблемы	10
1.6 Постановка задачи на проектирование.....	11
2 Реализация информационной системы.....	13
2.1 Описание бизнес-процесса «Тестирования учеников».....	13
2.2 Описание разработанных модулей	14
2.3 Описание структуры разработанных объектов	17
3 Инструкция к приложению.....	20
ЗАКЛЮЧЕНИЕ	25
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	27
Приложение А	28
Приложение Б.....	32

ВВЕДЕНИЕ

Современные информационные технологии играют важную роль в повышении эффективности образовательных процессов, упрощении управления учебной деятельностью и улучшении контроля знаний учащихся. Одним из направлений их применения является автоматизация тестирования в школах, что позволяет сделать процесс проверки знаний более объективным, удобным и доступным. В традиционной системе тестирования значительное время затрачивается на составление, проведение и проверку тестов, что снижает общую эффективность учебного процесса. Внедрение автоматизированных систем позволяет минимизировать эти затраты, повысить точность оценки знаний и обеспечить гибкость в настройке тестов под различные образовательные потребности.

Актуальность выбранной темы обусловлена необходимостью создания удобного, централизованного инструмента для организации и проведения тестирования в школах. На сегодняшний день многие образовательные учреждения продолжают использовать бумажные или слабо автоматизированные формы контроля знаний, что затрудняет анализ результатов и оперативную обратную связь с учащимися. Кроме того, традиционные методы тестирования требуют значительных ресурсов со стороны преподавателей, что может приводить к увеличению нагрузки и снижению качества проверки знаний. Разработка автоматизированной системы тестирования позволит решить эти проблемы, обеспечивая эффективность, доступность и удобство использования как для учеников, так и для преподавателей.

Целью данного курсового проекта является создание автоматизированной информационной системы (АИС) для тестирования учеников в школе, которая позволит упростить процесс проведения

тестов, автоматизировать проверку ответов и анализировать результаты обучения.

Для достижения поставленной цели требуется решить следующие задачи:

- Анализ существующих методов и инструментов для тестирования, выявление их преимуществ и недостатков;
- Разработка технического задания на создание АИС для автоматизированного тестирования;
- Проектирование архитектуры системы;
- Разработка структуры базы данных для хранения информации о тестах, пользователях, вопросах и ответах;
- Разработка алгоритмов для создания, управления и оценки тестов;
- Реализация пользовательского интерфейса системы;
- Тестирование и отладка системы для обеспечения ее надежности и функциональности.

Разработанная система позволит значительно повысить качество образовательного процесса, предоставляя удобный и эффективный инструмент для контроля знаний учащихся.

1 Анализ предметной области

1.1 Обзор предметной области

Тестирование является одним из наиболее эффективных методов оценки знаний учащихся, позволяя объективно определить уровень усвоения материала. Оно широко применяется в образовательных учреждениях для контроля успеваемости, подготовки к экзаменам и самостоятельного изучения тем. В современных условиях автоматизация тестирования становится важной задачей, поскольку традиционные методы проверки знаний требуют значительных временных затрат на подготовку, проведение и анализ результатов.

Автоматизированные системы тестирования реализуются в виде программных решений, которые предоставляют ученикам возможность проходить тесты в удобном формате, а преподавателям — быстро обрабатывать результаты и анализировать успеваемость. Такие системы могут быть интегрированы в образовательные платформы, использоваться в школьных компьютерных классах или быть доступными на мобильных устройствах.

Автоматизированное тестирование применяется в разных образовательных процессах: для текущего контроля знаний, итоговой аттестации, подготовки к экзаменам, а также адаптивного обучения, где сложность вопросов меняется в зависимости от ответов учащегося. Внедрение таких систем позволяет повысить объективность оценки, снизить нагрузку на преподавателей и предоставить учащимся возможность получать мгновенную обратную связь.

Разработка автоматизированной системы тестирования требует учета ряда ключевых факторов, таких как управление вопросами и ответами, регистрация учеников и отслеживание их результатов,

настройка различных режимов тестирования, а также формирование аналитических отчетов. Внедрение такого решения в образовательный процесс способствует повышению его эффективности, улучшает организацию контроля знаний и делает процесс тестирования более удобным и доступным для всех участников.

1.2 Описание структуры школы

Структура школы включает (рисунок 1) в себя четыре основных элемента: Администрация школы, Учителя, Ученики и Родители. Эти элементы взаимодействуют между собой, обеспечивая эффективное функционирование образовательного процесса.

а) Администрация школы:

- Управляет учебным процессом, кадрами и ресурсами.
- Организует контроль за качеством образования.
- Взаимодействует с учителями, учениками и родителями.
- Отвечает за разработку учебных планов и стратегий развития школы.

б) Учителя:

- Осуществляют обучение и воспитание учеников.
- Готовят и проводят уроки, тестирования, контрольные работы.
- Оценивают знания учеников и предоставляют отчёты администрации и родителям.
- Взаимодействуют с родителями для улучшения успеваемости детей.

в) Ученики:

- Получают знания и участвуют в учебном процессе.
- Выполняют домашние задания, тестирования и экзамены.

– Взаимодействуют с учителями и сверстниками в рамках учебной и внеклассной деятельности.

– Могут обращаться к администрации и родителям за поддержкой.

г) Родители:

– Контролируют успеваемость и посещаемость учеников.

– Взаимодействуют с учителями и администрацией школы.

– Принимают участие в собраниях и школьных мероприятиях.

– Оказывают поддержку детям в образовательном процессе.

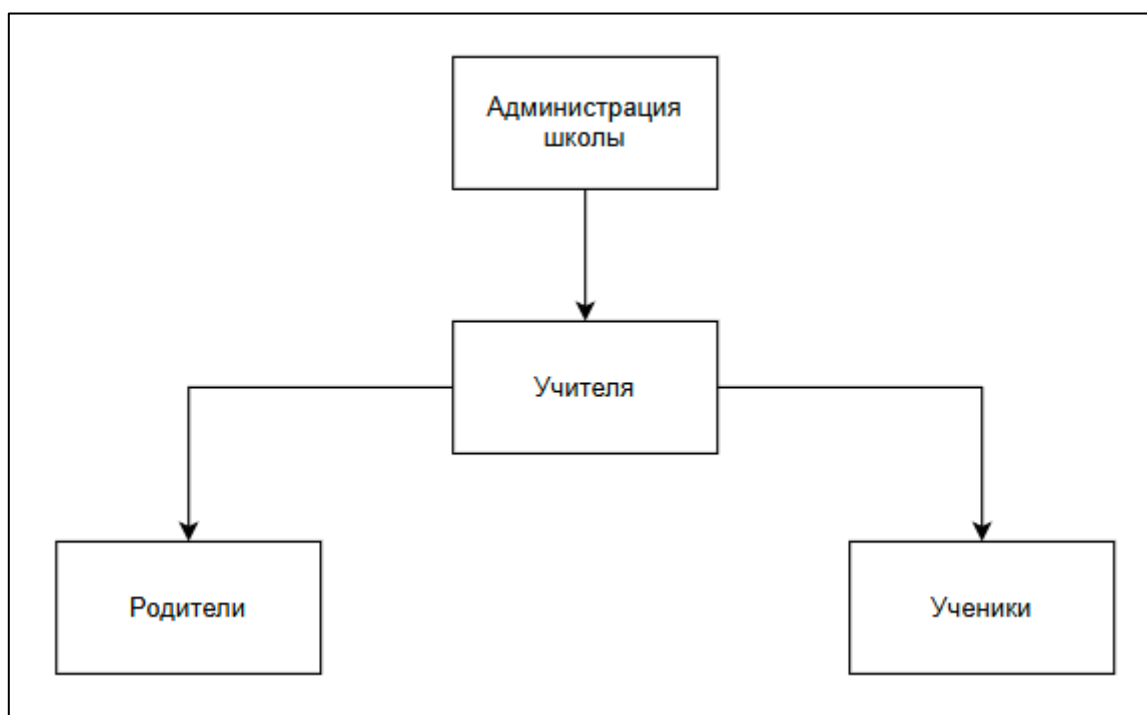


Рисунок 1 – Структурная схема школы

1.3 Описание уровня автоматизации школы

Уровень автоматизации школы определяется степенью внедрения информационных технологий в образовательный и административный процессы, что позволяет повысить их эффективность, упростить управление и улучшить взаимодействие между всеми участниками учебного процесса.

