МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ГОСУДАРСТВЕННОЕ ВЫСШЕЕ УЧЕБНОЕ ЗАВЕДЕНИЕ

«ПРИАЗОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

МАРИУПОЛЬСКИЙ МЕХАНИКО-МЕТАЛЛУРГИЧЕСКИЙ КОЛЛЕДЖ

Разработать автоматизированную информационную подсистему учёта работы учителя-логопеда Мариупольской психолого-медико-педагогической консультации при дошкольном заведении №166

ДИПЛОМНЫЙ ПРОЕКТ

Пояснительная записка

ДП.05010301.10.105.00.000.ПЗ

Разработал Хараман В.С.

Руководитель проекта Кривошеева Е.Н.

Консультанты:

Григоренко Т.Г.

Шичева О.Н.

2014

Реферат

Тема моего дипломного проекта состоит в следующем: разработать автоматизированную информационную подсистему учёта работы учителя-логопеда Мариупольской психолого-медико-педагогической консультации при дошкольном заведении №166.

Состоит из: реферата, содержания, введения, специальной части, производственной и экологической безопасности при разработке автоматизированной информационной подсистемы учёта работы учителя-логопеда, экономики и организация производства, выводов, перечня использованных источников и приложений.

Специальная часть включает в себя:

* анализ предметной области (описание предметной области, основные концепции, положенные в основу дипломного проекта, критическая характеристика существующих проектных решений и программного обеспечения существующего класса);
* постановка задачи (наименование программы, назначение и область применения программы, требования к функциональным характеристикам программы, требования к надежности программы, требования к квалификации и численности персонала, требования к технологии хранения и обработки информации проектируемой задачи, требования к системе управления базами данных, требования к составу и параметрам технических средств, требования к исходным кодам и языкам программирования, требования к программным средствам, используемым программой, требования к организации входных и выходных данных, требования к защите информации и программ, предварительный состав программной документации);
* моделирование проектированной задачи (диаграмма вариантов использования, диаграмма базы данных, диаграмма деятельности, диаграмма компонентов и развёртываний);
* программное обеспечение (обоснование выбора СУБД, обоснование выбора языка программирования, обоснование выбора инструментальных возможностей программной реализации задачи, инструкция пользователя);

Производственная и экологическая безопасность при разработке автоматизированной информационной подсистемы учёта работы учителя-логопеда включает в себя:

* производственную и экологическую безопасность, где необходимо рассмотреть факторы, которые определяют условия труда пользователя работающего с персональным компьютером, а также рассмотреть вопрос об обеспечении требований эргономики и технической эстетики;
* обеспечение электромагнитной безопасности при эксплуатации компьютерной техники.

Организационно-экономическая часть включает в себя:

* введение;
* оценка расходов на создание нового программного продукта, которое включает определение заработной платы исполнителя, которая состоит из заработной платы исполнителя по тарифу, суммы премии, основной заработной платы, дополнительной заработной платы, общей заработной платы; расчет отчислений на социальное страхование, расходы на расходные материалы, расходы на электроэнергию, амортизационные отчисления, непрямые расходы, плановые накопления;
* оценка экономической эффективности.

Выводы, включают в себя краткие итоги результатов выполненной работы и достигнутой при этом эффективности, а также затрагиваются вопрос перспективы развития и использования проектных решений.

Перечень использованных источников.

Приложение А – Листинг программы.

Приложение Б – Диаграмма вариантов использования.

Приложение В – Экономическая эффективность.

Пояснительная записка содержит 105 с., 48 рисунков, 7 таблиц, 17 формул, 3 приложения.

КЛЮЧЕВЫЕ СЛОВА: АИС,ЛОГОПЕД, БАЗА ДАННЫХ, СУБД, С++, phpMyAdmin, UML-ДИАГРАММЫ, MYSQL, ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ

Содержание

|  |  |
| --- | --- |
| Перечень основных обозначений и сокращений.............................................. | 8 |
| Введение............................................................................................................... | 9 |
| 1 Специальная часть |  |
| * 1. Анализ предметной области |  |
| 1.1.1 Описание задачи.…………………………………………………. | 10 |
| 1.1.2 Маркетинговое исследование ….…………………………....…... | 12 |
| 1.2 Постановка задачи ……………………………………………………..... | 14 |
| 1.2.1 Наименование программы. Назначение и область применения программы…………………………………………………………………....... | 14 |
| 1.2.2 Требования к функциональным характеристикам программы... | 14 |
| 1.2.3 Требования к надежности программы………………………....... | 28 |
| 1.2.4 Требования к квалификации и численности персонала………... | 28 |
| 1.2.5 Требования к технологии хранения и обработки информации проектируемой задачи. Требования к системе управления базами данных.. | 29 |
| 1.2.6 Требования к составу и параметрам технических средств…….. | 29 |
| 1.2.7 Требования к исходным кодам и языкам программирования… | 29 |
| 1.2.8 Требования к программным средствам, используемым программой………………………………………………………………….…. | 30 |
| 1.2.9 Требования к организации входных и выходных данных ....….. | 30 |
| 1.2.10 Требования к защите информации и программ……………...... | 30 |
| 1.2.11 Предварительный состав программной документации..……... | 31 |
| 1.3 Моделирование проектируемой задачи |  |
| 1.3.1 Диаграмма вариантов использования ………………..………..... | 31 |
| 1.3.2 Диаграмма деятельности ……………………………….……….. | 32 |
| 1.3.3 Диаграмма базы данных ………………………………….……... | 33 |
| 1.3.4 Диаграмма компонентов и развёртываний …..……………..….. | 34 |
| 1.4 Программное обеспечение |  |
| 1.4.1 Обоснование выбора системы управления базами данных...….. | 34 |
| 1.4.2 Обоснование выбора языка программирования ……….………. | 36 |
| 1.4.3 Обоснование выбора инструментальных возможностей программной реализации задачи ………………………………………...…... | 38 |
| 1.4.4 Инструкция пользователя ……………………………………….. | 38 |
| 2 Производственная и экологическая безопасность при разработке автоматизированной информационной подсистемы учёта работы учителя-логопеда Мариупольской психолого-медико-педагогической консультации при дошкольном заведении №166 |  |
| 2.1 Производственная и экологическая безопасность................................... | 54 |
| 2.1.1 Факторы, которые определяют условия труда пользователя работающего с персональным компьютером ….………………………..…... | 54 |
| 2.1.2 Обеспечение требований эргономики и технической эстетики.. | 59 |
| 2.2 Обеспечение пожаро- и взрывобезопасности …………...….……………………….……………… | 59 |
| 3 Организационно-экономическая часть |  |
| 3.1 Введение.................................................................................................... | 61 |
| 3.2 Оценка затрат на создание нового программного продукта................ | 61 |
| 3.3 Оценка экономической эффективности................................................... | 67 |
| Вывод….................................................................................................................. | 70 |
| Перечень использованных источников............................................................... | 71 |
| Приложение А Листинг программы....................................... ……………....... | 72 |
| Приложение Б Диаграмма вариантов использования..............………….…… | 11 104 |
| Приложение В Экономическая эффективность................................................. | 105 |

Перечень основных обозначений и сокращений

|  |
| --- |
| Case – Computer Aided Software Engineering  SQL – Structure Query Language (структурированный язык запросов)  UML – Unified Modeling Language (унифицированный язык моделирования)  БД – база данных |
| ПК – персональный компьютер |
| ПО – программное обеспечение |
| СУБД – система управления базами данных |
| ЭВМ – электронно-вычислительная машина |
| ПМПК – психолого-медико-педагогическая консультация |
| ИС – информационная система |
| ЛП – логопедический пункт  ОС – операционная система  ПЭВМ – персональная электронно-вычислительная машина  АИС – автоматизированная информационная система  ИСС – информационная справочная система |
| ФИО – Фамилия Имя Отчество |
| ООП – [Объектно-ориентированное программирование](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) |
|  |

Введение

В настоящее время обработка и хранение информации не является чисто теоретической задачей. Потеря информации или ее несвоевременное получение могут обернуться потерей денег. Именно этим обстоятельством можно объяснить столь бурный рост компьютерной техники и стремительное развитие электронных таблиц и систем управления базами данных (СУБД) в нашей стране и за рубежом. Для оперативного, гибкого и эффективного управления предприятиями, фирмами и организациями различных форм собственности, телекоммуникационными средствами гражданского и военного назначения, информационно – вычислительными, экологическими, радиолокационными системами широко внедряются системы автоматизированного управления, ядром которых являются базы данных (БД). При большом объеме информации и сложности, производимых с ней операций проблема эффективности средств организации хранения, доступа и обработки данных приобретет особое значение. Учитывая важность и значимость баз данных в современной жизни, весьма серьезные требования предъявляются к квалификации специалистов, создающих приложения на их основе.

В связи с положением вещей с интернетом в образовательных заведениях появилась возможность автоматизации процессов учета информации и безопасной ее хранения, что открывает принципиально новое стратегическое направление развитие сферы образовательной деятельности, открывает новые возможность ускорения, упрощения и оптимизации процессов обработки и учета информации образовательным заведением.

Самым большим преимуществом автоматизированной информационной системы является существенное сокращение затрат времени на рутинные операции.

Методологической и информационной базой исследования являются научные публикации периодических изданий, работы украинских и зарубежных исследователей, посвященные экономико-организационной теоретической и методической проблемам управления и проблемам компьютерной безопасности, вопросам Интернет-технологий, базам данных и программированию.

Целью моего дипломного проекта является разработать автоматизированную информационную подсистему учёта работы учителя-логопеда Мариупольской психолого-медико-педагогической консультации при дошкольном заведении №166, обеспечить комфортную работу с документациями учителя-логопеда, а также функции поиска и отчёты.

1 Специальная часть

* 1. Анализ предметной области

1.1.1 Описание задачи

Психолого-медико-педагогические консультации являются методическими учреждениями системы образования и науки Украины, осуществляющими консультативную, методическую, психолого-педагогическую, коррекционно-развивающую, аналитическую, прогностическую, профилактическую, просветительскую деятельность.

