Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

КОНСТРУКТОР ЭКЗАМЕНАЦИОННЫХ БИЛЕТОВ

Пояснительная записка

к курсовой работе по дисциплине

«Основы алгоритмизации и программирования»

**БГУИР КР 6-05-0612-01 029 ПЗ**

Руководитель В.А. Кабариха

Учащийся гр. А.М. Усов

310901

Минск 2024

Содержание

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

**БГУИР КР 6-05-0612-01 010 ПЗ**

**)**

Разраб.

Усов

Провер.

Василькова

Реценз.

Н. Контр.

Утверд.

Конструктор экзаменационных билетов

Пояснительная записка

Лит.

Листов

35

БГУИР

# Введение

История создания и развития технологий для подготовки экзаменационных билетов

Традиционно подготовка экзаменационных билетов была крайне трудоемким и длительным процессом для преподавателей. Им приходилось вручную составлять сотни вопросов по пройденному материалу, распечатывать их, компоновать в билеты и следить за актуальностью вопросов из года в год. Этот процесс занимал колоссальное количество времени и требовал серьезных усилий.

С развитием компьютерных технологий появились первые программы, позволявшие автоматизировать отдельные этапы создания экзаменационных материалов. Преподаватели могли вводить вопросы в электронные базы данных, формировать из них случайные наборы для билетов. Это несколько ускоряло процесс, но по-прежнему требовало много ручного труда.

Современные тенденции заключаются в разработке специализированных приложений-конструкторов для максимальной автоматизации процесса на всех этапах - от создания шаблонов билетов и банка вопросов до формирования комплектов билетов и их печати. Такие решения способны существенно экономить время преподавателей, снижать вероятность ошибок, обеспечивать актуальность материалов.

Однако существующие программные продукты часто обладают ограниченным функционалом, плохо интегрируются с другими системами учебного заведения, имеют неудобный и запутанный интерфейс. Зачастую они решают лишь локальные задачи автоматизации, в то время как комплексный подход отсутствует.

Цель данной курсовой работы - создание универсального программного решения "Конструктор экзаменационных билетов" для комплексной автоматизации всего процесса подготовки экзаменационных материалов в учебных заведениях.

Для достижения поставленной цели были сформулированы следующие задачи:

- Провести анализ требований к экзаменационным билетам в различных учебных заведениях

- Изучить существующие программные решения, выявить их достоинства и недостатки

- Спроектировать архитектуру приложения с учетом требований масштабируемости и расширяемости

- Разработать интуитивно понятный пользовательский интерфейс

- Реализовать функционал по созданию гибких шаблонов билетов и формированию банка вопросов

- Обеспечить возможность генерации комплектов билетов по заданным правилам

- Предусмотреть различные варианты вывода подготовленных билетов (печать, электронный формат)

Разработанное приложение "Конструктор экзаменационных билетов" позволит преподавателям существенно сэкономить время и усилия на создание актуальных экзаменационных материалов высокого качества. Использование готовых шаблонов и систематизированного банка вопросов упростит и ускорит этот процесс. Понятный интерфейс обеспечит легкость в освоении и работе с программой.

Таким образом, данная курсовая работа ориентирована на решение актуальной проблемы в сфере образования и имеет важную практическую ценность для учебных заведений разных уровней. Внедрение "Конструктора экзаменационных билетов" позволит автоматизировать рутинные операции и высвободить время преподавателей для более важных задач.

В ходе выполнения курсовой работы предполагается изучить существующие аналоги и лучшие практики в области автоматизации подготовки экзаменационных материалов. На основе этого анализа будут сформулированы требования к разрабатываемому программному обеспечению, спроектирована его архитектура и определены оптимальные технологии реализации.

Особое внимание планируется уделить проектированию удобного и интуитивно понятного пользовательского интерфейса, который позволит преподавателям быстро осваивать работу с программой без дополнительного обучения.