На административном уровне автоматизация охватывает управление расписанием, учет учеников и преподавателей, ведение документации и контроль посещаемости. Внедрение цифровых журналов и дневников позволяет учителям быстрее выставять оценки, а родителям и ученикам — оперативно получать информацию о текущей успеваемости и домашних заданиях. Электронные системы документооборота помогают автоматизировать работу с приказами, заявлениями и отчетами, снижая нагрузку на администрацию.

Для преподавателей автоматизированные системы предоставляют удобные инструменты для ведения уроков, учета посещаемости, формирования учебных планов и отчетов об успеваемости. Интерактивные доски, образовательные платформы и цифровые ресурсы делают занятия более наглядными и вовлекающими для учеников.

Ученики получают доступ к электронным учебным материалам, могут взаимодействовать с преподавателями через образовательные порталы и участвовать в дистанционном обучении. Это особенно важно в условиях необходимости гибкой организации учебного процесса, включая дистанционное и смешанное обучение.

Автоматизация также затрагивает родительский контроль — через специальные платформы родители могут отслеживать посещаемость, успеваемость своих детей и получать важные уведомления от школы.

Таким образом, высокий уровень автоматизации школы позволяет сократить время на рутинные задачи, повысить прозрачность учебного процесса, улучшить организацию работы учителей и администрации, а также сделать обучение более доступным и удобным для учеников и их родителей.

1.4 Выявление узких мест текущего состояния объекта анализа

Анализ текущего процесса тестирования выявил ряд ключевых проблем:

- Высокая трудоемкость подготовки тестов – вопросы создаются вручную, отсутствует централизованное хранилище для их повторного использования, что увеличивает затраты времени преподавателей;
- Сложность проверки ответов – традиционные методы требуют значительных временных затрат на проверку работ и выставление оценок;
- Отсутствие аналитики – у преподавателей нет инструментов для анализа успеваемости учащихся, оценки качества вопросов и отслеживания динамики результатов;
- Отсутствие персонализации – тесты не адаптируются под уровень знаний конкретного ученика, что снижает их эффективность и мотивацию к обучению;
- Ограниченные возможности по организации группового тестирования – большинство существующих решений ориентированы на индивидуальное тестирование, а функции для проведения групповых или командных проверок знаний реализованы слабо.

1.5 Обзор существующих методов решения проблемы

В настоящее время существует несколько подходов к автоматизации тестирования в образовательных учреждениях:

- Использование стандартных платформ (Kahoot, Quizizz, Google Forms) – такие сервисы позволяют быстро создавать и проводить тесты, однако их функциональность ограничена, а возможности настройки и персонализации остаются минимальными;

– Разработка специализированных приложений – создание индивидуальных решений, адаптированных под потребности учебного заведения, позволяет эффективно управлять контентом, автоматизировать проверку знаний и собирать детализированную аналитику;

– Применение технологий искусственного интеллекта – использование алгоритмов машинного обучения дает возможность автоматически адаптировать уровень сложности тестов и анализировать результаты учащихся для персонализации обучения.

Наиболее перспективным решением является разработка специализированной системы, сочетающей гибкость управления вопросами, автоматизированную проверку ответов, расширенные аналитические инструменты и возможность проведения как индивидуального, так и командного тестирования.

1.6 Постановка задачи на проектирование

Целью проектирования является создание программного решения для автоматизированного тестирования учеников в школе, которое будет обеспечивать хранение и обработку данных о пользователях, вопросах, ответах и результатах тестирования. Для реализации данного проекта требуется создать структуру базы данных, способную эффективно управлять контентом тестов, анализировать результаты учеников и обеспечивать гибкость в настройке различных типов тестов.

Структура базы данных будет включать следующие ключевые объекты:

– Пользователи (идентификатор, дата регистрации, класс, роль (ученик/учитель));

- Вопросы (идентификатор, текст вопроса, категория, дата добавления, уровень сложности);
- Ответы (идентификатор, текст ответа, ссылка на вопрос, правильность ответа);
- Результаты тестов (идентификатор, пользователь, дата прохождения, набранные баллы, процент правильных ответов, время на выполнение);
- Категории вопросов (идентификатор, название, описание);
- Настройки тестов (идентификатор, количество вопросов, время на прохождение, тип теста (например, одиночный выбор, множественный выбор), режим тестирования (например, на оценку или для самопроверки)).

Эта структура обеспечит удобное и эффективное хранение и обработку данных, что позволит быстро адаптировать систему под различные сценарии тестирования, а также эффективно анализировать результаты учеников. В процессе разработки системы планируется внедрение механизмов аутентификации (для защиты данных пользователей), ведения статистики, управления контентом (для добавления и редактирования вопросов), а также формирования отчетов по результатам тестирования (для учителей и учеников).

2 Реализация информационной системы

2.1 Описание бизнес-процесса «Тестирования учеников»

Описал бизнес-процесс с помощью BPMN-диаграммы (рисунок 2). Диаграмма визуализирует этапы взаимодействия учеников с системой тестирования, начиная с создания тестов и заканчивая формированием отчета о результатах. Модель охватывает ключевые действия, включая аутентификацию пользователей, выбор теста, прохождение тестирования и обработку результатов. Данная схема позволяет наглядно представить последовательность шагов процесса тестирования и возможные варианты его выполнения.

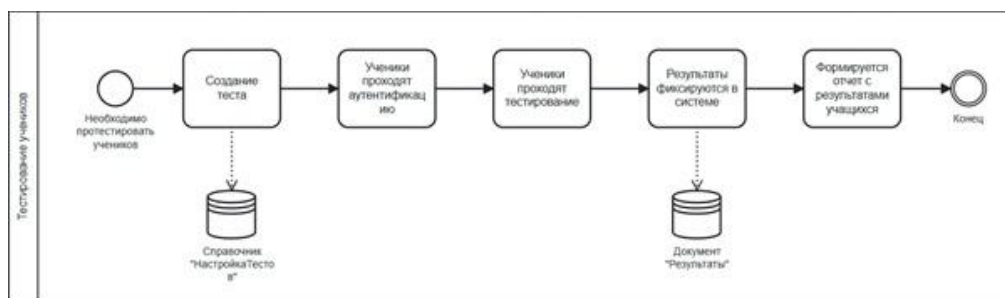


Рисунок 2 – BPMN диаграмма бизнес процесса

Процесс тестирования обучающихся начинается с возникновения необходимости проверить знания учеников. На первом этапе создается тест, который сохраняется в справочнике «Настройка тестов». Этот справочник содержит настройки тестов, включая вопросы, варианты ответов и критерии оценки.

Затем ученики проходят процесс аутентификации, чтобы подтвердить свою личность. Успешная аутентификация предоставляет им доступ к тестированию. После этого учащиеся выбирают тест из доступных вариантов и приступают к его прохождению.