ПМПК города Мариуполя выполняет функцию городской службы по вопросам определения вида обучения. Деятельность психолого-медико-педагогических консультаций направлена ​​на:

-выявление, психолого-педагогическое изучение, оценку труда и потенциальных возможностей развития детей в возрасте до 18 лет, которые нуждаются в коррекции физического и (или) умственного развития, имеют признаки риска возникновения трудностей познавательной деятельности и поведения;

-подготовку заключения и рекомендаций по развитию, содержанию, форм и методов обучения с учетом особенностей учебно-познавательной деятельности ребенка;

-консультирование родителей (лиц, их заменяющих), педагогических работников, по вопросам выбора возможных форм и методов обучения, в том числе инклюзивного (интегрированного), в сочетании с реабилитационными мероприятиями, социальной адаптации и интеграции в общественную жизнь детей, нуждающихся в коррекции физического и (или) умственного развития;

-учет, систематизацию и анализ данных о количестве детей, нуждающихся в коррекции физического и (или) умственного развития, по образовательным потребностям с учетом видов нарушений в регионе с целью предоставления предложений органам управления образованием по созданию надлежащих условий и принятия мер для реализации права таких детей на получения образования, обеспечения ее гибкости и разнообразия в местах проживания;

-участие в составлении индивидуальной программы реабилитации ребенка с инвалидностью (совместно с врачебно-консультационными комиссиями учреждений здравоохранения);

-организацию индивидуальной коррекционно-развивающей помощи детям, которые нуждаются в коррекции физического и (или) умственного развития, прежде всего со сложными и тяжелыми нарушениями;

-сотрудничество со школьными психолого-медико-педагогическими комиссиями специальных общеобразовательных школ (школ-интернатов);

-оказание научно-методической помощи педагогическим работникам общеобразовательных учебных заведений по вопросам инклюзивного (интегрированного) обучения и создания оптимальных условий для реализации потенциальных возможностей, развития способностей, талантов, получения качественного образования детьми с особыми образовательными потребностями;

-системную психолого-педагогическую поддержку детей с особыми образовательными потребностями, которые учатся в условиях инклюзивного (интегрированного) обучения, по индивидуальной форме в домашних условиях;

-психолого-педагогическое сопровождение семей, воспитывающих ребенка со сложными нарушениями развития, в том числе с инвалидностью;

-привлечения родителей (лиц, их заменяющих) к участию в реабилитационном процессе путем их участия в проведении коррекционно-развивающих занятий, предоставление методических рекомендаций по развитию и воспитания их детей;

-просветительскую деятельность среди населения в целях профилактики возникновения нарушений в психическом развитии детей, трудности в обучении и формировании личности.

В консультации проводится диагностика интеллектуальных способностей ребенка, его психической зрелости и готовности к школьному обучению; выявление индивидуальных особенностей развития, личностных особенностей и возможностей ребенка к обучению; разработка рекомендаций по уточнению, а в необходимых случаях и изменению программы обучения; проведение медицинской диагностики состояния соматического и нервно-психического здоровья; проведение компьютерных энцефалографии и эхо-энцефалографии.

Все виды услуг детям, проживающим на территории города Мариуполя, - бесплатны.

Учитель-логопед:

-проводит углублённое логопедическое обследование детей дляопределения структуры и степени выраженности имеющегося дефекта;

-комплектует группы для занятий с учётом патологии речи в период с 1 по 15 сентября;

- планирует направления и содержание индивидуальной и групповой коррекционной работы, методической работы на учебный год (с сентября по июнь включительно);

- проводит индивидуальные и групповые занятия по коррекции выявленных нарушений;

- использует разнообразные формы, методы, приёмы и средства обучения врамках государственного стандарта;

- консультирует педагогов, родителей (лиц, их заменяющих) по применению методов и приёмов оказания помощи детям, по результатам коррекции;

- предоставляет председателю ПМПК, отчёт о динамике развития речи детей, имеющих нарушения речи;

- оформляет документы в пределах своей компетенции для представления ребёнка на ПМПК;

- способствует формированию культуры личности, социализации детей.

Для того чтобы попасть в логопедический пункт необходимо обратиться в детский сад, затем обследовавшийся у врачей и специалистов прийти в ПМПК и получить выписку из протокола для ребёнка и уже тогда идти к логопеду.

Оформляет документацию установленного образца:

* планы работы учителя-логопеда;
* карты речевого развития;
* индивидуальные карты развития детей;
* книга анализа результативности работы;
* график работы;
* картотека дидактических игр;
* расписание занятий;
* список детей, зачисленных в ЛП;
* речевой экран.

Учитель-логопед передаёт районному логопеду речевой экран и список зачисленных детей каждый год.

Будут автоматизированы все документации, которые оформляет учитель-логопед. Система должна осуществлять формирование отчётов речевого экрана и списка зачисленных детей на данный логопедический пункт и поиск.В главном меню будут ссылки для перехода на соответствующую документацию и дальнейшего редактирования и учёта данных. Любая документация будет отображаться в виде таблицы или списка.  
Учёт работы учителя-логопеда будет намного комфортней и эффективней с данной программой.

Архитектура данной информационной системы – распределённая клиент-серверная. В клиент-серверных ИС база данных и СУБД находятся на сервере, а на рабочих станциях находятся клиентские приложения. Программа будет разрабатываться в среде EmbarcaderoRADStudio 2010. Так как эта среда обладает большой функциональностью и достаточно комфортна при работе с ней.

В наши дни любая организация должна пользоваться возможностями, которые дают современные технологии, иначе перспективного роста ей не видать.

1.1.2 Маркетинговое исследование

Рассмотрим готовые решения, способные заменить разрабатываемое программное обеспечение по теме «Автоматизированная информационная подсистема учёта работы учителя-логопеда Мариупольской психолого-медико-педагогической консультации при дошкольном заведении №166». Среди них можно отметить: «Играем и учимся», «Ваш логопед».

Программный продукт 1: "Играем и учимся"

Играем и учимся - это программное обеспечение которое помогает развивать ребёнка как дома так и в руках учителя-логопеда. А также делает занятия интересней и эффективней.

Основные функции:

* путём игр, развивает фонематическое представление у детей, формирует правильное произношение, обучает звуковому анализу и делению слов на слоги;
* заполнение журнала посещаемости преподавателей и детей;
* хранения информации о каждом ребёнке;
* комплексная характеристика навыков, полученных им в процессе обучения;
* взаимодействия с общезаведенческой системой, реализованной с использованием СУБД Oracle 10;
* формирования комплексных отчетов в различных разрезах (преподаватели, педагоги, дисциплины, учебные группы, студенты, учебные периоды);
* печать осуществляется путем экспорта отчетов, писем и графиков в Microsoft Word, Microsoft Excel;

Результаты внедрения:

* мониторинг успеваемости и сохранности студентов, отслеживать факты неуспеваемости и своевременно принимать меры;
* упрощение и уменьшение рутинной работы преподавателей и педагогов.

Программный продукт 2: «Ваш логопед»

Ваш логопед - объектом автоматизации является рабочее место любого учителя-логопеда. Учитель-логопед с помощью данной системы, установленной на персональный компьютер, управляет основными функциями учебной работы.

Основные функции:

* автоматическое составление расписания занятий;
* распределение и контроль учебной нагрузки;
* заполнение графиков консультаций;
* составление планов учебно-логопедической деятельности.

Результаты внедрения:

* упрощение и уменьшение рутинной работы учителя-логопеда.

Оценка целесообразности разработки нового программного продукта.

Сведём полученные результаты в итоговую таблицу для принятия решения о целесообразности разработки нового программного продукта (Таблица 1).

Таблица 1 – Сводные данные маркетингового исследования

|  |  |  |
| --- | --- | --- |
| Наименование ПО | Процент соответствия поставленной задаче | Стоимость |
| Программный продукт 1: «Играем и учимся»; | 80 % | 5000,00 |
| Программный продукт 2: «Ваш логопед» | 60 % | 4000,00 |
| Разрабатываемый программный продукт (АИС учёта работы учителя-логопеда ПМПК при детском саде №166) | 100 % |  |

Программный продукт 1 ("Играем и учимся") почти удовлетворяет поставленной задаче. Однако функциональность данного программного обеспечения не является максимальной, и множество функций, которые необходимы сейчас для учёта и работы с детьми - отсутствуют. К тому же распространение данного программного продукта является коммерческим.

Программный продукт 2 («Ваш логопед») не полностью соответствует требованиям поставленной задачи. Является конечным вариантом, что говорит о невозможности добавление необходимых функций. Так же данное приложение является устаревшим и не поддерживает современные ОС (Windows 7,8). Данный программный продукт распространяется на коммерческой основе.

Все вышеперечисленные аспекты подтверждают целесообразность разработки нового ПО. Для осуществлении поставленных выводов необходимо провести экономический анализ эффективности. Расчёты будут проведены в разделе 3.

1.2 Постановка задачи

1.2.1Наименование программы. Назначение и область применения программы

Наименование программы «Автоматизированная информационная система «Учёт работы учителя-логопеда»».

Назначение программы. Автоматизированная информация система обеспечит интеграцию интерактивных возможностей коммуникации Учитель-логопед – Заведующая ПМПК, функционала учёта документаций учителя-логопеда, и составление графика работы непосредственно от заведующей ПМПК. Станет удобным инструментом для специалистов в области преподавания логопедии.

Область применения программы. Создаваемая автоматизированная информационная система может применяться учителем-логопедом и заведующей ПМПК с целью использования современного комплексного специализированного инструментария для их коммуникаций.

1.2.2 Требования к функциональным характеристикам программы

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм. Навигационные элементы должны быть выполнены в удобной для пользователя форме. Средства редактирования информации должны удовлетворять принятым соглашениям в части использования функциональных клавиш, режимов работы, поиска, использования оконной системы. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям системы.

Интерфейс должен быть рассчитан на преимущественное использование манипулятора типа «мышь», то есть управление системой должно осуществляться с помощью набора экранных меню, кнопок, значков и т.п. элементов. Клавиатурный режим ввода должен используется главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм.

Планируется реализация системы на русском языке. Все модули системы, надписи экранных форм должны быть представлены на русском языке, в том числе и сообщения выдаваемые пользователю, за исключением системных сообщений.

Экранные формы должны проектироваться с учетом требований унификации:

- все экранные формы пользовательского интерфейса должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;

- для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие (навигационные) элементы. Термины, используемые для обозначения типовых операций (добавление информационной сущности, редактирование поля данных), а также последовательности действий пользователя при их выполнении, должны быть унифицированы;

- внешнее поведение сходных элементов интерфейса (реакция на наведение указателя «мыши», переключение фокуса, нажатие кнопки) будут реализованы одинаково для однотипных элементов.

В системе должна быть предусмотрена авторизация. Это будет сделано для того, чтобы не произошло несанкционированного доступа к данным программы. Доступ к системе должен осуществляться с компьютера, подключенного к сети Интернет.

Форма «Авторизация»

На рисунке 1 изображена форма авторизации программы.

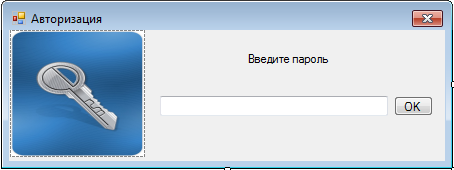


Рисунок 1 – Авторизация

На этой форме необходимо ввести пароль программы, чтобы её запустить. Пароль знает системный администратор, и он сообщает его учителю-логопеду. Это необходимо чтобы защитить данные от несанкционированного доступа.