Одной из ключевых задач является реализация гибкой системы шаблонов для создания различных видов экзаменационных билетов с возможностью их настройки под специфику учебного заведения и предмета. Формирование банка вопросов предполагается организовать с возможностью импорта из внешних источников и категоризации по темам.

Важной частью работы станет реализация функционала генерации комплектов билетов по заданным правилам, например, с учетом обязательного включения определенных вопросов или исключения повторяющихся. Также необходимо предусмотреть различные варианты вывода подготовленных материалов - печать, сохранение в электронном виде и т.д.

По завершении разработки планируется провести тестирование созданного программного обеспечения с привлечением реальных преподавателей для оценки его функциональности и удобства использования. Результаты тестирования лягут в основу выводов и рекомендаций по дальнейшему совершенствованию "Конструктора экзаменационных билетов".

Таким образом, выполнение данной курсовой работы позволит создать современное и эффективное программное решение, способное значительно упростить процесс подготовки экзаменационных материалов в учебных заведениях. Это обеспечит экономию времени преподавателей, повысит качество и актуальность экзаменационных билетов, а также облегчит процесс проверки знаний студентов.

# 1 Требования к программе

## 1.1 Описание предметной области

Основная задача курсовой работы заключается в разработке программного средства «Конструктор экзаменационных билетов».

Предметная область – это сфера образования и подготовки к экзаменам, которая имеет существенное значение и непосредственное отношение к функционалу разрабатываемого программного средства.

Учебные компьютерные программы, такие как «Конструктор экзаменационных билетов», позволяют моделировать процесс создания и проверки билетов, являясь эффективным инструментом для обучения. В то же время, такая программа позволяет визуализировать абстрактные объекты и процессы, которые в реальном мире не существуют, а являются моделями реальных объектов и явлений в области образования.

Процесс представления предметной области задачи в виде набора объектов, обменивающихся сообщениями, называется объектной декомпозицией. В рамках этого процесса основные элементы задачи делятся на классы. Для каждого класса определяются его свойства и поведение, а также взаимодействие с другими классами.

В разрабатываемой системе главными объектами являются экзаменационные билеты и информация, содержащаяся в них. Пользователь запускает программу, выбирает предмет, тему и уровень сложности билетов, после чего программа генерирует экзаменационные билеты для дальнейшего анализа и использования.

Язык программирования: C++

Парадигма программирования: функциональное программирование

Средство разработки: Microsoft Visual Studio

Способ хранения данных: Файловая система, текстовые файлы

## 1.2 Функциональное назначение

Курсовой проект "Конструктор экзаменационных билетов" предназначен для автоматизации процесса создания и управления экзаменационными билетами. Программа должна предоставлять пользователям удобные инструменты для генерации билетов по различным темам и предметам, а также возможности для настройки уровня сложности и количества вопросов в каждом билете.

Компьютерная версия "Конструктора экзаменационных билетов" должна обеспечивать интуитивно понятный интерфейс и автоматизацию процессов создания, редактирования и анализа билетов. Приложение предназначено для оказания помощи преподавателям, экзаменаторам и другим специалистам в сфере образования, позволяя им эффективно разрабатывать и управлять экзаменационными материалами.

Основные задачи курсового проекта "Конструктор экзаменационных билетов":

- Предоставление удобных инструментов для создания и редактирования экзаменационных билетов по различным темам и предметам.

- Обеспечение гибких настроек уровня сложности и количества вопросов в каждом билете.

- Реализация функций автоматической генерации билетов на основе заданных параметров.

- Предоставление возможностей для анализа и оценки результатов экзаменов.

Основные функции и возможности программы:

– Создание и редактирование вопросов и ответов для экзаменационных билетов.

– Настройка уровня сложности и количества вопросов в каждом билете.

– Автоматическая генерация билетов на основе заданных параметров (тема, уровень сложности, количество вопросов и т.д.).

– Импорт и экспорт данных, позволяющие пользователям загружать собственные вопросы и сохранять билеты в различных форматах (PDF, Word, Excel).