Во время тестирования ученики отвечают на вопросы, и их ответы фиксируются в системе. По завершении тестирования система

автоматически обрабатывает ответы и сохраняет результаты в документе «Результаты». Этот документ содержит данные о правильных и неправильных ответах, а также итоговый балл.

После обработки результатов формируется отчет о прохождении теста. Отчет может включать информацию о результатах каждого ученика, аналитические данные и статистику успеваемости.

Процесс завершается формированием окончательного отчета, который может быть использован для анализа результатов тестирования и принятия решений по корректировке учебного процесса.

2.2 Описание разработанных модулей

В ходе разработки системы тестирования были разработаны два модуля «Тесты» и «Тест».

Процесс работы модуля формы "Тесты" (рисунок 2) начинается с этапа передачи данных о тестах из справочника. На этом этапе система запрашивает информацию о доступных тестах, извлекая их из справочника "НастройкаТестов". Данные включают в себя название тестов, список вопросов и возможные варианты ответов. Далее, полученные данные добавляются на форму, чтобы пользователь мог выбрать интересующую его тест.

После этого пользователь самостоятельно выбирает одну из доступных тестов, после чего система запрашивает у него ввод имени для идентификации. Имя пользователя необходимо для создания индивидуального документа с результатами тестов, который будет использоваться для сохранения ответов и расчета итоговых результатов.

После ввода имени происходит переход на форму тестов, где пользователю последовательно отображаются вопросы с вариантами ответов. При выборе ответа система фиксирует его, сверяет с правильным

вариантом и ведет учет количества верных и неверных ответов. Когда все вопросы пройдены, тест завершается, и результаты сохраняются в соответствующем документе.

Данный процесс подробно реализован в коде модуля, представленного в Приложении А. Код описывает механизмы загрузки тестов, обработки выбора пользователя, взаимодействия с элементами формы и сохранения результатов прохождения тестов.

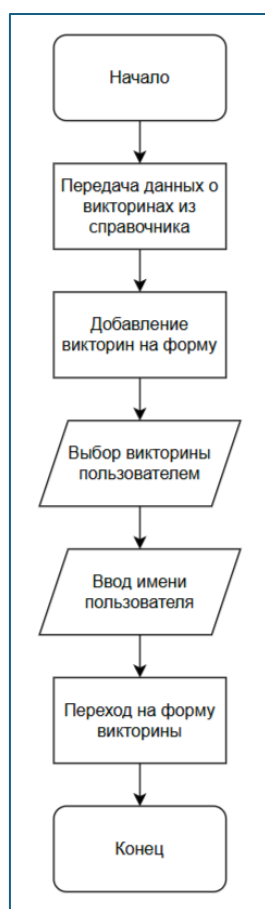


Рисунок 2 – Схема модуля формы «Тесты»

Процесс работы модуля формы "Тест" (рисунок 3) начинается с загрузки вопросов из справочника. На этом этапе система извлекает информацию о вопросах и возможных вариантах ответов из базы данных, где хранятся заранее подготовленные тесты.

После загрузки вопросов система отображает их на форме, предоставляя пользователю интерфейс для взаимодействия. Вопросы

сопровожаются несколькими вариантами ответов, из которых пользователь должен выбрать один.

Далее пользователь осуществляет выбор ответа. Система фиксирует сделанный выбор и передает его для дальнейшей обработки. На этапе обработки ответы пользователя сравниваются с правильными вариантами, производится расчет количества верных и неверных ответов.

По завершении обработки данных результаты прохождения тесты сохраняются в документ, который может быть использован для дальнейшего анализа.

Данный процесс детально реализован в коде модуля, представленного в Приложении Б. Код описывает механизмы загрузки вопросов, отображения формы, обработки пользовательских ответов и сохранения итоговых результатов.

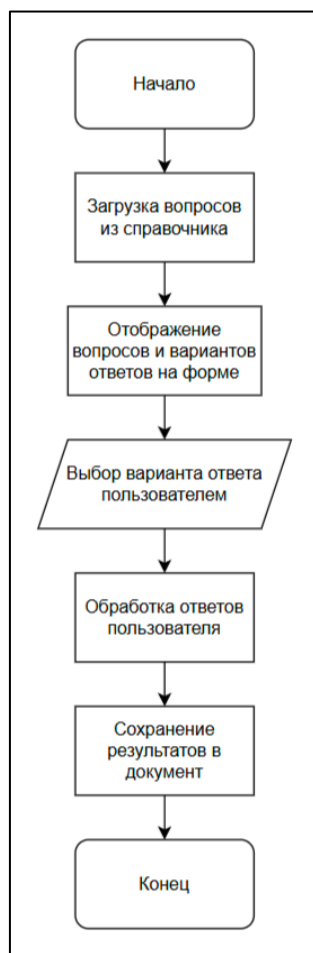


Рисунок 3 – Схема модуля формы «Тест»

2.3 Описание структуры разработанных объектов

Структура разрабатываемых объектов (рисунок 4) включает три основных элемента: справочник "НастройкаТестов", документ "Результаты", регистр накопления "Результаты".

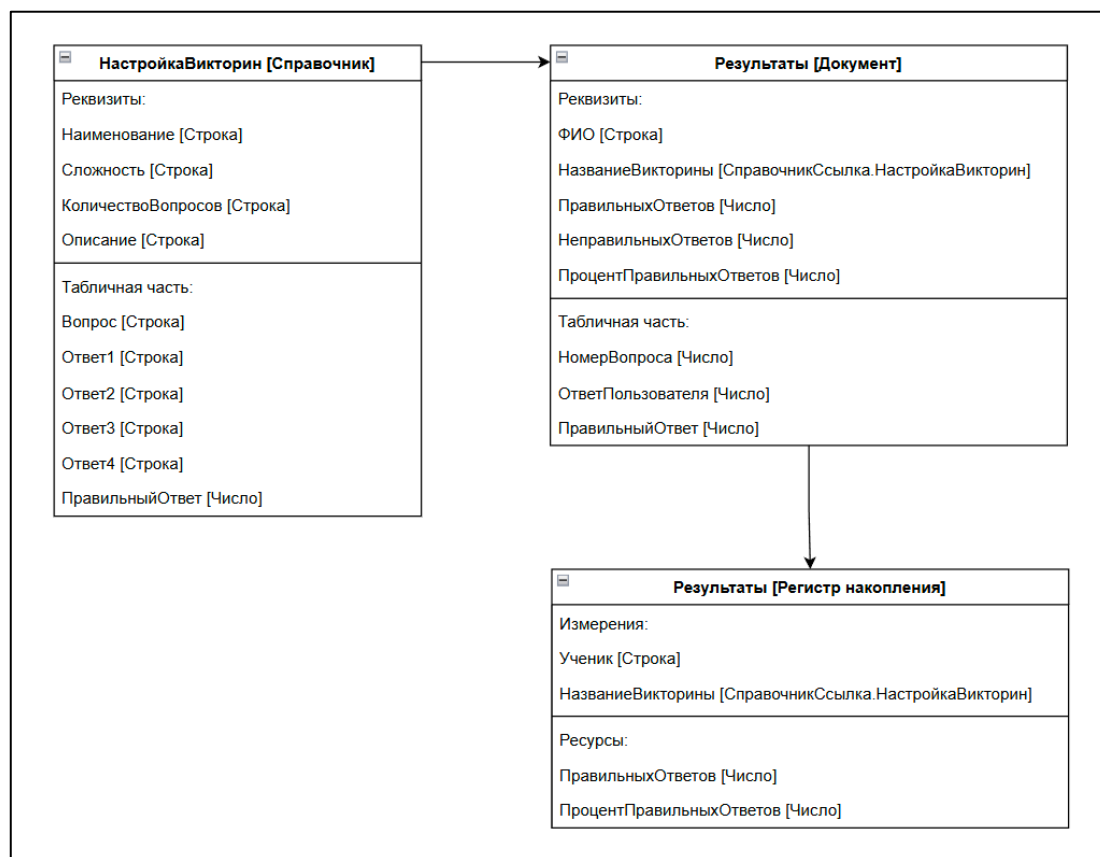


Рисунок 4 – Структура разработанных объектов

Справочник "НастройкаТестов" - используется для хранения информации о созданных тестах.

