**Технологии, использованные при разработке.**

* **ООП и среда Delphi 7**

**Объектно-ориентированное программирование.**

В качестве среды разработки для создания программы я выбрал Borland Delphi 7. В данной среде реализована технология *Объектно-Ориентированного Программирования* *(ООП).* Каждый компонент программы является объектом, имеющим свойства, методы и события. Такой подход позволяет наиболее точно моделировать реальный мир и процессы, происходящие в нем.

Важнейшими атрибутами, определяющими суть и природу объекта, являются свойства, методы и события:

* Свойства – переменные различных типов, характеризующие объект. Допустим, мы хотим описать модель какого-либо физического объекта. Тогда для него имеют место быть такие свойства, как масса, объем, плотность, цвет, скорость, и т. п. Каждое свойство имеет тип данных, в зависимости от определяющей характеристики. Например, имя сотрудника хранится в строковой переменной, а мощность реактивного двигателя выражена численным значением, следовательно, для него отводится числовая переменная.
* Методы – процедуры, применяемые к объекту, для совершения с ним каких-либо действий (таких как создание или удаление объекта, изменение его свойств и т. д.).
* События – изменения состояния объекта, которые вызывают определенную реакцию. Например, события возникают, когда пользователь производит какие-либо операции с объектом, например нажатие клавиши или наведение курсора мыши.

Существуют три основных принципа объектно-ориентированного программирования:

* Инкапсуляция – объединение данных объекта в классе. Класс является своеобразным образцом объекта, на основе которого создаются его экземпляры. Инкапсуляция позволяет разделять данные объекта на приватные (private), защищенные (protected) и открытые (public). К приватным данным невозможно получить доступ извне класса, защищенные доступны только для чтения (применимо к свойствам), к открытым осуществляется свободный доступ.
* Наследование – создание нового класса на основе уже существующего с сохранением функционала объекта-предка и добавлением новых свойств, методов и событий (по необходимости). Данный подход дает возможность создавать различные модификации на основе единого объекта.
* Полиморфизм – единый интерфейс для различных типов данных. Один и тот же метод может быть применен к объектам разного типа и разной конструкции. При этом для каждого типа объектов такой метод может быть реализован по-разному.

**Интегрированная среда разработки Delphi 7.**

Визуальное программирование позволяет создавать программу не только посредствам написания кода, но и с применением графического интерфейса, что делает работу программиста намного удобнее. Все объекты и внешний вид программы проектируются заранее, настраиваются свойства, методы и события объектов.

Delphi 7 является интегрированной средой разработки, использующей компилятор языка Object Pascal. Помимо интерфейсов визуального программирования она содержит инструменты отладки и набор вспомогательных программ (для проектирования баз данных, создание графических файлов и т. д.).

* **Базы данных**

При работе программы используется реляционная база данных, состоящая из одной таблицы.

*Реляционная база данных* – база данных, состоящая из связанных таблиц. Каждая таблица хранит сведения об однотипных объектах, имеющих одинаковый набор свойств. Каждая запись в таблице отвечает за отдельный объект, а каждый столбец за определенное свойство.

База данных электронной тестирующей программы, созданной в рамках данной научно-исследовательской работы, состоит из одного файла (таблицы) с расширением .db. При помощи серверной части программы возможен ее просмотр, редактирование, добавление и удаление записей в разделе «журнал». Также доступна функция фильтрации, сортировки и поиска по записям. База данных поставляется вместе с исполняемым файлом и автоматически перемещается в необходимую директорию при первом запуске программы.

* **Клиент-серверное взаимодействие**

Клиент-серверное взаимодействие осуществляется по протоколу TCP/IP с применением локальной IP-адресации.

Во время работы клиент-серверной системы соединение с сервером инициируется клиентом, после чего сервер создает отдельный поток для обмена данными с клиентом.

Закодированные тестирования отправляются клиенту, после чего расшифровываются и визуализируются. Клиент вводит результаты, после чего клиентская программа собирает их и отправляет обратно на сервер. Сервер обрабатывает полученные данные (сверяет с шаблонами правильных ответов), после чего автоматически выставляет оценку и сохраняет результаты в базу данных.

* **Тестирование**

Тестирование программы проводилось с помощью виртуальной машины Oracle VM VirtualBox, а также с применением компьютеров разных операционных систем, находящихся в одной локальной сети. С помощью Oracle VM VirtualBox можно эмулировать различные операционные системы с настраиваемым набором выделяемых физических ресурсов (оперативная память, видеопамять, объем выделяемого пространства на жестком диске, разрядность системы).