– Функция автоматической оценки результатов экзамена на основе ответов пользователя.

– Генерация отчетов и статистики по результатам экзаменов.

– Удобный и интуитивно понятный пользовательский интерфейс.

Программа "Конструктор экзаменационных билетов" должна обеспечивать следующие функции:

- Информационная функция – предоставление пользователям доступа к базе данных вопросов и ответов по различным темам и предметам.

- Обучающая функция – помощь преподавателям в создании эффективных экзаменационных материалов, способствующих проверке знаний и навыков учащихся.

- Аналитическая функция – предоставление инструментов для анализа результатов экзаменов и оценки эффективности экзаменационных билетов.

Вывод: Разрабатываемая программа "Конструктор экзаменационных билетов" позволит автоматизировать процесс подготовки, проведения и анализа экзаменов в учебных заведениях. Она предоставит преподавателям удобные инструменты для создания билетов по индивидуальным шаблонам, формирования банка вопросов, генерации комплектов и проведения тестирования. Функционал просмотра рейтинга и экспорта на печать упростит анализ результатов

Таким образом, "Конструктор экзаменационных билетов" станет современным и многофункциональным программным решением, которое позволит преподавателям экономить время на рутинных операциях по подготовке экзаменов и сконцентрироваться на более важных аспектах учебного процесса. Автоматизация всех этапов работы с билетами значительно упростит эту задачу и повысит эффективность проверки знаний студентов.

Кроме того, использование готовых шаблонов и централизованного банка вопросов обеспечит соблюдение единых стандартов и требований при формировании экзаменационных материалов в рамках учебного заведения. Это повысит качество и объективность оценки знаний обучающихся.

С точки зрения заказчика, внедрение "Конструктора экзаменационных билетов" позволит решить ряд актуальных проблем, связанных с трудоемкостью подготовки экзаменов, низкой эффективностью существующих процессов и отсутствием комплексной автоматизации. Данное программное обеспечение станет полезным инструментом, способствующим оптимизации рабочих процессов и повышению качества образовательных услуг.

# 2 Конструирование программы

## 2.1 Описание модулей программы

Давайте подробно опишем модули программы «Конструктор экзаменационных билетов», уделяя внимание их функциональности и назначению.

Модуль авторизации и регистрации

Этот модуль отвечает за идентификацию пользователей и контроль доступа к различным частям программы. Он позволяет новым пользователям создавать учетные записи, а существующим пользователям – входить в систему, используя свои учетные данные. В этом модуле реализованы функции безопасности для шифрования паролей и защиты данных пользователя. Он также обрабатывает восстановление забытых паролей и управление правами доступа различных ролей пользователей.

Модуль управления тестами

Модуль управления тестами является центром для создания и редактирования тестовых вопросов и тестов. Преподаватели могут использовать его для составления вопросов, выбора типа вопроса (один верный ответ, несколько верных ответов, открытый вопрос и т.д.), добавления ответов и указания правильных ответов. Он также позволяет группировать вопросы в тесты и устанавливать параметры тестирования, такие как время прохождения, порядок вопросов и критерии оценки.

Модуль работы с билетами

Этот модуль автоматизирует процесс создания экзаменационных билетов из предварительно подготовленных вопросов. Он позволяет настраивать количество вопросов в билете, их сложность и темы. Модуль также поддерживает печать билетов для оффлайн-экзаменов и экспорт в различные форматы для электронного использования.

Модуль тестирования

Модуль тестирования предназначен для проведения тестирования студентами. Он предоставляет пользовательский интерфейс для выбора и прохождения тестов, отслеживает время ответа на вопросы и автоматически оценивает ответы согласно заранее установленным правилам. Результаты тестов сохраняются для последующего анализа.

Модуль анализа результатов и рейтингов

Этот модуль собирает статистику по результатам тестов и генерирует рейтинги успеваемости. Он позволяет преподавателям просматривать общие и индивидуальные результаты, анализировать эффективность тестов и вносить корректировки в учебный процесс. Модуль также включает инструменты для создания отчетов и экспорта данных для дальнейшего анализа.