Реквизиты:

- Наименование (Строка) – название тестов;
- Сложность (Строка) – уровень сложности тестов;
- КоличествоВопросов (Строка) – число вопросов в тесте;
- Описание (Строка) – текстовое описание тестов.

Табличная часть:

- Вопрос (Строка) – текст вопроса;
- Ответ1 (Строка) – первый вариант ответа;
- Ответ2 (Строка) – второй вариант ответа;
- Ответ3 (Строка) – третий вариант ответа;
- Ответ4 (Строка) – четвертый вариант ответа;
- ПравильныйОтвет (Число) – номер правильного ответа.

Документ "Результаты" - Хранит данные о прохождении пользователем тестов.

Реквизиты:

- ФИО (Строка) – фамилия, имя и отчество пользователя;
- НазваниеТеста (СправочникСсылка.НастройкаТестов) – ссылка на пройденный тест;
- ПравильныхОтветов (Число) – количество правильных ответов;
- НеправильныхОтветов (Число) – количество неверных ответов;
- ПроцентПравильныхОтветов (Число) – процент правильных ответов.

Табличная часть:

- НомерВопроса (Число) – порядковый номер вопроса;
- ОтветПользователя (Число) – выбранный пользователем ответ;
- ПравильныйОтвет (Число) – правильный вариант ответа.

Регистр накопления "Результаты" - Используется для хранения статистики прохождения тестов.

Измерения:

- Ученик (Строка) – ФИО пользователя;
- НазваниеТеста (СправочникСсылка.НастройкаВикторин) – ссылка на пройденный тест.

Ресурсы:

- ПравильныхОтветов (Число) – количество правильных ответов;
- ПроцентПравильныхОтветов (Число) – процент правильных ответов.

3 Инструкция к приложению

Для начала прохождения теста необходимо войти в систему и перейти в подсистему «Ученик», после чего выбрать раздел «Тесты» (рисунок 5). Откроется окно со списком доступных тестов, где каждому тесту сопутствует подробное описание, уровень сложности и общее количество вопросов. На этом этапе пользователю рекомендуется ознакомиться с предоставленной информацией, чтобы выбрать наиболее подходящий тест. После принятия решения следует нажать кнопку «Начать тест», расположенную напротив выбранного варианта.

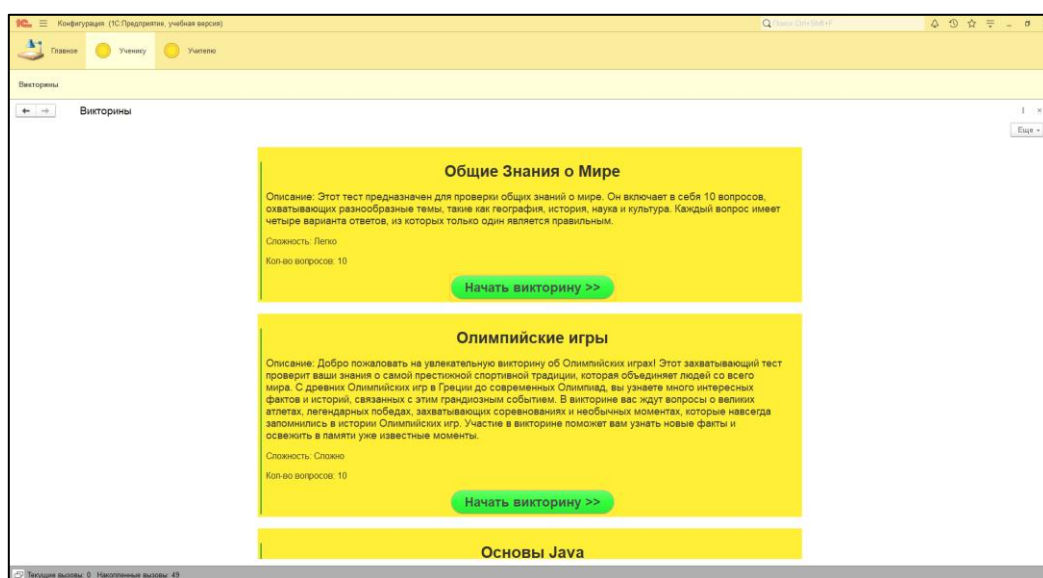


Рисунок 5 – Окно с выбором тестов

После нажатия на кнопку откроется новое окно (рисунок 6), в котором пользователю необходимо ввести свои фамилию, имя и отчество (при наличии). Корректный ввод этих данных позволит корректно сохранить результаты тестирования и сформировать отчёт о его прохождении.

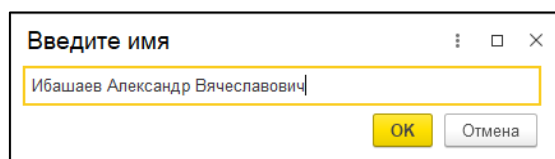


Рисунок 6 – Окно с вводом имени

После ввода данных тестирование начинается автоматически. Вопросы теста последовательно отображаются на экране (рисунок 7), предлагая несколько вариантов ответа. Пользователь должен внимательно ознакомиться с вопросами и выбрать наиболее подходящий вариант ответа. Процесс продолжается до тех пор, пока не будут даны ответы на все вопросы теста.

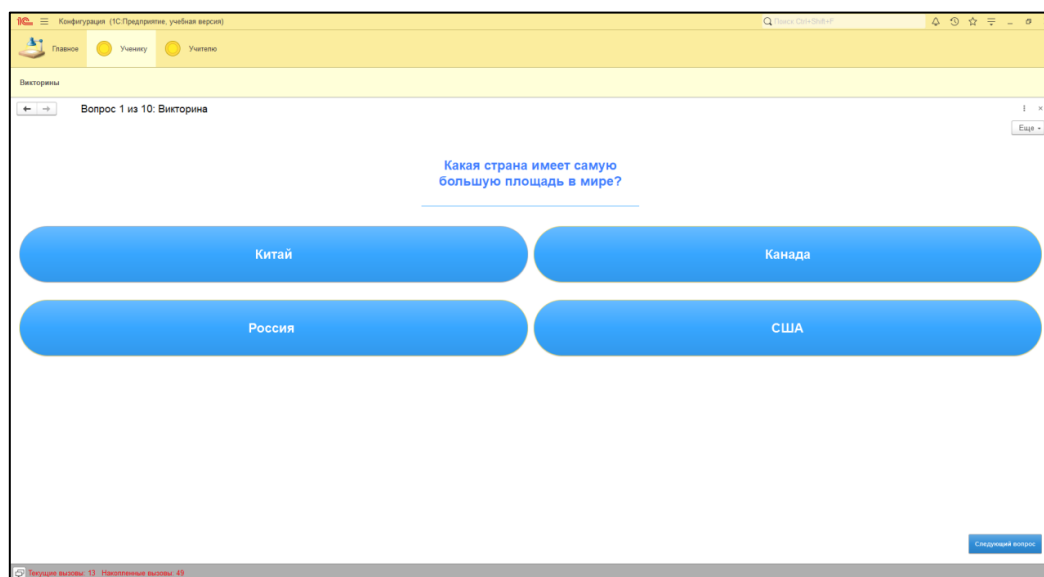


Рисунок 7 – Окно с решением теста

По завершении тестирования система автоматически обрабатывает результаты и отобразит уведомление (рисунок 8) о том, что тест был успешно завершён. Пользователь сможет увидеть итоговое сообщение с подтверждением завершения теста, после чего его результаты сохранятся в системе.

