Титульник

содержание

**Введение**

Цифровые технологии сильно изменили жизнь человека. Компьютер предоставляет большие возможности для работы с разного вида информацией.

В настоящее время существует большое количество программ, которые позволяют создавать и проходить тесты. Компьютер как средство для тестирования имеет свои достоинства и недостатки. Однако в объективности оценки ему отказать трудно. Видимо, именно этот фактор способствует расширению сферы применения компьютерного тестирования - оно успешно используется в школах, вузах, при поступлении на работу.

Сегодня разработчики предлагают разнообразные компьютерные инструменты для подготовки тестовых заданий, начиная от различных редакторов и программ для разработки презентаций и до использования языков программирования и возможностей сети Интернет. Но разработка качественного тестового инструментария – длительный, трудоемкий и дорогостоящий процесс.

Целью данного курсового проекта является разработка программного средства, позволяющего создавать и проходить тесты. В качестве интерфейса прикладного программирования был выбран обширный API-интерфейс – Windows Presentation Foundation (WPF), предназначенный для создания настольных программ с графически насыщенным пользовательским интерфейсом. В основе графической технологии WPF лежит мощная инфраструктура, основанная на DirectX. Это является одним из основных отличий WPF от более ранней технологии создания пользовательских интерфейсов – Windows Forms.

Для работы с WPF использовался объектно-ориентированный язык программирования с С-подобным синтаксисом – С#, разработанный для создания приложений на платформе Microsoft .NET Core. Кроме того, используемая в WPF модель разделения кода и дизайна предполагала работу с декларативным языком описания интерфейса –eXtensible Application Markup Language (XAML).

1.Теоретическое введение

1.1 Объектно-ориентированное программирование

Объектно-ориентированное программирование — это метод программирования, при использовании которого главными элементами программы являются понятия объектов и классов.

ООП возникло в результате развития идеологии процедурного программирования, где данные и подпрограммы (процедуры, функции) их обработки формально не связаны. В центре ООП находится понятие объекта.

Объект — это сущность, экземпляр класса, в которой можно посылать сообщения, и которая может на них реагировать, используя свои данные. Данные объекта скрыты от остальной программы. Сокрытие данных называется инкапсуляцией.

Наличие инкапсуляции достаточно для объектности языка программирования, но ещё не означает его объектной ориентированности — для этого требуется наличие наследования.

Но даже наличие инкапсуляции и наследования не делает язык программирования в полной мере объектным с точки зрения ООП. Основные преимущества ООП проявляются только в том случае, когда в языке программирования реализован полиморфизм; то есть возможность объектов с одинаковой спецификацией иметь различную реализацию.

Абстрагирование — это способ выделить набор значимых характеристик объекта, исключая из рассмотрения не значимые. Инкапсуляция — это свойство системы, позволяющее объединить данные и методы, работающие с ними в классе, и скрыть детали реализации от пользователя. Наследование — это свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником или производным классом. Полиморфизм — это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.

1.2 C#

C# – язык программирования, сочетающий объектно-ориентированные и контекстно-ориентированные концепции. Разработан в 1998-2001 годах группой инженеров под руководством Андерсa Хейлсбергa в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств вроде Visual Studio.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML. Переняв многое от своих предшественников – языков C++, Delphi, Modula и Smalltalk – С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++).

//дописать

**2. Обзор прототипов**

Программное средство *«Конструктор тестов»*

Конструктор тестов - программа для создания и проведения тестов. Приложение позволяет:

* создавать тесты из пяти различных типов вопросов;
* форматировать текст в вопросах;
* выводить тесты и результаты тестирования на печать;
* проверять орфографию;
* настраивать интерфейс приложения;
* использовать неограниченное количество тем и вопросов;
* задавать каждому вопросу индивидуальную оценку в баллах;
* искать вопросы в базе;
* включать в вопросы мультимедийные файлы (картинки, звук, видео).

Программа содержит три модуля: один предназначен для составления тестов и заполнения базы вопросов (рисунок 1.1); второй для проведения тестирования и третий - для оценки результатов тестирования.

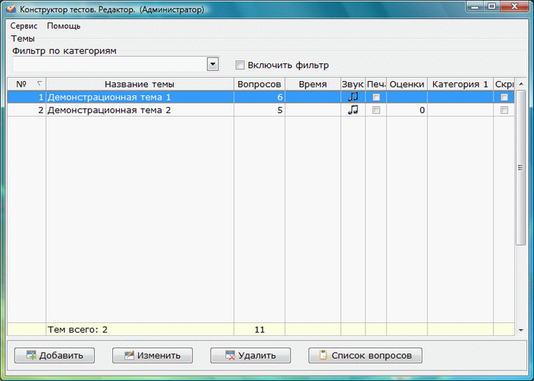


Рисунок 1.1 – Интерфейс для составления теста и заполнения базы вопросов

Тестируемый имеет возможность пропускать сложные вопросы. Преподаватель может приостанавливать тестирование, ограничивать его временными рамками, а также централизованно получать данные о прохождении тестов. Поддерживается разграничение прав доступа, резервное копирование и синхронизация базы. С сайта разработчиков программы можно загрузить множество вариантов готовых тестов на самую разнообразную тематику: ПДД, ЕГЭ, русский, французский и английский языки, экология, информатика, математика и др.