Ограничения на данной форме:

- Поле для ввода пароля – может содержать в себе лишь символы латинского и русского алфавита, длиной не менее 3 символов, не больше 20.

Форма «Главное меню»

На рисунке 2 изображена форма главного меню программы.

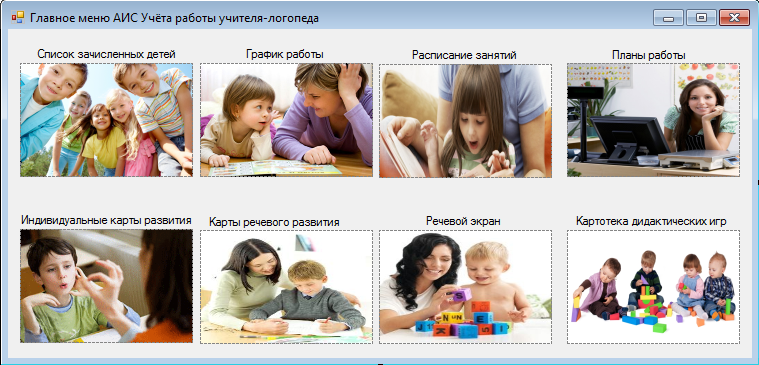


Рисунок 2 - Главное меню программы

На главной форме находятся рисунки, которые являются кнопками – средством перехода между другими формами.

Форма «Список детей зачисленных на ЛП»

На рисунке 3 изображена форма в которой хранится список детей зачисленных на данный логопедический пункт.

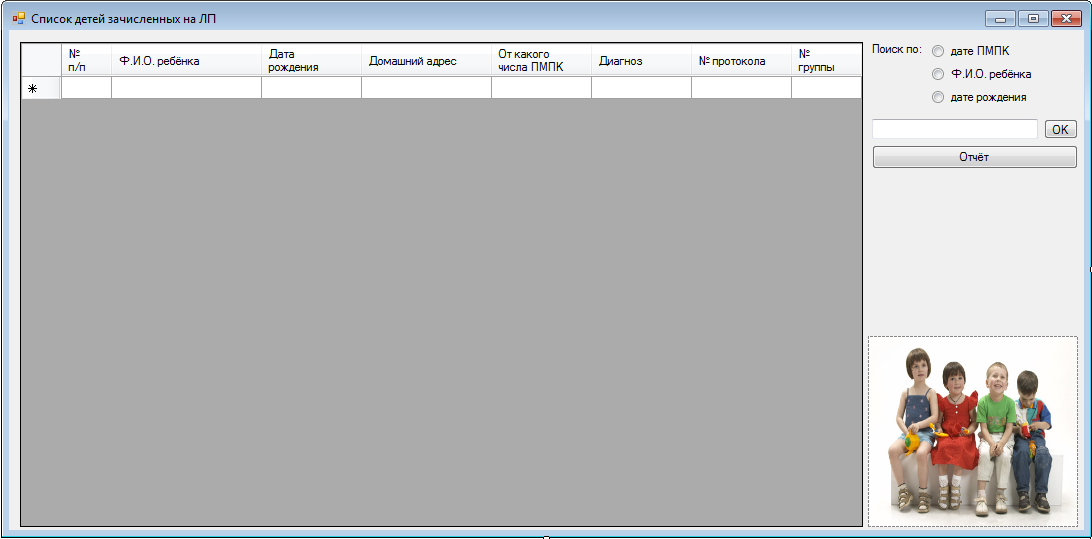


Рисунок 3 – Список детей зачисленных на логопедический пункт

Ограничения на данной форме:

- № п/п – поле типа счётчик, автоматически нумерует записи по порядку;

- Ф.И.О. ребёнка – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Дата рождения – обязательное поле в формате ДД.ММ.ГГ.

- Домашний адрес –обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

-От какого числа ПМПК – обязательное поле в формате ДД.ММ.ГГ.

- Диагноз - обязательное поле, длина не меньше 3 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

- № протокола –обязательное числовое поле, длина не меньше 3 символов, не больше 50 символов;

- № группы - обязательное числовое поле, длина не меньше 3 символов, не больше 50 символов.

Также на форме присутствует поиск по трем полям: От какого числа ПМПК, Ф.И.О. ребёнка и дата рождения; Присутствует кнопка «Отчёт», при нажатии на которую открывается отчёт списка детей зачисленных на ЛП.

Форма «Отчёт Список детей зачисленных на ЛП»

На рисунке 4 изображена форма, в которой хранится отчёт «Список детей зачисленных на логопедический пункт».

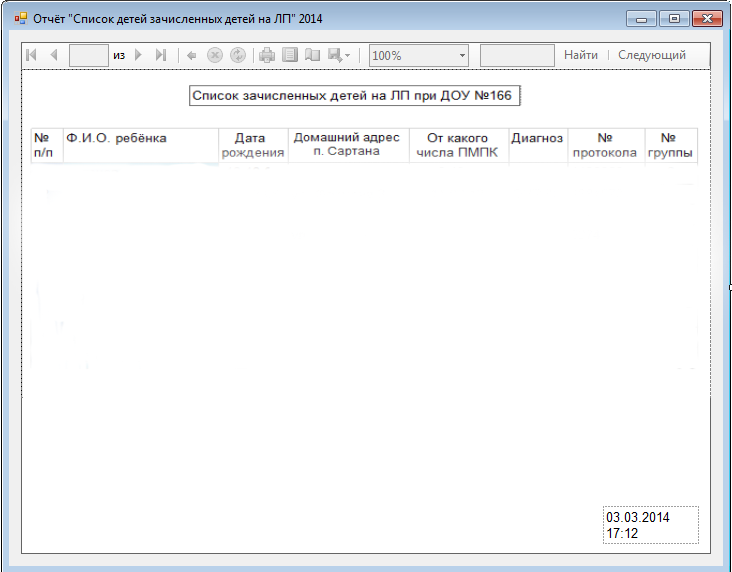


Рисунок 4 – Отчёт «Список детей зачисленных на ЛП»

На данной форме генерируется отчёт из предыдущей формы «Список детей зачисленных на ЛП», ниже отображается дата отчёта.

Форма «График работы»

На рисунке 5 изображена форма в которой хранится график работы учителя-логопеда.

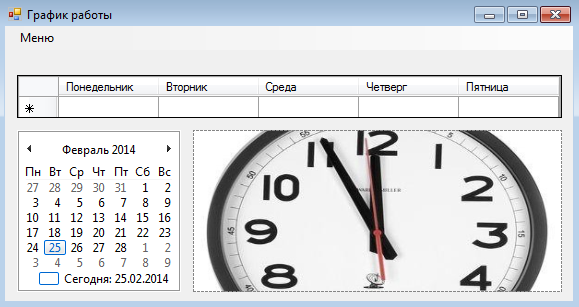


Рисунок 5 – График работы

Ограничения на данной форме:

- Понедельник – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

- Вторник – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

- Среда – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

- Четверг – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

- Пятница – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

На форме присутствует инструмент Календарь, для более удобной работы с формой. График работы составляет заведующая ПМПК и поэтому редактирование данной таблицы с этой формы невозможно.

Форма «Расписание занятий»

На рисунке 6 изображена форма в которой будет храниться расписание занятий для учителя-логопеда.

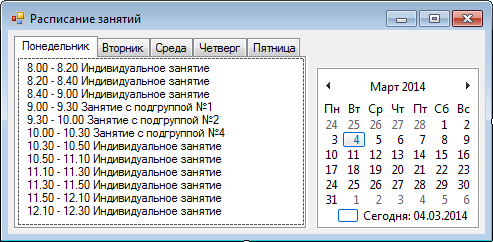


Рисунок 6 – Расписание занятий

На данной форме расположен инструмент TabPage с помощью которого можно посмотреть расписание своих занятий на тот день который вам необходим. Также для удобства пользования присутствует инструмент Календарь.

Форма «Индивидуальный план коррекционной работы и анализ результативности».

На рисунке 7 изображена форма, в которой содержатся планы работы учителя-логопеда, а также анализ результативности работы.

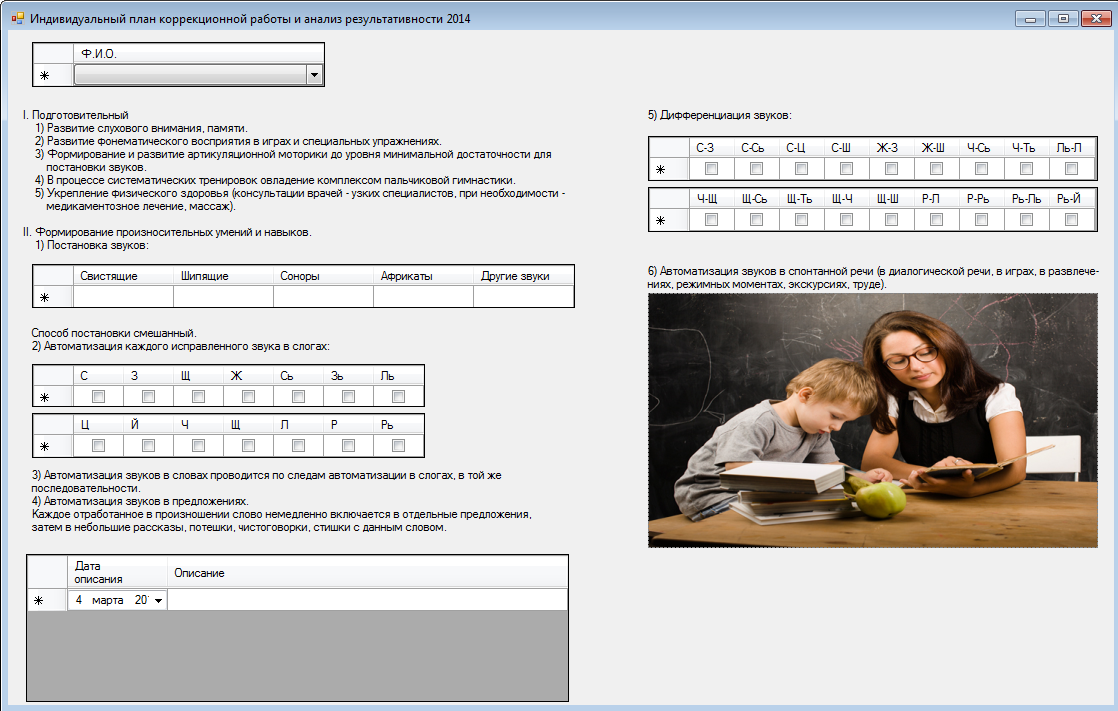


Рисунок 7 – Индивидуальный план коррекционной работы и анализ результативности

Ограничения на данной форме:

- Ф.И.О. – обязательное поле с выпадающим списком, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Свистящие – обязательное поле, длина не меньше 1 символа, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Шипящие – обязательное поле, длина не меньше 1 символа, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Соноры – обязательное поле, длина не меньше 1 символа, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Африкаты – обязательное поле, длина не меньше 1 символа, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Другие звуки – обязательное поле, длина не меньше 1 символа, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

На форме такжеприсутствуют 32 логических поля, которые могут быть либо включенными либо выключенными:

- С;-З; -Щ; -Ж; -Сь;-Зь;-Ль; -Ц;-Й;-Ч;-Щ;-Л;-Р;-Рь; -С-З;-С-Сь; -С-Ц; -С-Ш;-Ж-З;-Ж-Ш;-Ч-Сь;-Ч-Ть;-Ль-Л;-Ч-Щ;-Щ-Сь;-Щ-Ть;-Щ-Ч;-Щ-Ш;-Р-Л;-Р-Рь;-Рь-Ль;-Рь-Й;

- Дата описания – поле в формате ДД.ММ.ГГ реализованное с помощьюинструментавыборадаты;

- Описание – тектовое поле, длина не меньше 5 символов, не больше 100 символов, допустимысимволырусского и латинського алфавита в любомрегистре.