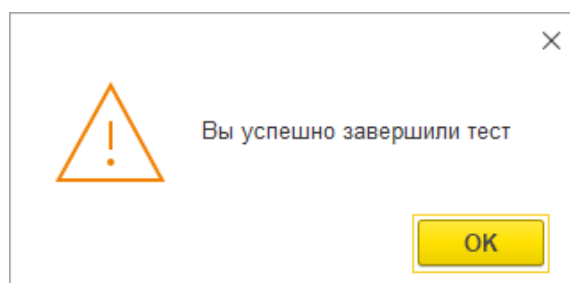


Рисунок 8 – Уведомление после прохождения теста

Для редактирования или создания новых тестов необходимо перейти в подсистему «Учитель» и выбрать раздел «Настройка тестов» (рисунок 9). После этого откроется справочник, содержащий список всех доступных тестов, а также инструменты для их редактирования, удаления или добавления новых.

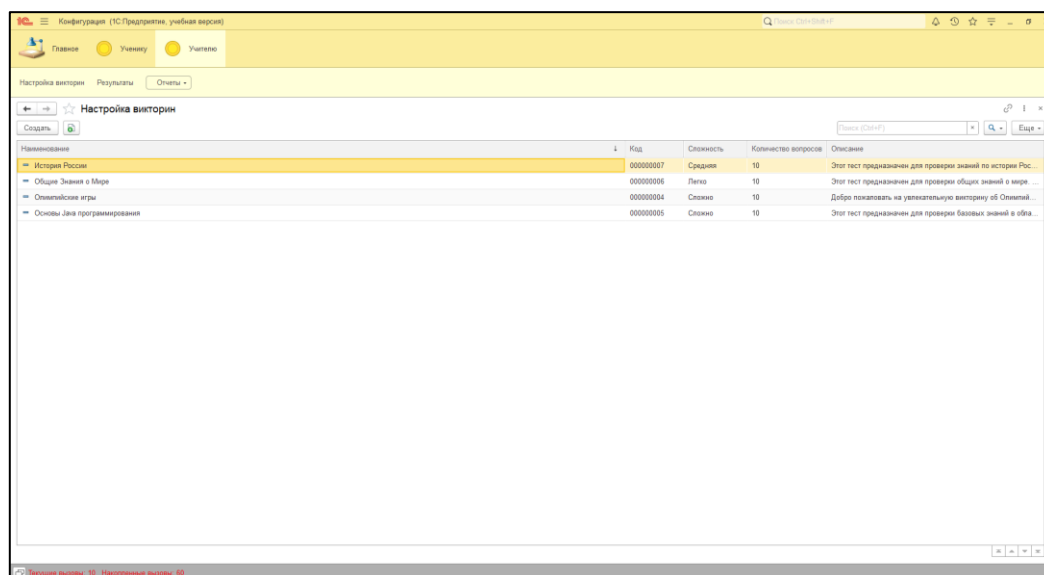


Рисунок 9 – Справочник «Настройка тестов»

При нажатии на запись справочника откроется окно (рисунок 10). В данном окне можно изменять существующие вопросы, корректировать описание тестов, изменять их сложность, а также добавлять новые тесты в систему.

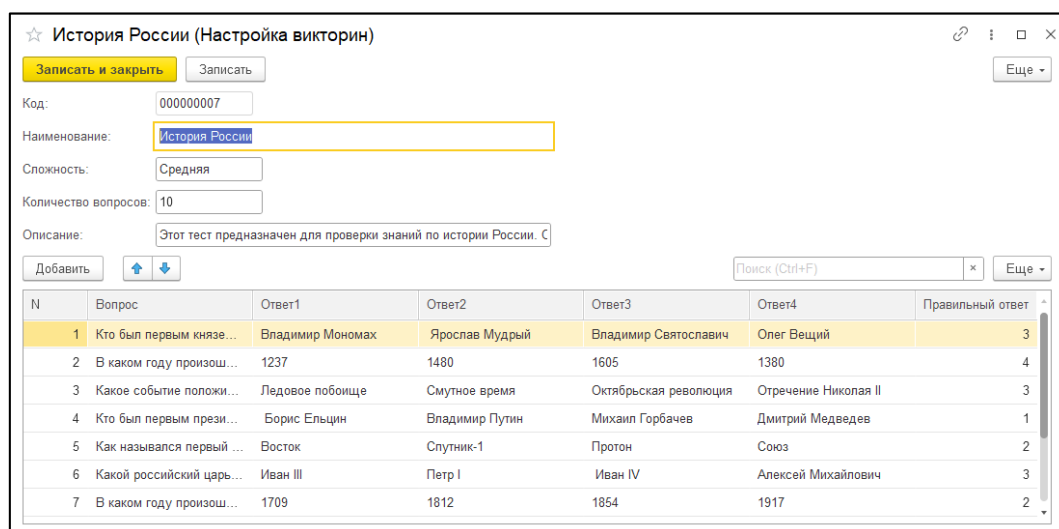


Рисунок 10 – Запись справочника

Чтобы просмотреть результаты пройденных тестов, пользователю необходимо снова перейти в подсистему «Ученик» и открыть раздел «Результаты» (рисунок 11). В этом разделе отображается список всех завершённых тестов, включая сведения о набранных баллах, количестве правильных ответов и общем результате. Это позволяет учителю анализировать успехи учеников.

Дата	№	ФИО	Название викторины	Правильных ответов	Неправильных ответов	Процент правильных ответов
18.06.2024 19:12:59	00000002	Ибашев Александр	Олимпиадные игры	8	2	80,000
18.06.2024 19:52:09	00000013	Котов Данила	История России	8	2	80,000
18.06.2024 19:56:33	00000014	Заманов Азамат	Основы Java программирования	9	1	90,000
18.06.2024 19:56:56	00000010	Маскин Феликс	История России	7	3	70,000
18.06.2024 19:57:34	00000009	Васильев Даниил	Общие Знания о Мире	9	1	90,000
18.06.2024 19:59:52	00000015	Шарифутдинов Динар	Общие Знания о Мире	7	3	70,000

Рисунок 11 – Документ «Результаты»

Запись данного документа выглядит следующим образом (рисунок 12).

№	Номер вопроса	Ответ пользователя	Правильный ответ
1	1	3	3
2	2	1	1
3	3	3	2
4	4	3	4
5	5	3	3
6	6	4	4
7	7	3	3
8	8	2	2
9	9	4	4
10	10	3	3

Рисунок 12 – Запись документа

Для формирования отчёта о прохождении тестирования необходимо открыть подсистему «Ученик» и выбрать раздел «Отчёты». В данном разделе можно сформировать детальный отчёт по результатам тестирования (рисунок 13), который может быть использован для анализа успеваемости, подготовки отчётных документов или дальнейшей работы над ошибками.

Наименование викторины	Ученик	Процент правильных ответов
Общие Знания о Мире	Василий Дружин	90
Основы Java программирования	Земляникова Алена	90
Олимпиадные игры	Ильин Александр	80
История России	Колесников Данила	80
История России	Максим Фисин	70
Общие Знания о Мире	Шарфутдинов Динар	70
Итого		80

Рисунок 13 – Отчет о прохождении тестирования

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была разработана автоматизированная информационная система «Приложение для тестирования учеников», предназначенная для проведения интеллектуальных тестирований и игр. Разработка включала анализ предметной области, выявление проблем текущего состояния, постановку задачи на проектирование, создание структуры базы данных, программной архитектуры и пользовательского интерфейса.