Модуль экспорта данных

Модуль экспорта данных обеспечивает функциональность для преобразования тестов и билетов в различные форматы (например, PDF, DOCX), что упрощает их печать и распространение. Он также поддерживает сохранение результатов тестирования в электронной форме для удобства хранения и доступа.

Каждый из этих модулей разработан таким образом, чтобы выполнять конкретные функции в рамках программы, и представляет собой неотъемлемый компонент в общей архитектуре системы. Интеграция модулей друг с другом обеспечивается через четко определенные интерфейсы и протоколы взаимодействия, что позволяет достичь модульности и упрощает масштабирование приложения.

Модуль административного управления

Этот модуль предназначен для администраторов системы, позволяя им управлять пользователями, их ролями, доступом к функционалу и настройками системы в целом. Администраторы могут добавлять новые учетные записи, блокировать или удалять существующие, а также настраивать параметры безопасности и резервного копирования данных.

## 2.2 Описание хранения данных в программе

В программе "Конструктор экзаменационных билетов" данные хранятся в файловой системе в виде иерархии текстовых файлов. Этот подход обеспечивает простоту реализации, кроссплатформенность и возможность ручного редактирования файлов при необходимости.

Иерархия файлов организована следующим образом:

./users/users.txt – файл, где хранятся данные все пользователей в определенном формате где каждая строка имеет данные другого пользователя:

«[Имя Пользователя] [Логин] [Пароль]»

./tests/publicTests.txt – файл, где хранятся данные общественного доступных тестов,

Которые при создании имели значение поля isPublic как True

./tests/[Имя Пользователя]Test.txt – вид файлов, где хранятся данные тестов каждого

пользователя в формате:

«[Имя теста] [Описание теста] [Имя автора теста] [Публичный ли тест] [Количество обычных вопросов] [Количество вопросов с вариантами ответов] /[вопросы]/ /[ответы]/ [Сложность вопроса]

Такой способ хранения данных пользователей позволяет очень удобно хранить данные пользователя в быстро доступном формате.

## 2.3 Описание пользовательских функций

1. Выбор теста

Функция которая при вызове возвращает выбранный пользователем тест

После вызова Происходит чтение файла с тестами и пользователь выбирает тест

Test GetChoice()

2. Функция экспорта теста

Функция которая принимает два параметр – количество вопросов в билете , Количество билетов, возвращает массив билетов

При вызове этой функции все вопросы с помощью рандомайзера распределяет определенное количество вопросов на определенное количество билетов

Bilet \* CreateExamBilets(int num\_questions, int num\_bilets)

3.Функция редактирования вопроса

Функция которая на входе принимает объект теста , на выходе уже отредактированный тест

При вызове этой функции вначале указывается что пользователь хочет отредактировать, а далее происходит выбор нужного поля и его редактирование

Test EditTest(Test myTest)

4. Функция создания тестов

Функция которая на входе ничего не принимает и возвращает созданный тест

При вызове по очереди пользователем заполняются поля и после тест записывается в файл

Test CreateTest()

5. Функция прохождения теста

Функция на входе принимает сам тест который пользователь заранее выбрал , на выходе возвращается рейтинг за этот тест в результате прохождения

Int TryTest(Test test)

6. Функция выбора пользователя

Функция на вход ничего не принимает и на выходе возвращает пользователя

(Только для администратора)

Person GetPerson()

7. Функция редактирования пользователя

Функция на вход принимает пользователя , а на выходе возвращает уже отредактированного пользователя

(Только для администратора)

Person EditPerson(Person Person)

7.Функция редактирования рейтинга

Функция на входе принимает тест , на выходе ничего не возвращает

Функция берет рейтинг выбранного теста и по полям предлогает каждое отредактировать

(Только для администратора)

Void EditReiting(Test Test)