Также на форме присутствует текст документации, для облегчённой и комфортной работы.

Форма «Индивидуальная карта развития»

На рисунке 8 изображена форма, в которой содержится индивидуальная карта развития ребёнка.

Ограничения на данной форме:

- Ф.И.О. – обязательное поле с выпадающим списком, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Дата рождения – обязательное поле в формате ДД.ММ.ГГ.

- Диагноз – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

- Звукопроизношение – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

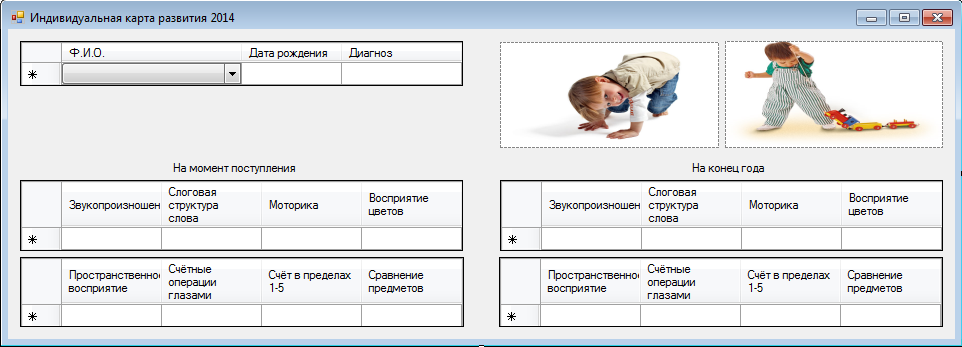


Рисунок 8 – Индивидуальная карта развития

- Слоговая структура слова – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Моторика – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Восприятие цветов – обязательное поле, длина не меньше 1 символа, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Пространственное восприятие – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Счётные операции глазами – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

- Счёт в пределах 1-5 – числовое поле, длина не больше 10 символов..

- Сравнение предметов – обязательное поле, длина не меньше 3 символов, не больше 50 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

На форме присутствуют 2 таблицы одна из которых отображает информацию о развитии ребёнка на момент поступления на логопедический пункт, и вторая таблица которая отображает ту же информацию но на конец года.

Форма «Карта речевого развития»

На рисунке 9 изображена форма в которой хранится документация речевого развития детей.

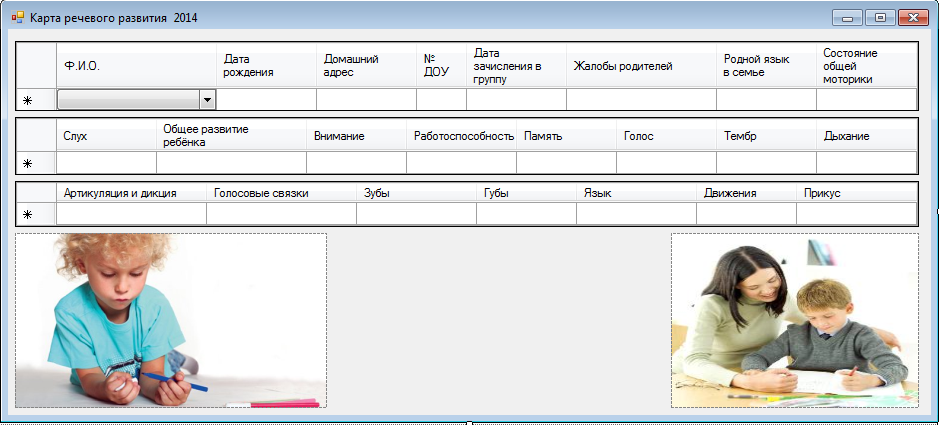


Рисунок 9 – Карта речевого развития

Ограничения на данной форме:

- Ф.И.О. – обязательное поле с выпадающим списком, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Дата рождения – обязательное поле в формате ДД.ММ.ГГ.

- Адрес – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

- № ДОУ – обязательное числовое поле, длина не меньше 1 символа, не больше 10 символов;

- Дата зачисления в группу – обязательное поле в формате ДД.ММ.ГГ.

- Жалобы родителей – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Родной язык в семье – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Состояние общей моторики – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

-Слух – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Общее развитие ребёнка – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Внимание – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Работоспособность – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Память – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Голос – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Тембр – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Дыхание – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

-Артикуляция и дикция – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Голосовые связки – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Зубы – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Язык – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Движения – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

- Прикус – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел.

Форма «Речевой экран»

На рисунке 10 изображена форма в которой хранится документация речевого экрана.

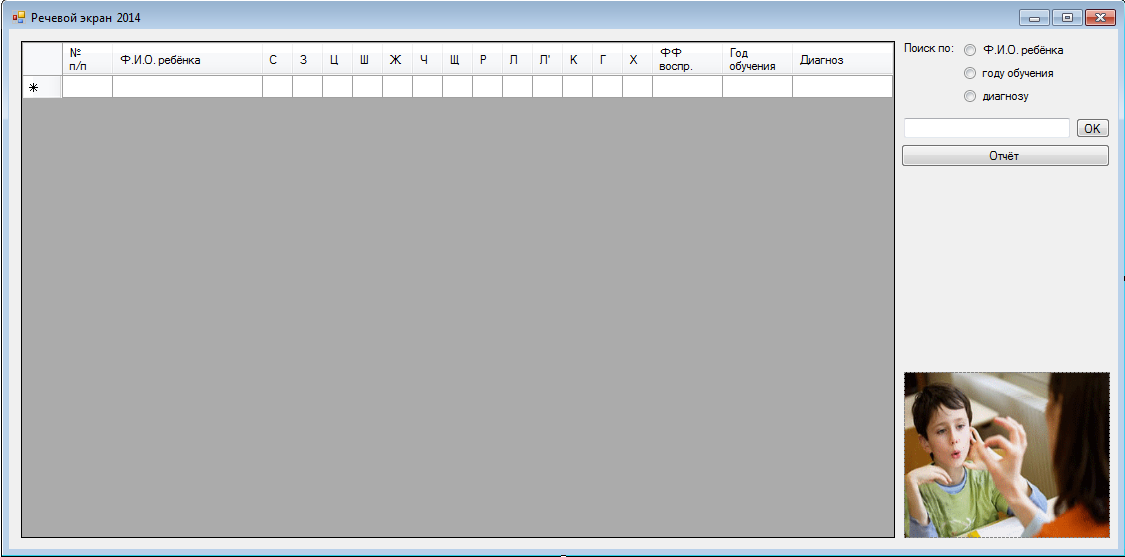


Рисунок 10 – Речевой экран

Ограничения на данной форме:

- № п/п – поле типа счётчик, автоматически нумерует поля по порядку.

- Ф.И.О. ребёнка – обязательное поле, длина не меньше 5 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре и пробел;

На форме также присутствуют 15 необязательных текстовых поля, длина которых не меньше 1 символа, не более 3 символов, в этих полях содержатся такие метки как «()» – звук полностью поставлен в речи, «\_» - начата работа над звуком, и «::» - означает что звук поставлен только в слогах:

- С; -З; -Ц; -Ш; -Ж; -Ч; -Щ; -Р; -Л; -Л’; -К; -Г; -Х;

-ФФ воспр. – обязательное поле, длина не меньше 3 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

- Год обучения – обязательное числовое поле, длина не меньше 1 символа, не больше 5.

- Диагноз – обязательное поле, длина не меньше 3 символов, не больше 255 символов, допустимы символы русского и латинского алфавита в любом регистре, а также цифры и пробел;

Также на форме присутствует поиск по таким направлениям: Ф.И.О. ребёнка, год обучения. Далее на форме отображается кнопка «Отчёт» при нажатии на которую генерируется отчёт по речевому экрану.

Форма «Отчёт “Речевой экран”»

На рисунке 11 изображена форма в которой хранится документация речевого экрана.

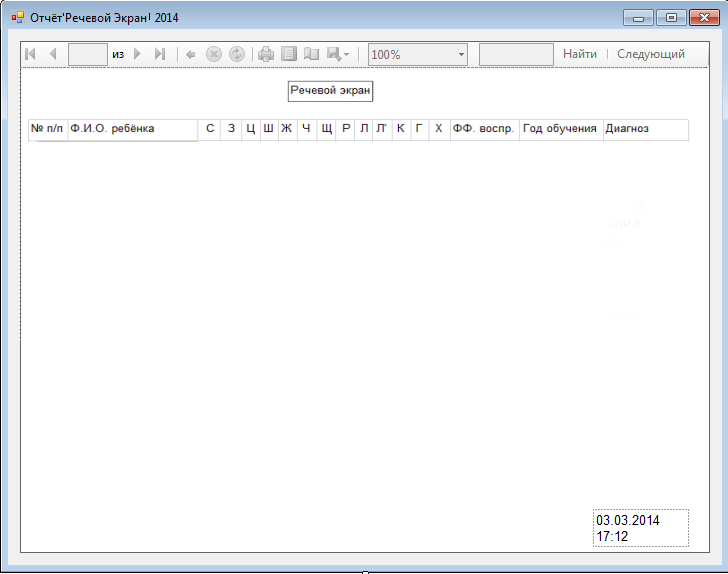


Рисунок 11 – Отчёт «Речевой экран»

На данной форме генерируется отчёт из предыдущей формы «Речевой экран», ниже отображается дата отчёта.