Программное средство «Универсальный тест»

Универсальный тест - приложение для создания тестов и проведения тестирования. Программа содержит в себе два модуля. Модуль под названием Tester позволяет проверять знания учащихся, студентов и персонала при помощи готовых тестов на самую разнообразную тематику - по русскому и английскому языку, информатике, ОБЖ, геометрии, физике, медицине, социологии, религиоведению, бухгалтерскому учету и т.д. База имеющихся тестов состоит из трех групп - тестов для школьников, тестов для ВУЗов, ПТУ и техникумов и тестов для проверки профессиональных знаний. Эти тесты составлялись пользователями предыдущих версий программы. Во время тестирования перед пользователем выводится серия вопросов с вариантами ответов (рисунок 1.2). Вопросы выбираются из базы в случайном порядке.

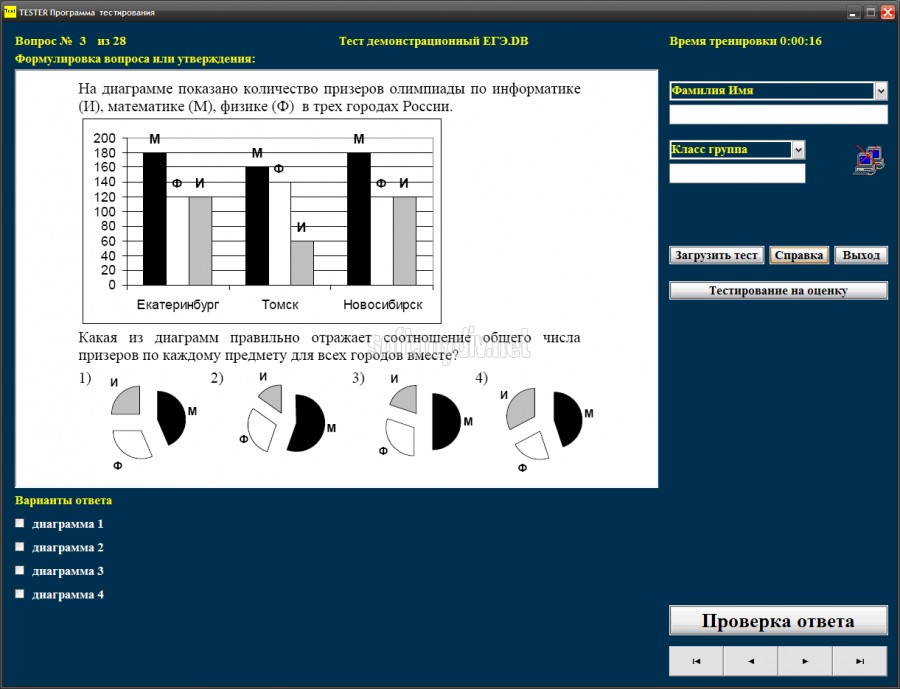


Рисунок 1.2 – Интерфейс для тестирования

При помощи модуля Test maker можно составлять и собственные тесты. В них можно использовать графику и тексты с форматированием. Доступ к модулю Test maker предоставляется либо на платной основе, либо в обмен на составленный пользователем собственный тест. Имеется возможность тестирования по локальной сети и сбора полученных результатов при помощи встроенного сервера.

Программное средство *«AnsTester»*

AnsTester - приложение для создания разнообразных тестов, основанных на системе начисления баллов. AnsTester изначально предназначался для проведения компьютерного тестирования в средней школе. Помимо тестирования, программа позволяет проводить различные опросы. В пакет входят несколько модулей - один предназначен для проведения тестирования (рисунок 1.3), второй - для просмотра результатов тестов и третий - для создания тестов.

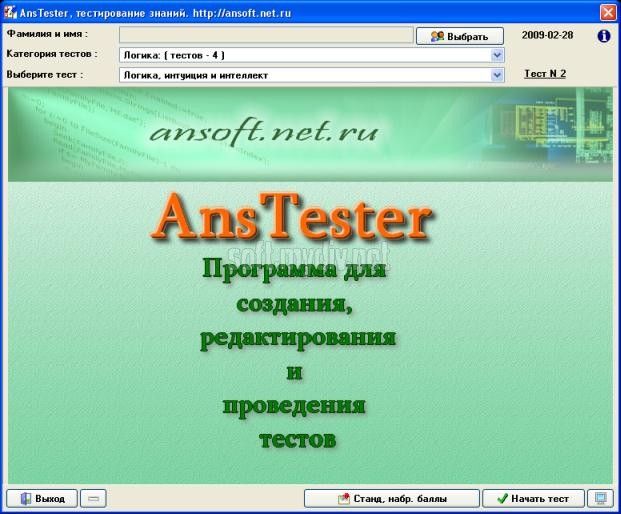


Рисунок 1.3 – Интерфейс для проведения тестирования