Данное тестирование позволило выявить системные требования для программы, исправить баги и ошибки, а также повысить эффективность работы программы.

* **Системные требования**

Программа не использует сложных графических алгоритмов и не производит высоконагруженных вычислений, для стабильной работы требуется устройство с операционной системой Windows 7/8/8.1/8.x.

 Что касается процессора/оперативной памяти и прочего физического оборудования, то данные системные требования можно считать эквивалентными требованиям установленной операционной системы. Также для работы программы требуется сетевая карта и сетевое оборудование (сетевой кабель для подключения к локальной сети или возможность подключения к Wi-Fi сетям, которая есть на большинстве современных устройств).

Примечание: сетевое оборудование не требуется при работе серверной программы в режиме создания учебных материалов, их можно создать и сохранить заранее, до подключения к сети.

* **Общие принципы работы программы**

Схема работы программы и ее взаимодействия с файловой системой и локальной сетью представлена ниже:

**СЕРВЕР**

Интерфейс создания учебных материалов

ДИСК

Оперативная память

Запись данных

Чтение данных

Компонент клиент-серверного взаимодействия

База данных

Запись в БД

Интерфейс редактирования БД

**КЛИЕНТ**

**КЛИЕНТ**

**КЛИЕНТ**

Обмен

Данными

Обработка информации

Обработка информации

Обработка информации

**КЛИЕНТ**

Обработка информации

* **Особенности программы**
1. Portable (программа не требует установки).

Обе части программы, как серверная, так и клиентская не требуют установки и могут быть запущены не только с жесткого диска, но и со съемных носителей. Однако при этом в качестве рабочей директории рекомендуется использовать папку на носителе, с которого запускается исполняемый файл, в противном случае после извлечения внешнего носителя база данных и прочие служебные файлы, в которых хранятся данные и настройки программы останутся на компьютере и могут быть утеряны.

 При использовании программы постоянно на одном устройстве рекомендуется использовать рабочую директорию, отличную от хранилища файлов и данных пользователя, чтобы пользователь не смог случайно удалить их, ведь тогда корректная работа программы будет не возможной. а в худшем случае программа и вовсе не запустится.

1. Программа не нуждается в большом количестве системных ресурсов (см. системные требования).

Эта особенность является очень важной, в частности при использовании на слабых бюджетных устройствах, или на устройствах, загруженных большим количеством других системных задач (например, майнингом биткоинов).

1. Все сервера, используемые программой локальны, не требуется использование хостинга и подключения к интернету.

Это особенно важно, когда программу используют сразу много людей и сетевое соединение может быть перегружено. Также не требуются средства на оплату хостинга и услуг системного администратора, обслуживающего удаленный сервер. В качестве сервера может выступать любой компьютер локальной сети.

* **Существующие аналоги, их преимущества и недостатки**

В настоящее время существует колоссальное количество различных образовательных систем, программ и ресурсов, реализованных в виде программного продукта, веб-сайта или мобильного приложения. Каждый из них имеет определенные практические спецификации, предназначение и технические особенности. Рассмотрим несколько из наиболее известных и широко распространенных, в том числе используемых нашим образовательным учреждением.

**LMS Moodle.**

LMS Moodle – образовательная система, созданная в 1999 году. Она представляет широкие возможности для создания образовательных веб-ресурсов. Используется в том числе и в нашем образовательном учреждении. Как и любой веб-ресурс система LMS Moodle мультиплатформенна, доступ к ней можно получить при помощи устройств с различными операционными системами, в том числе с мобильных устройств. В LMS Moodle присутствуют специальные интерфейсы, позволяющие преподавателю составлять собственные учебные материалы для конкретного учебного заведения. LMS Moodle автоматически проверяет и выставляет оценки за многие тестовые задания, что значительно облегчает работу преподавателя.

Что касается практического применения, LMS Moodle очень удобна для выполнения домашних заданий, ученик может сделать это в любой точке земного шара, где есть доступ в интернет. Однако использование данной образовательной системы в классе, когда большое количество учеников одновременно используют подключение к сети, перегружает сеть и снижает эффективность работы.

**Электронный дневник школьника (ЭЖД).**

ЭЖД – государственная образовательная система учета успеваемости и автоматизации образовательного процесса, пришедшая на смену неудачному проекту «МРКО», работоспособность которого оставляла желать лучшего. На данный момент в ЭЖД помимо дневника имеется возможность использования электронных учебников и сдачи домашних заданий посредствам выгрузки файла на сервер. Однако не имеет функций автоматической проверки и среды выполнения заданий.