Форма «Картотека дидактических игр».

На рисунке 12 изображена форма в которой хранится документация речевого экрана.

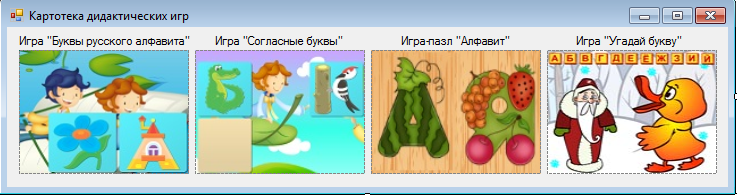


Рисунок 12 – Картотека дидактических игр

На форме присутствуют картинки, которые являются кнопками перехода на другие формы игр.

Форма «Игра “Буквы русского алфавита”»

На рисунке 13 изображена форма, в которой находится образовательная игра для детей с буквами русского алфавита.



Рисунок 13 – Игра «Буквы русского алфавита»

На форме присутствует Flash игра «Буквы русского алфавита», необходимо соединить точки (буквы) между собой в правильной последовательности.

Форма «Игра “Согласные буквы”»

На рисунке 14 изображена форма в которой находится образовательная игра для детей с согласными буквами русского алфавита.



Рисунок 14 – Игра «Согласные буквы»

На форме присутствует Flash игра «Согласные буквы», где необходимо найти пару каждой согласной букве.

Форма «Игра-пазл“Алфавит”»

На рисунке 15 изображена форма в которой находится образовательная игра для детей.



Рисунок 15 – Игра-пазл «Алфавит»

На форме присутствует Flashигра «Алфавит», где необходимо вставить букву на её место.

Форма «Игра“Угадай букву”»

На рисунке 16 изображена форма в которой находится образовательная игра для детей где необходимо отгадать букву.



Рисунок 16 – Игра «Угадай букву»

На форме присутствует Flash игра «Угадай букву», где необходимо смотря на картинку на экране отгадать на какую букву начинается данное слово.

1.2.3 Требования к надежности программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением пользователем совокупности организационно-технических мероприятий, перечень которых приведен ниже:

* организацией бесперебойного питания технических средств;
* использованием лицензионного программного обеспечения;   
  регулярным выполнением рекомендаций о типовых нормах времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств;
* регулярным выполнением требований по защите информации и испытаниях программных средств на наличие компьютерных вирусов.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств. Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

1.2.4 Требования к квалификации и численности персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2штатных единиц — учитель-логопед и системный администратор.

На должность системного администратора назначается лицо, отвечающее следующим требованиям: профессиональное образование, стаж работы в соответствующей области не менее года.

В перечень задач, выполняемых системным администратором, должно входить:

* поддержания работоспособности технических средств;
* установка и поддержание работоспособности системных программных средств – операционной системы;
* установка программы;
* сохранение работоспособности системы при некорректных операциях пользователя.

Учитель-логопед должен:

* знать все документации, с которыми он должен работать;
* быть опытным пользователем ПК.

Учитель-логопед выполняет следующие должностные обязанности:

* вести учёт документаций, которые будут храниться в базе данных ПМПК;
* анализирует информацию, полученную на основе работы с детьми, и вносит её в базу данных ПМПК через форму программы;
* формирует отчёты и отправляет их заведующему ПМПК.

1.2.5 Требования к технологии хранения и обработки информациипроектируемой задачи. Требования к системе управления базами данных

Требования к системе управления базами данных СУБД должна обладать возможностью работы в многопользовательском режиме, восстановления работоспособности при программно-аппаратных сбоях, таких как отключение электропитания, проблемы с аппаратным обеспечением, нештатное завершение работы.

С целью обеспечения надежного функционирования в СУБД должно быть предусмотрено:

* сохранение работоспособности системы при некорректных операциях пользователя (ввод некорректных данных);
* сохранение целостности данных при нештатном завершении работы системы;
* резервное копирование данных;
* журналирование операций системы.

Данным требованиям соответствуют такие СУБД, как MSSQL Server, PostgreSQL, MySql, SQLite.

1.2.6 Требования к составу и параметрам технических средств

Требования к серверу: процессор INTEL XEON, минимальный объем оперативной памяти 8Гб, минимальный объем дискового пространства 100гб, операционная системаWindows 2000 Server и выше.

Требования к клиентской машине: процессор IntelXeon 5110или выше, минимальный объём оперативной памяти 2Гб, минимальный объём дискового пространства на жестком диске 50Гб, операционная система Windows 7.

Для пользователя необходимо наличие установленного .NETFrameworkv4.0, и выходом в Интернет.

1.2.7 Требования к исходным кодам и языкам программирования

Разрабатываемая система должна предусматривать возможность ее дальнейшего развития, модификации и включения новых функций в систему, улучшение кода.

Разрабатываемая система должна представлять собой клиент-серверную архитектуру, база данных которой находится в сети Интернет.

Структура программы должна предусматривать возможность ее развития за счет разработки и включения в нее новых форм и функций.

В качестве языка программирования могут быть выбраны такие языки, как С++, С#, Delphi 7. Выбор других языков нецелесообразен.

1.2.8 Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной версией операционной системы Windows 2000 Server или выше.

1.2.9 Требования к организации входных и выходных данных

Организация входных и выходных данных должна соответствовать п.1.2.2.

В процессе работы программы входной информацией для программы должны являться: файлы баз данных, манипуляции мышью, а также информация, вводимая пользователем на клавиатуре ЭВМ, согласно режимам, определяемых выходной экранной информацией. Перечень допустимых клавиш представлен в п.1.2.2.

1.2.10 Требования к защите информации и программ

Вход в программу будет осуществляться путём ввода логина и пароля, чтобы не случилось несанкционированного доступа к данным программы.

1.2.11 Предварительный состав программной документации

Состав программной документации должен включать следующие доку­менты:

* техническое задание (постановку задачи);
* руководство пользователя, содержащее описание всех задокументированных возможностей программы;
* текст программы, содержащий исходный код на одном из языков, указанных в пункте 1.2.7;
* руководство программиста, содержащее данные о технологии проектирования и программирования данной программы, а также о языке программирования, СУБД, назначении программы;
* список литературы, содержащей теоретический материал, необходимый для создания программы.

1.3 Моделирование проектируемой задачи

1.3.1 Диаграмма вариантов использования

В данной подсистеме существуют три пользователя, которые непосредственно работают с программой, либо с базой данных программы, а именно:

* учитель-логопед;
* заведующая психолого-медико-педагогической консультации;
* системный администратор ПМПК.

Актёр «Учитель-логопед» это специалист, который занимается коррекцие йнарушений речи у детей, ведёт учёт всей документации (Карты речевого развития, планы работы, движение детей на ЛП, речевой экран, карты развития, картотеку игр, расписание занятий, список детей в ЛП, анализ результативности) и отправляет из программы результаты и отчёты заведующей ПМПК через факс.

Актёр «Заведующая ПМПК» это специалист, который координирует работу всей психолого-медико-педагогической консультации. Она следит за всеми документациями с разных логопедических пунктов через общую базу данных, в которой, собственно, и хранятся все выше перечисленные документы. Также она составляет график работы каждому учителю-логопеду в каждом логопедическом пункте.

Актёр «Системный администратор ПМПК» это специалист, который занимается установкой и настройкой программного обеспечения и оборудования для психолого-медико-педагогической консультации.

На рисунке 1 изображена диаграмма прецедентов для проектируемой задачи.



Рисунок 17 – Диаграмма прецедентов

1.3.2 Диаграмма деятельности

Разрабатываемая система характеризуется не только структурой составляющих ее элементов, но также и поведением (функциональностью). При моделировании поведения проектируемой системы возникает необходимость моделирования логической реализации выполняемых системой операций.

Диаграмма деятельности фокусируется на последовательности выполнения (потоке) и взаимосвязи действий (элементарных операций) в составе единого процесса, которые в совокупности приводят к получению желаемого результата.

В соответствии с вышеизложенными соображениями, диаграмма деятельности модели проектируемой информационной системы должна выглядеть, как это показано на рисунке 18.

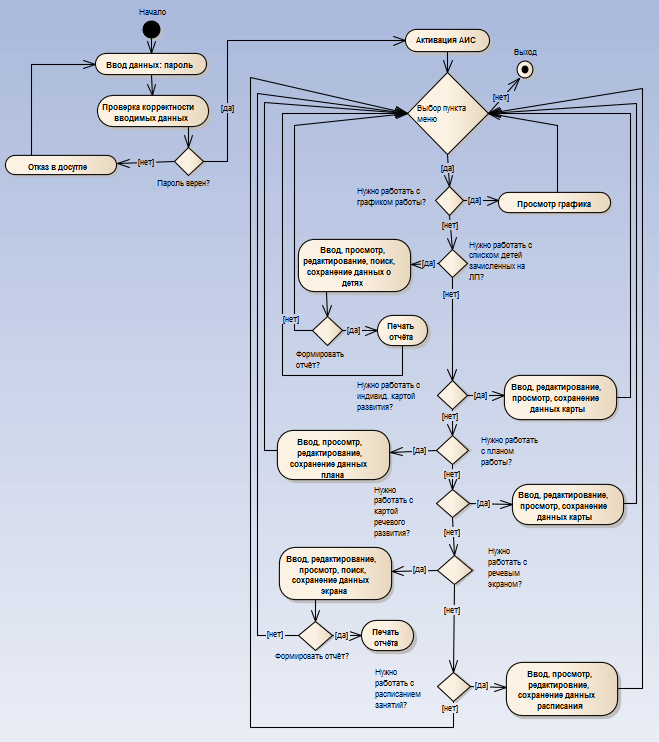


Рисунок 18 – Диаграмма деятельности АИС

1.3.3 Диаграмма базы данных

Основные задачи:

* обеспечение хранения в БД всей необходимой информации;
* обеспечение возможности получения данных по всем необходимым запросам;
* сокращение избыточности и дублирования данных;
* обеспечение [целостности базы данных](http://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D1%81%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

Основные этапы проектирования баз данных:

Концептуальное (инфологическое) проектирование — построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную [СУБД](http://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94) и [модель данных](http://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является как образом реальности, так и образом проектируемой базы данных для этой реальности.

Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные [ER-диаграммам](http://ru.wikipedia.org/wiki/ER-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85#.D0.9D.D0.BE.D1.82.D0.B0.D1.86.D0.B8.D0.B8).

Чаще всего концептуальная модель базы данных включает в себя:

описание информационных объектов, или понятий предметной области и связей между ними.

описание ограничений целостности, т.е. требований к допустимым значениям данных и к связям между ними.

На рисунке 19 изображена диаграмма БД проектируемой подсистемы.