В результате проведенной работы были достигнуты следующие цели и выполнены поставленные задачи:

- Проведен анализ предметной области, выявлены ключевые проблемы существующих решений.
- Разработана концепция системы, включающая основные пользовательские сценарии и функциональные требования.
- Определена структура базы данных, обеспечивающая удобное хранение информации о пользователях, тестах, вопросах и результатах.
- Спроектирована программная архитектура системы, основанная на клиент-серверной модели.
- Разработаны алгоритмы основных функций системы, включая процесс регистрации пользователей, создания тестов и проверки ответов.

Разработанная система обладает рядом преимуществ по сравнению с аналогами: автоматизация проверки ответов, хранение результатов и статистики, гибкость в настройке тестов, поддержка командных и индивидуальных режимов игры.

Предварительные расчеты эффективности системы показали, что внедрение автоматизированного приложения позволит существенно сократить время на проведение тестов, повысить точность проверки ответов, упростить администрирование и анализ результатов.

Таким образом, все поставленные в начале работы задачи были успешно решены, а цель курсовой работы достигнута. Разработанная система может быть использована в образовательных учреждениях, корпоративных тренингах и развлекательных мероприятиях, обеспечивая удобство и гибкость в проведении тестов.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Радченко М.Г., Хрусталева Е.Ю. «1С:Предприятие 8.2.Практическое пособие разработчика» М., ООО «1С- Паблишинг», 2009 г.
2. Габец А.П., Гончаров Д.И., Козырев Д.В. и др. Профессиональная разработка в системе 1С:Предприятие 8. – М.:»1С-Паблишинг», СПб.:Питер, 2006.
3. Гартвич А.В. Планирование закупок, производства и продаж в 1С: Предприятие 8. - М.: совместное издание «1С-Паблишинг» и ИД «Питер», 2007 г.
4. Клепцова О.Ю. Бюджетирование в 1С: Предприятие 8. Информационные технологии бюджетного управления. М.: совместное издание «1С-Паблишинг»и ИД «Питер», 2007 г.
5. Ситосенко Е. Управление заказами в системе программ 1С:Предприятие 8. - М.: «1С — Паблишинг», 2007 г.
6. Шуримова Е.Л. Информационные технологии управления взаимоотношениями с клиентами.- М.: «1С — Паблишинг», 2007 г.
7. Колесников С. Н. Планирование деятельности производственного предприятия от промфинтехпланирования к MRPII и дальше.- М.: «1С — Паблишинг», 2007 г.
8. Берко И.А. Учет и отчетность по МСФП в 1С: Предприятии 8 в вопросах и ответах.- М.:совместное издание «1С-Паблишинг» и ИД «Питер», 2007 г.
9. Рязанцева Н. А., Рязанцев Д. Н. 1С: Предприятие 8.0 Управление производственным предприятием. Секреты работы.— Санкт – Петербург, БХВ – Петербург, 2006 г.

Приложение А

Код модуля формы Викторины:

```
&НаКлиенте
Процедура ВвестиИмя()
Имя = "";
ЗаголовокВопроса = "Введите имя";
Если ВвестиСтроку(Имя, ЗаголовокВопроса, 30, Ложь) Тогда
    ЗапомнитьИмя(Имя);
КонецЕсли;
КонецПроцедуры
```

```
&НаСервере
Процедура ЗапомнитьИмя(Значение)
ПараметрыСеанса.ТекущийПользователь = Значение;
КонецПроцедуры
```

```
&НаКлиенте
Процедура ПриОткрытии(Отказ)
ДобавитьВикторинуНаГлавныйЭкран();
КонецПроцедуры
```

```
&НаСервере
Функция ПолучитьСписокВикторин()
Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
        | НастройкаВикторин.Наименование КАК Наименование,
        | НастройкаВикторин.Сложность КАК Сложность,
        | НастройкаВикторин.КоличествоВопросов КАК
КоличествоВопросов,
        | НастройкаВикторин.Описание КАК Описание,
        | НастройкаВикторин.Код КАК Код
    |ИЗ
        | Справочник.НастройкаВикторин КАК НастройкаВикторин";
РезультатЗапроса = Запрос.Выполнить();
МассивДанных = Новый Массив;
Выборка = РезультатЗапроса.Выбрать();
Пока Выборка.Следующий() Цикл
    СтруктураДанных = Новый Структура;
    СтруктураДанных.Вставить("Сложность", Выборка.Сложность);
    СтруктураДанных.Вставить("Наименование", Выборка.Наименование);
    СтруктураДанных.Вставить("Описание", Выборка.Описание);
    СтруктураДанных.Вставить("КоличествоВопросов",
Выборка.КоличествоВопросов);
    СтруктураДанных.Вставить("Код", Выборка.Код);
    МассивДанных.Добавить(СтруктураДанных);
КонецЦикла;
Возврат МассивДанных;
```

КонецФункции

&НаКлиенте

Процедура НачатьВикторину(Элемент)

Кнопка = ЭтаФорма.ТекущийЭлемент;

Имя = Кнопка.Имя;

Имя = СтрЗаменить(Имя,"НачатьВикторину", "");

ВвестиИмя();

ЗапомнитьВыбраннуюВикторину(Имя);

ОткрытьФорму("ОбщаяФорма.Викторина");

КонецПроцедуры

&НаСервере

Процедура ЗапомнитьВыбраннуюВикторину(Значение)

ПараметрыСеанса.ТекущаяВикторина = Значение;

ПараметрыСеанса.ТекущийВопрос = 1;

КонецПроцедуры

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

КонецПроцедуры

&НаСервере

Процедура СозданиеПанелиВикторины(Элемент)

НовыйЭлемент = Элементы.Добавить("Кнопка1", Тип("КнопкаФормы"),
ЭтаФорма);

НовыйЭлемент.Вид = ВидКнопкиФормы.ОбычнаяКнопка;

НовыйЭлемент.ИмяКоманды = "СчитатьВорон";

КонецПроцедуры

&НаСервере

Процедура ДобавитьВикторинуНаГлавныйЭкран()

СписокВикторин = ПолучитьСписокВикторин();

РодительскийЭлемент = ЭтаФорма;

ИсходнаяГруппа = ЭтаФорма.Элементы.Найти("Викторина");

ИсходноеНазваниеВикторины =

ЭтаФорма.Элементы.Найти("НазваниеВикторины");

ИсходноеОписание = ЭтаФорма.Элементы.Найти("Описание");

ИсходнаяСложность = ЭтаФорма.Элементы.Найти("Сложность");

ИсходноеКоличествоВопросов =

ЭтаФорма.Элементы.Найти("КоличествоВопросов");

ИсходнаяКнопка = ЭтаФорма.Элементы.Найти("НачатьВикторину");

счетчик = 1;

Для Каждого Элемент Из СписокВикторин Цикл

НоваяПанель =

РодительскийЭлемент.Элементы.Добавить("НоваяПанель" + счетчик,

Тип("ГруппаФормы"),РодительскийЭлемент);

НоваяПанель.Вид = ВидГруппыФормы.ОбычнаяГруппа;

НоваяПанель.Высота = ИсходнаяГруппа.Высота;

НоваяПанель.Ширина = ИсходнаяГруппа.Ширина;

НоваяПанель.ЦветФона = ИсходнаяГруппа.ЦветФона;
 НоваяПанель.Отображение = ИсходнаяГруппа.Отображение;
 НоваяПанель.РастягиватьПоГоризонтали =
 ИсходнаяГруппа.РастягиватьПоГоризонтали;
 НоваяПанель.ГоризонтальноеПоложениеВГруппе =
 ИсходнаяГруппа.ГоризонтальноеПоложениеВГруппе;
 НоваяПанель.ГоризонтальноеПоложениеПодчиненных =
 ИсходнаяГруппа.ГоризонтальноеПоложениеПодчиненных;
 Новаяпанель.Группировка = ИсходнаяГруппа.Группировка;
 НазваниеВикторины =
 РодительскийЭлемент.Элементы.Добавить("НазваниеФормы" + счетчик,
 Тип("ДекорацияФормы"),НоваяПанель);
 НазваниеВикторины.Заголовок = Элемент.Наименование;
 НазваниеВикторины.Шрифт = ИсходноеНазваниеВикторины.Шрифт;
 НазваниеВикторины.ВысотаЗаголовка =
 ИсходноеНазваниеВикторины.ВысотаЗаголовка;
 НазваниеВикторины.Высота = ИсходноеНазваниеВикторины.Высота;
 НазваниеВикторины.ГоризонтальноеПоложение =
 ИсходноеНазваниеВикторины.ГоризонтальноеПоложение;
 НазваниеВикторины.РастягиватьПоГоризонтали =
 ИсходноеНазваниеВикторины.РастягиватьПоГоризонтали;
 НазваниеВикторины.АвтоМаксимальнаяШирина =
 ИсходноеНазваниеВикторины.АвтоМаксимальнаяШирина;
 Описание =
 РодительскийЭлемент.Элементы.Добавить("ОписаниеВикторины" + счетчик,
 Тип("ДекорацияФормы"),НоваяПанель);
 Описание.Заголовок = "Описание: " + Элемент.Описание;
 Описание.Шрифт = ИсходноеОписание.Шрифт;
 Описание.РастягиватьПоГоризонтали =
 ИсходноеОписание.РастягиватьПоГоризонтали;
 Описание.ГоризонтальноеПоложениеВГруппе =
 ИсходноеОписание.ГоризонтальноеПоложениеВГруппе;
 Описание.ГоризонтальноеПоложение =
 ИсходноеОписание.ГоризонтальноеПоложение;
 Описание.Высота = ИсходноеОписание.Высота;
 Описание.Ширина = ИсходноеОписание.Ширина;
 Описание.АвтоМаксимальнаяШирина =
 ИсходноеОписание.АвтоМаксимальнаяШирина;
 Сложность =
 РодительскийЭлемент.Элементы.Добавить("СложностьВикторины" + счетчик,
 Тип("ДекорацияФормы"), НоваяПанель);
 Сложность.Заголовок = "Сложность: " + Элемент.Сложность;
 Сложность.Шрифт = ИсходнаяСложность.Шрифт;
 Сложность.ГоризонтальноеПоложениеВГруппе =
 ИсходнаяСложность.ГоризонтальноеПоложениеВГруппе;
 Сложность.ГоризонтальноеПоложение =
 ИсходнаяСложность.ГоризонтальноеПоложение;
 КоличествоВопросов =
 РодительскийЭлемент.Элементы.Добавить("КоличествоВопросовВикторины" +
 счетчик, Тип("ДекорацияФормы"), НоваяПанель);

```

        КоличествоВопросов.Заголовок = "Кол-во вопросов: " +
Элемент.КоличествоВопросов;
        КоличествоВопросов.Шрифт = ИсходноеКоличествоВопросов.Шрифт;
        КоличествоВопросов.ГоризонтальноеПоложениеВГруппе =
ИсходноеКоличествоВопросов.ГоризонтальноеПоложениеВГруппе;
        КоличествоВопросов.ГоризонтальноеПоложение =
ИсходноеКоличествоВопросов.ГоризонтальноеПоложение;
        Кнопка = РодительскийЭлемент.Элементы.Добавить("Начать Викторину"
+ Элемент.код, Тип("КнопкаФормы"), НоваяПанель);
        Кнопка.Высота = ИсходнаяКнопка.Высота;
        Кнопка.Ширина = ИсходнаяКнопка.Ширина;
        Кнопка.Шрифт = ИсходнаяКнопка.Шрифт;
        Кнопка.ИмяКоманды = ИсходнаяКнопка.ИмяКоманды;
        Кнопка.ЦветФона = ИсходнаяКнопка.ЦветФона;
        Кнопка.Заголовок = ИсходнаяКнопка.Заголовок;
        Кнопка.Фигура = ИсходнаяКнопка.Фигура;
        Код = РодительскийЭлемент.Элементы.Добавить("Код" + счетчик,
Тип("ДекорацияФормы"), НоваяПанель);
        Код.Заголовок = Элемент.Код;
        Код.Видимость = Ложь;
        счетчик = счетчик+1;
    КонецЦикла
КонецПроцедуры

```

Приложение Б

Код модуля формы Викторина:

&НаКлиенте

Процедура Считать(Команда)

КоличествоВорон = КоличествоВорон + 1;

КонецПроцедуры

&НаКлиенте

Процедура ПриОткрытии(Отказ)

МассивВопросов = ПолучитьМассивВопросов();

СозданиеДокументаПользователя(МассивВопросов);

ДляПервогоИПоследнегоВопроса(МассивВопросов.Количество());

ВставитьВопрос(МассивВопросов);

КонецПроцедуры

&НаСервере

Процедура ВставитьВопрос(МассивВопросов)

НомерВопроса = ПараметрыСеанса.ТекущийВопрос;

Вопрос = МассивВопросов[НомерВопроса-1];

НазваниеВопроса = ЭтаФорма.Элементы.Найти("НазваниеВопроса");

НазваниеВопроса.Заголовок = Вопрос.Вопрос;

Ответ1 = ЭтаФорма.Элементы.Найти("ВариантОтвета1");

Ответ1.Заголовок = Вопрос.Ответ1;

Ответ1.Доступность = Истина;

Ответ2 = ЭтаФорма.Элементы.Найти("ВариантОтвета2");

Ответ2.Заголовок = Вопрос.Ответ2;

Ответ2.Доступность = Истина;

 Ответ3 = ЭтаФорма.Элементы.Найти("ВариантОтвета3");

Ответ3.Заголовок = Вопрос.Ответ3;

Ответ3.Доступность = Истина;

Ответ4 = ЭтаФорма.Элементы.Найти("ВариантОтвета4");

Ответ4.Заголовок = Вопрос.Ответ4;

Ответ4.Доступность = Истина;

ЭтаФорма.Заголовок = "Вопрос "+ НомерВопроса+" из

" + МассивВопросов.Количество());

КонецПроцедуры

&НаСервере

Функция ДляПервогоИПоследнегоВопроса(КоличествоВопросов = 1)

КнопкаПредыдущийВопрос =

ЭтаФорма.Элементы.Найти("ПредыдущийВопрос");

Если ПараметрыСеанса.ТекущийВопрос = 1 Тогда

 КнопкаПредыдущийВопрос.Видимость = Ложь;

Иначе

 КнопкаПредыдущийВопрос.Видимость = Истина;