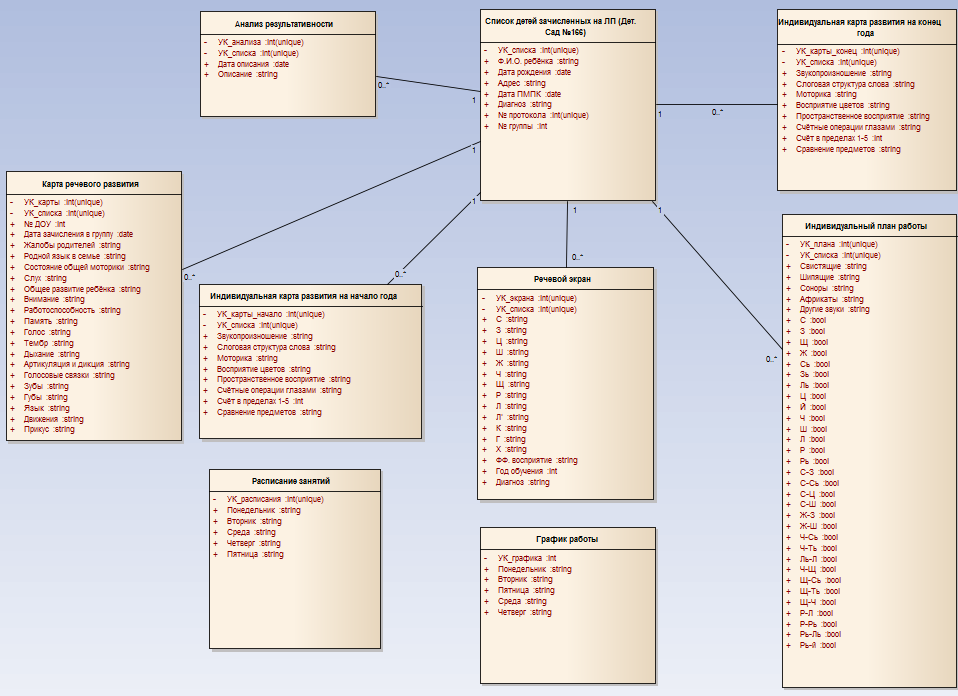


Рисунок 19 – Диаграмма БД проектируемой подсистемы

1.3.4 Диаграмма компонентов и развёртываний

Рассмотренные ранее UML-диаграммы отражают концептуальные аспекты построения модели информационно-справочной системы и относятся к логическому уровню представления. Элементы логического представления (например, классы и ассоциации) не существуют физически, а лишь отражают понимание структуры системы или аспекты ее поведения. Однако для создания реальной ИСС необходимо реализовать все элементы логического представления в виде реальных сущностей физического представления модели системы. В UML для этих целей имеется два вида диаграмм: диаграмма развёртывания и диаграмма компонентов.

Хотя диаграммы развёртывания и диаграммы компонентов можно изображать отдельно, также допускается помещать диаграмму компонентов на диаграмму развёртывания. Это целесообразно делать, чтобы показать какие компоненты выполняются и на каких узлах (рисунок 20).

Диаграмма компонентов показывает зависимости и взаимодействия между компонентами программного обеспечения. Диаграмма компонентов изображает физическую архитектуру компьютер базированной системы. Диаграмма компонентов разрабатывается для визуализации общей структуры исходного программного кода и спецификации сборки исполняемого программного кода системы. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами.

Компонент - это контейнер пакетов, участвующих в работе системы. Рабочие компоненты представлены пакетами исходного и исполняемого кода. Во многих средах программирования компонент соответствует модулю, а модуль – файлу. Для взаимодействия компонентов системы компоненты имеют порты (для организации взаимодействия распределённых компонентов) и реализуют некоторый набор интерфейсов.

В соответствии с вышеизложенными соображениями диаграмма рабочих компонентов модели проектируемой системы выглядит, как это показано на рисунке 20

1.4 Программное обеспечение

1.4.1 Обоснование выбора системы управления базами данных

Для разрабатываемой мною задачи используется клиент серверная архитектура, потому что с данной базой данных будут работать заведующая психолого-медико-педагогической консультации и учителя-логопеды данной консультации в разных логопедических пунктах.

Так как архитектура программного обеспечения для поставленной задачи будет клиент серверной, то в качестве СУБД мною была выбрана реляционная СУБД MySQL, поскольку эта СУБД быстрая, бесплатная и надёжная. [3]

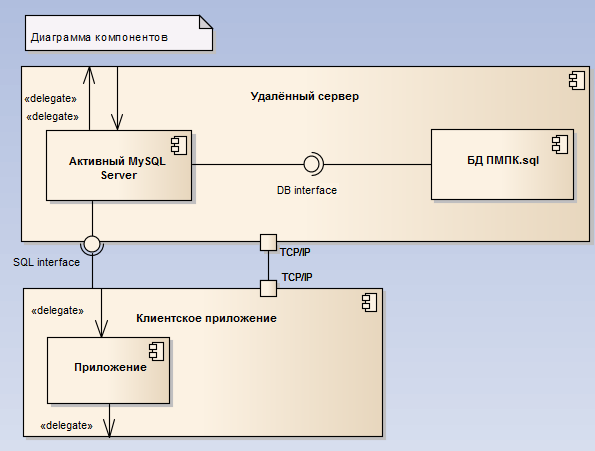


Рисунок 20 – Диаграмма компонентов

MySQL является собственностью компании Oracle Corporation, получившей её вместе с поглощённой Sun Microsystems, осуществляющей разработку и поддержку приложения. Распространяется под GNU General Public License или под собственной коммерческой лицензией. Помимо этого разработчики создают функциональность по заказу лицензионных пользователей, именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Несмотря на то что MySQL постоянно совершенствуется, он уже сегодня обеспечивает широкий спектр полезных функций. Благодаря своей доступности, скорости и безопасности MySQL очень хорошо подходит для доступа к базам данных по Internet.

Технические возможности СУБД MySQL. MySQL является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API).

Cтруктура MySQL трехуровневая: базы данных — таблицы — записи. Базы данных и таблицы MySQL физически представляются файлами с расширениями frm, MYD, MYI. Логически - таблица представляет собой совокупность записей. А записи - это совокупность полей разного типа. Имя базы данных MySQL уникально в пределах системы, а таблицы - в пределах базы данных, поля - в пределах таблицы. Один сервер MySQL может поддерживать сразу несколько баз данных, доступ к которым может разграничиваться логином и паролем. Зная эти логин и пароль, можно работать с конкретной базой данных. Например, можно создать или удалить в ней таблицу, добавить записи и т. д. Обычно имя-идентификатор и пароль назначаются хостинг провайдерами, которые и обеспечивают поддержку MySQL для своих пользователей.

1.4.2 Обоснование выбора языка программирования

В качестве языка программирования для реализации поставленной задачи был выбран С++, так как C++ — чрезвычайно мощный язык, содержащий средства создания эффективных программ практически любого назначения, от низкоуровневых утилит и драйверов до сложных программных комплексов самого различного назначения. [1]

C++ — [компилируемый](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) [статически типизированный](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) [язык программирования](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) общего назначения.

Поддерживает такие [парадигмы программирования](http://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D1%8B_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование, обеспечивает модульность, раздельную компиляцию, обработку исключений, абстракцию данных, объявление типов (классов) объектов, виртуальные функции. Стандартная библиотека включает, в том числе, общеупотребительные контейнеры и алгоритмы. C++ сочетает свойства как [высокоуровневых](http://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), так и [низкоуровневых языков](http://ru.wikipedia.org/wiki/%D0%9D%D0%B8%D0%B7%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). В сравнении с его предшественником — языком [C](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), — наибольшее внимание уделено поддержке [объектно-ориентированного](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [обобщённого программирования](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5). [2]

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание [операционных систем](http://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), разнообразных прикладных программ, [драйверов](http://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%B0%D0%B9%D0%B2%D0%B5%D1%80) устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений (игр). Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ. Например, наплатформе x86 это [GCC](http://ru.wikipedia.org/wiki/GNU_Compiler_Collection), [Visual C++](http://ru.wikipedia.org/wiki/Visual_C%2B%2B), [Intel C++ Compiler](http://ru.wikipedia.org/wiki/Intel_C%2B%2B_Compiler), [Embarcadero (Borland) C++ Builder](http://ru.wikipedia.org/wiki/Embarcadero_C%2B%2B_Builder) идругие. C++ оказал огромное влияние на другие языки программирования, в первую очередь на [Java](http://ru.wikipedia.org/wiki/Java) и [C#](http://ru.wikipedia.org/wiki/C_Sharp). [4]

Синтаксис C++ унаследован от языка [C](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)). Одним из принципов разработки было сохранение совместимости с C. Тем не менее, C++ не является в строгом смысле надмножеством C; множество программ, которые могут одинаково успешно транслироваться как [компиляторами](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) C, так и компиляторами C++, довольно велико, но не включает все возможные программы на C. C++ содержит средства разработки программ контролируемой эффективности для широкого спектра задач, от низкоуровневых утилит и драйверов до весьма сложных программных комплексов. В частности:

* высокая совместимость с языком Си : код на Си может быть с минимальными переделками скомпилирован компилятором C++. Внешнеязыковой интерфейс является прозрачным, так что библиотеки на Си могут вызываться из C++ без дополнительных затрат, и более того — при определённых ограничениях код на С++ может экспортироваться внешне не отличимо от кода на Си (конструкция extern "C"); [5]
* как следствие предыдущего пункта — вычислительная производительность. Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы. Один из базовых принципов С++ — «не платишь за то, что не используешь» (см. Философия C++) — то есть ни одна из языковых возможностей, приводящая к дополнительным накладным расходам, не является обязательной для использования. Имеется возможность работы с памятью на низком уровне; [6]
* поддержка различных стилей программирования: традиционное императивное программирование (структурное, объектно-ориентированное), обобщённое программирование, функциональное программирование, порождающее метапрограммирование;
* автоматический вызов деструкторов объектов в адекватном порядке (обратном вызову конструкторов) упрощает и повышает надёжность управления памятью и другими ресурсами (открытыми файлами, сетевыми соединениями, соединениями с базами данных и т.п.);
* перегрузка операторов позволяет кратко и ёмко записывать выражения над пользовательскими типами в естественной алгебраической форме;
* имеется возможность управления константностью объектов (модификаторы const, mutable, volatile). Использование константных объектов повышает надёжность и служит подсказкой для оптимизации. Перегрузка функций-членов по признаку константности позволяет определять выбор метода в зависимости цели вызова (константный для чтения, неконстантный для изменения). Объявление mutable позволяет сохранять логическую константность при виде извне кода, использующего кэши и ленивые вычисления.
* шаблоны C++ дают возможность построения обобщённых контейнеров и алгоритмов для разных типов данных. Попутно шаблоны дают возможность производить вычисления на этапе компиляции;
* возможность расширения языка для поддержки парадигм, которые не поддерживаются компиляторами напрямую. Например, библиотека Boost.Bind позволяет связывать аргументы функций. Используя шаблоны и множественное наследование, можно имитировать классы-примеси и комбинаторную параметризацию библиотек. Такой подход применён в библиотеке Loki, класс SmartPtr которой позволяет, управляя всего несколькими параметрами времени компиляции, сгенерировать около 300 видов «умных указателей» для управления ресурсами;
* возможность встраивания предметно-ориентирванных языков программирования в основной код. Такой подход использует, например библиотека Boost.Spirit, позволяющая задавать EBNF-грамматику парсеров прямо в коде C++. Boost.Spirit реализует рекурсивно-нисходящий алгоритм, что накладывает соответствующие ограничения (такие как недопустимость левой рекурсии);
* доступность. Для С++ существует огромное количество учебной литературы, переведённой на всевозможные языки. Язык имеет низкий порог вхождения, но среди всех языков такого рода обладает наиболее широкими возможностями.

1.4.3 Обоснование выбора инструментальных возможностей программной реализации задачи

Для реализации дипломного проекта мною была выбрана среда разработки Embarcadero RAD Studio, так как она наиболее удобна при создании простого и понятного интерфейса для автоматизированной информационной системы. А также Embarcadero RAD Studio позволяет быстро создавать безопасные подключаемые к базам данных приложения, способные обеспечить широчайшие возможности для работы пользователей.

Embarcadero RAD Studio — среда быстрой разработки приложений (RAD) для Microsoft Windows фирмы Embarcadero Technologies. RAD Studio дает возможность разрабатывать приложения, скомпилированные и оптимизированные для различных аппаратных платформ. Использование единого кода и следование единому проектному графику упрощают рабочий процесс и позволяют создавать приложения для разных платформ с помощью единого набора ресурсов для разработки. Средства разработки полноценных нативных приложений позволяют отказаться от применения скриптов и виртуальных машин, достичь максимальной производительности и безопасности, а также полностью раскрыть потенциал устройств, обеспечивая максимальное удобство работы пользователей. RADStudio содержит в себе набор средств для разработки приложений, но самое главное для меня это Embarcadero C++ Builder, — это среда C++, которая полностью соответствует концепции быстрой разработки приложений, объединяет средства ANSI C++ и многофункциональную расширяемую инфраструктуру визуальных компонентов.

1.4.4 Инструкция пользователя

Пользователю для использования программы необходимо установить компонент ActiveXShockwaveFlash. После того как программа установится, необходимо её запустить. Первое что мы увидим – окно авторизации. На рисунке 21 показано окно авторизации.

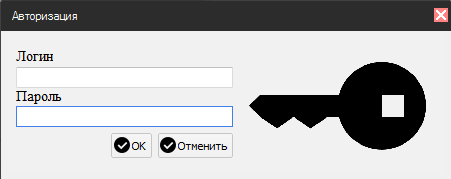


Рисунок 21 – Окно авторизации

На данной форме необходимо ввести логин и пароль, который сообщает системный администратор.Если данные неверны, то мы увидим окно «Ошибка авторизации», которое показано на рисунке 22.

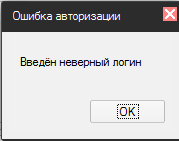


Рисунок 22 – Окно «Ошибка авторизации»

Если же логин и пароль верны, то мы увидим окно, уведомляющее о том, что проверка пройдена успешно. Данное окно показано на рисунке 23.

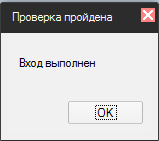


Рисунок 23 – Окно «Проверка пройдена»

После авторизация мы видим главное меню программы, показанное на рисунке 24.



Рисунок 24 – Главное меню

Находясь в главном меню программы, мы видим 8 изображений, которые являются средством перехода между формами, для этого необходимо дважды кликнуть по одной из картинок. На рисунке 25 рассмотрим форму, в которой содержится список зачисленных детей.

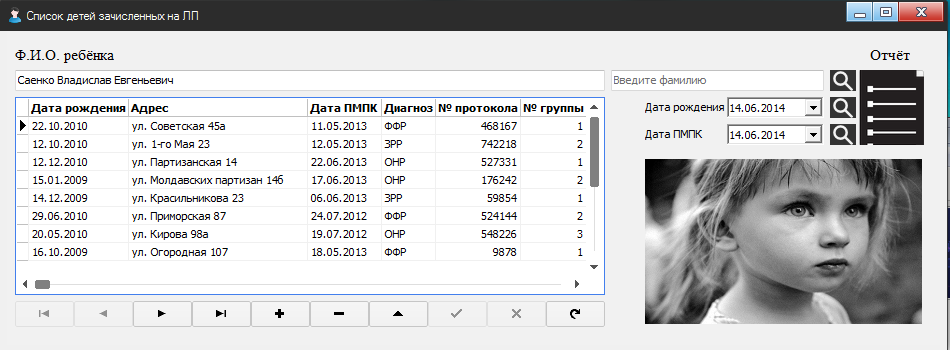


Рисунок 25 – Список детей зачисленных на ЛП

На данной форме мы видим таблицу с данными, а также навигатор для удобной работы с таблицей. Нажимая на кнопки со стрелочками, мы перемещаемся по таблице, и например, нажав на кнопку «+», мы начнём добавление записи, это действие показано на рисунке 26.

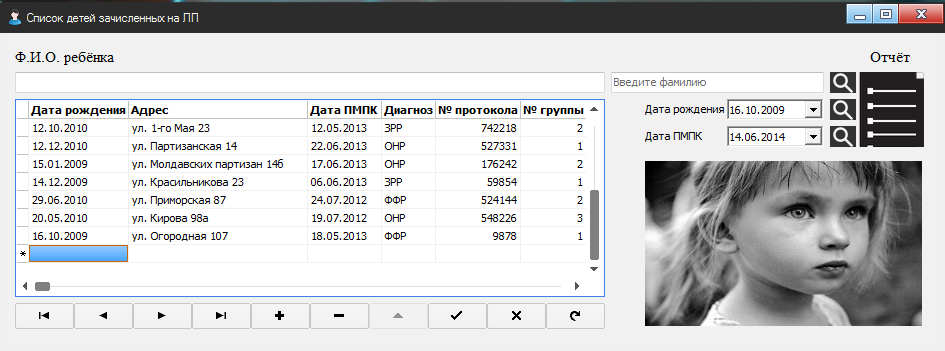


Рисунок 26 – Добавление записи в таблицу «Список зачисленных детей»

Также, присутствует поле для поиска фамилии, для этого необходимо лишь ввести фамилию ребёнка (не обязательно полностью), нажать на кнопку поиска правее и курсор таблицы переместится на нужную пользователю запись. Осуществление поиска продемонстрировано на рисунке 27.

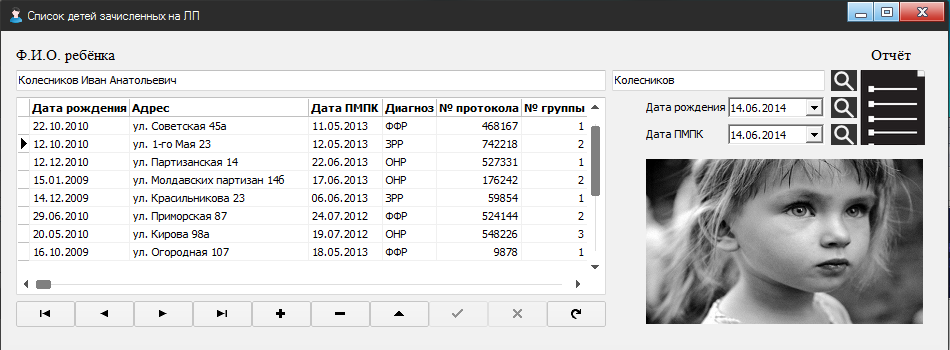


Рисунок 27 – Поиск по фамилии на форме «Список зачисленных детей»

Ниже поиска по фамилии находятся два поиска по датам, а именно по дате рождения и дате ПМПК, выбрав нужную дату через календарь и нажав на кнопку поиска правее, курсор переместится на необходимую запись. Это действие показано на рисунке 28.

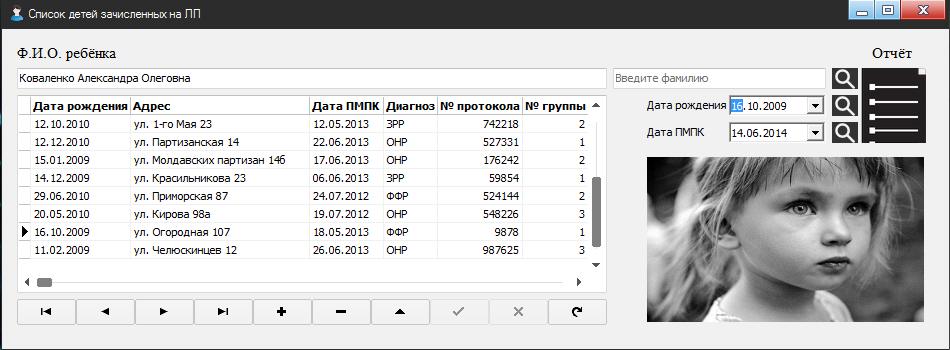


Рисунок 28 – Поиск по дате на форме «Список зачисленных детей»

В правом верхнем углу находится кнопка, которая генерирует отчёт. При нажатии на неё, откроется окно, где нам предложат, что именно мы хотим сделать с отчётом. Данное окно показано на рисунке 29.

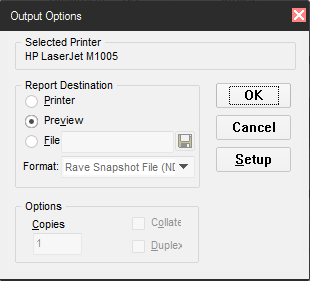


Рисунок 29 – Окно «OutputOptions»

В этом окне мы видим, какой принтер выбран для печати. Ниже находятся варианты действий с отчётом, а именно: распечатать, просмотреть и сохранить в файл. Выбрав вариант Preview и нажав на кнопку ОК, нам откроется непосредственно сам отчёт. Окно с отчётом показано на рисунке 30.