КонецЕсли;


```

КнопкаСледующийВопрос = ЭтаФорма.Элементы.Найти("СледующийВопрос");
Если ПараметрыСеанса.ТекущийВопрос = КоличествоВопросов Тогда
    КнопкаСледующийВопрос.Заголовок = "Закончить викторину";
Иначе
    КнопкаСледующийВопрос.Заголовок = "Следующий вопрос";
КонецЕсли;
КонецФункции

Функция ПолучитьМассивВопросов()
    Викторина =
        Справочники.НастройкаВикторин.НайтиПоКоду(ПараметрыСеанса.ТекущаяВикторина);

    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
        | НастройкаВикторинВопросы.НомерСтроки КАК НомерСтроки,
        | НастройкаВикторинВопросы.Вопрос КАК Вопрос,
        | НастройкаВикторинВопросы.Ответ1 КАК Ответ1,
        | НастройкаВикторинВопросы.Ответ2 КАК Ответ2,
        | НастройкаВикторинВопросы.Ответ3 КАК Ответ3,
        | НастройкаВикторинВопросы.Ответ4 КАК Ответ4,
        | НастройкаВикторинВопросы.ПравильныйОтвет КАК
ПравильныйОтвет,
        | НастройкаВикторин.Ссылка КАК Ссылка
    |ИЗ
        | Справочник.НастройкаВикторин.Вопросы КАК
НастройкаВикторинВопросы
        | ЛЕВОЕ СОЕДИНЕНИЕ Справочник.НастройкаВикторин
КАК НастройкаВикторин
        | ПО НастройкаВикторинВопросы.Ссылка =
НастройкаВикторин.Ссылка
    |ГДЕ
        | НастройкаВикторин.Вопросы.Ссылка.Наименование = &Имя";
    Запрос.УстановитьПараметр("Имя", Викторина.Наименование);
    РезультатЗапроса = Запрос.Выполнить();
    МассивВопросов = Новый Массив;
    Выборка = РезультатЗапроса.Выбрать();
    Пока Выборка.Следующий() Цикл
        СтруктураДанных = Новый Структура;
        СтруктураДанных.Вставить("НомерСтроки", Выборка.НомерСтроки);
        СтруктураДанных.Вставить("Вопрос", Выборка.Вопрос);
        СтруктураДанных.Вставить("Ответ1", Выборка.Ответ1);
        СтруктураДанных.Вставить("Ответ2", Выборка.Ответ2);
        СтруктураДанных.Вставить("Ответ3", Выборка.Ответ3);
        СтруктураДанных.Вставить("Ответ4", Выборка.Ответ4);
        СтруктураДанных.Вставить("ПравильныйОтвет",
Выборка.ПравильныйОтвет);
        МассивВопросов.Добавить(СтруктураДанных);
    КонецЦикла;
    Возврат МассивВопросов;

```

КонецФункции

&НаКлиенте

Процедура НажатиеНаВариантОтвета(Команда)

Ответ1 = ЭтаФорма.Элементы.Найти("ВариантОтвета1");

Ответ2 = ЭтаФорма.Элементы.Найти("ВариантОтвета2");

 Ответ3 = ЭтаФорма.Элементы.Найти("ВариантОтвета3");

Ответ4 = ЭтаФорма.Элементы.Найти("ВариантОтвета4");

Если Ответ1.Доступность = Ложь Или Ответ2.Доступность = Ложь Или
Ответ3.Доступность = Ложь Или Ответ4.Доступность = Ложь Тогда

 Ответ1.Доступность = Истина;

 Ответ2.Доступность = Истина;

 Ответ3.Доступность = Истина;

 Ответ4.Доступность = Истина;

КонецЕсли;

ВыбранныйВариант = ЭтаФорма.ТекущийЭлемент;

ОтправитьИнформациюОВыбореПользователя();

ВыбранныйВариант.Доступность = Ложь;

КонецПроцедуры

&НаСервере

Процедура ОтправитьИнформациюОВыбореПользователя()

Кнопка = ЭтаФорма.ТекущийЭлемент;

Имя = Кнопка.Имя;

МассивВопросов = ПолучитьМассивВопросов();

НомерВарианта = СтрЗаменить(Имя, "ВариантОтвета", "");

НомерВопроса = СтрЗаменить(ЭтаФорма.Заголовок, "Вопрос ", "");

НомерВопроса = СтрЗаменить(НомерВопроса, " из " +

МассивВопросов.Количество(), "");

 ОбъектДокумента =

Документы.Результаты.НайтиПоРеквизиту("ИмяПользователя",

ПараметрыСеанса.ТекущийПользователь).ПолучитьОбъект();

 Строка = ОбъектДокумента.ОтветыПользователя[Число(НомерВопроса)-1];

 Строка.ОтветПользователя = Число(НомерВарианта);

 счетчикПравильныхОтветов = 0;

 счетчикНеправильныхОтветов = 0;

Для Каждого Элемент Из ОбъектДокумента.ОтветыПользователя Цикл

 Если Элемент.ОтветПользователя = Элемент.ПравильныйОтвет Тогда
 счетчикПравильныхОтветов = счетчикПравильныхОтветов + 1;

 Иначе

 счетчикНеправильныхОтветов = счетчикНеправильныхОтветов

+1;

КонецЕсли;

КонецЦикла;

ОбъектДокумента.ПравильныхОтветов = счетчикПравильныхОтветов;

ОбъектДокумента.НеправильныхОтветов = счетчикНеправильныхОтветов;

ОбъектДокумента.ПроцентПравильныхОтветов =
(ОбъектДокумента.ПравильныхОтветов/ОбъектДокумента.ОтветыПользователя.Количество()*100;

ОбъектДокумента.Записать();
КонецПроцедуры

&НаКлиенте
Процедура ПредыдущийВопрос(Команда)
ПредыдущийВопросНаСервере();
КонецПроцедуры

&НаСервере
Процедура ПредыдущийВопросНаСервере()
ПараметрыСеанса.ТекущийВопрос = ПараметрыСеанса.ТекущийВопрос-1;
МассивВопросов = ПолучитьМассивВопросов();
ДляПервогоИПоследнегоВопроса(МассивВопросов.Количество());
ВставитьВопрос(МассивВопросов);
КонецПроцедуры

&НаКлиенте
Процедура СледующийВопрос(Команда)
МассивВопросов = ПолучитьМассивВопросов();
Если
СледующийВопросНаСервереИлиЗавершениеВикторины(МассивВопросов) Тогда
ЭтаФорма.Заккрыть();
ОткрытьФорму("ОбщаяФорма.Викторины");
КонецЕсли;
КонецПроцедуры

&НаСервере
Функция
СледующийВопросНаСервереИлиЗавершениеВикторины(МассивВопросов)
ПараметрыСеанса.ТекущийВопрос = ПараметрыСеанса.ТекущийВопрос+1;
Если ПараметрыСеанса.ТекущийВопрос-1 = МассивВопросов.Количество()
Тогда

Возврат Истина;
КонецЕсли;
ДляПервогоИПоследнегоВопроса(МассивВопросов.Количество());
ВставитьВопрос(МассивВопросов);
Возврат Ложь;
КонецФункции

&НаКлиенте
Процедура ЗавершениеВикторины()

КонецПроцедуры

&НаСервере
Процедура СозданиеДокументаПользователя(МассивВопросов)
НовыйДокумент = Документы.Результаты.СоздатьДокумент();

```

        НовыйДокумент.ИмяПользователя = ПараметрыСеанса.ТекущийПользователь;
        НовыйДокумент.НазваниеВикторины =
Справочники.НастройкаВикторин.НайтиПоКоду(ПараметрыСеанса.ТекущаяВикторин
а);
        НовыйДокумент.Дата = ТекущаяДата();
        Для счетчик = 1 По МассивВопросов.Количество() Цикл
            НоваяСтрока = НовыйДокумент.ОтветыПользователя.Добавить();
            НоваяСтрока.ПравильныйОтвет = МассивВопросов[счетчик-
1].ПравильныйОтвет;
            НоваяСтрока.НомерВопроса = счетчик;
        КонечЦикла;
        НовыйДокумент.Записать();
        КонечПроцедуры

```