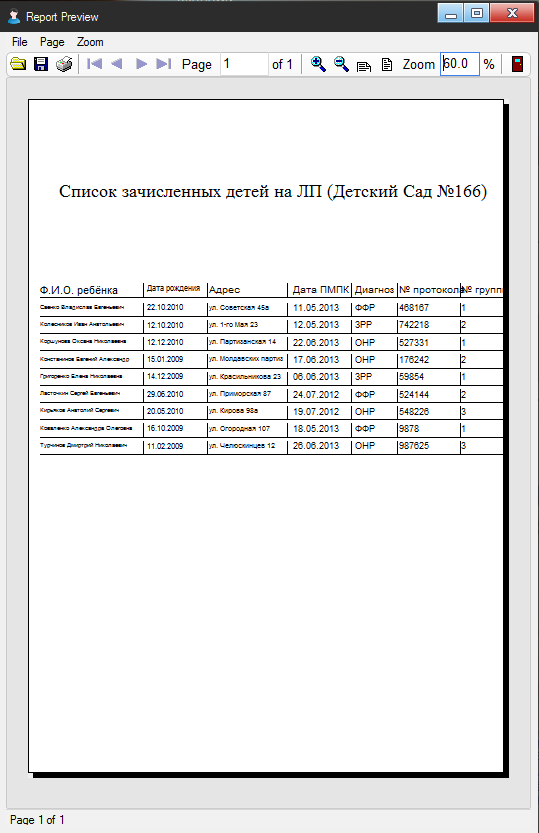


Рисунок 30 – Отчёт с формы «Список зачисленных детей»

Также в верхней панели инструментов присутствуют иконки для того чтобы открыть файл, сохранить и распечатать. При нажатии иконки с изображением принтера, откроется окно, в котором можно детальней настроить дальнейшую печать. Окно настройки печати можно увидеть на рисунке 31.

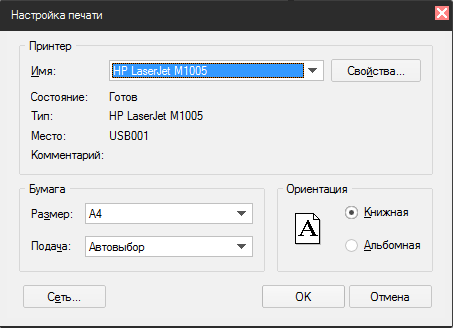


Рисунок 31 – Настройка печати

В данном окне можно выбрать принтер или факс, который вам необходим. Правее находится кнопка «Свойства», для настройки принтера. Можно выбрать размер бумаги, подачу и ориентацию печати. В левом нижнем углу присутствует кнопка «Сеть», для настройки сетевых принтеров или факсов. Окно «Настройка печати» также можно вызвать из предыдущего окна «OutputOptions»нажав по кнопке «Setup». Вернувшись в главное меню, рассмотрим следующую форму, которая называется «График работы», она показана на рисунке 32.

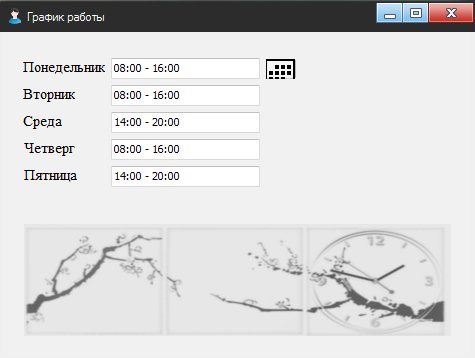


Рисунок 32 – Форма «График работы»

На данной форме находятся 5 полей из таблицы «График работы». Редактирование данных здесь невозможно, т.к. график работы составляет заведующая ПМПК. Правее мы видим кнопку, при нажатии на которую отобразится календарь, это необходимо для удобства пользователя. Календарь можно увидеть на рисунке 33.

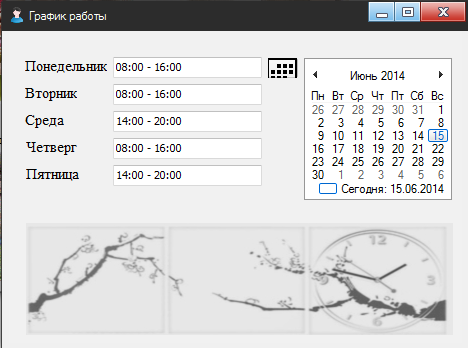


Рисунок 33 – Календарь на форме «Графикработы»

Вернувшись в главное меню, мы перейдём на следующую форму «Расписание занятий» которая показана на рисунке 34.

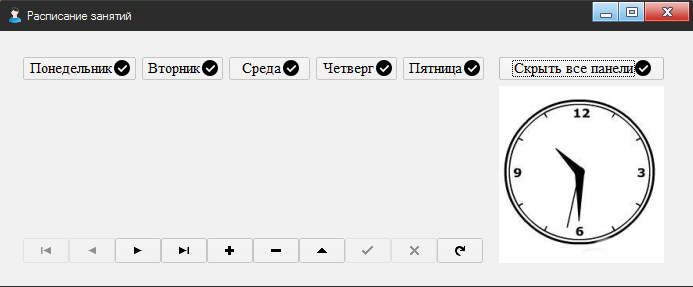


Рисунок 34 – Форма «Расписание занятий»

На форме присутствуют 5 кнопок, которые отвечают за соответствующий день недели. Например, нажав на кнопку «Понедельник» пользователь увидит поле, в котором будет храниться расписание занятий в понедельник. Это можно увидеть на рисунке 35.

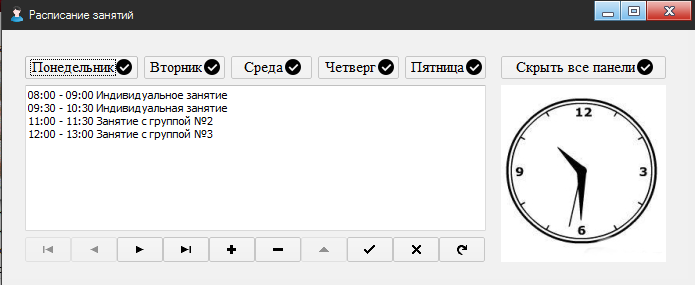


Рисунок 35 – Расписание на понедельник в форме «Расписание занятий»

Также присутствует навигатор для более удобного редактированияданных. В правом верхнем углу находится кнопка «Скрыть все панели» которая скрывает все панели расписания, при её нажатии все открытые панели будут скрыты. Вернёмся в главное меню и откроем следующую форму «Планы работы», она показана на рисунке 36.

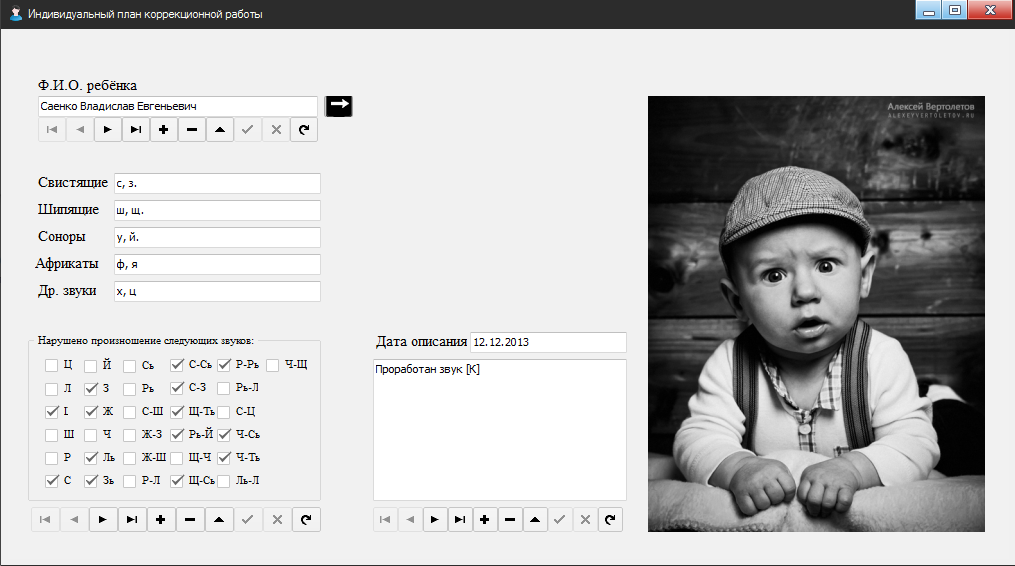


Рисунок 36 – Индивидуальный план коррекционной работы

На данной форме присутствует поле ФИО, гдене обходимо выбратьребёнка. Выбратьегоможно через навигатор, который находится ниже поля, либо нажав на стрілочку правее. Если пользователь нажмёт на стрілочку указывающую вправо, то ему откроется окно с детьми, которое показано на рисунке 37.

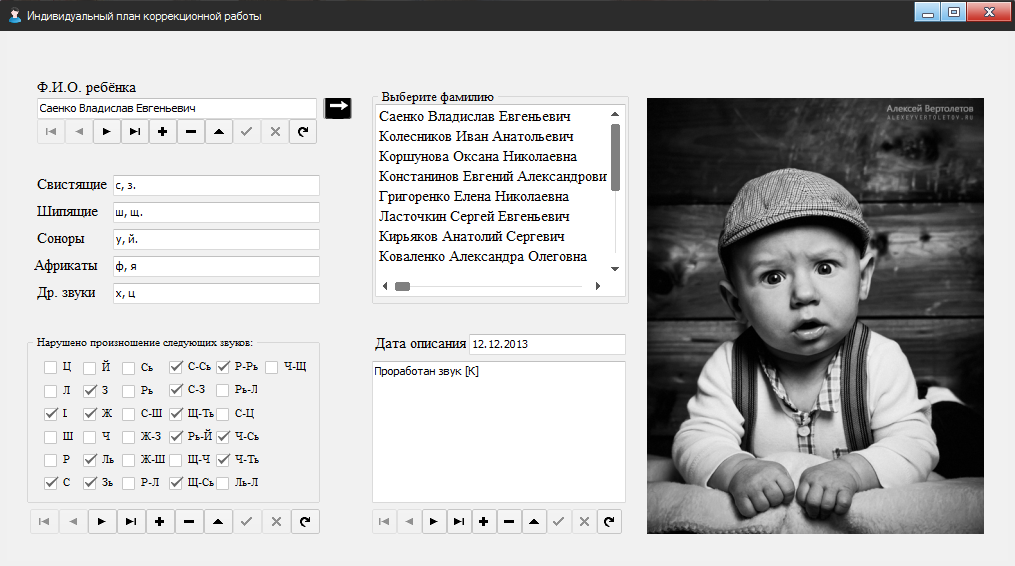


Рисунок 37 – План работы с включенням окном выбора детей

Ниже поля ФИО у нас находятся 5 текстовых и 32 логических поля из таблицы «План работы», а правее находятся поля из таблицы «Анализ результативности». Под каждой таблицей расположен навигатор для удобства пользователя. Вернувшись в главное меню, выберем следующую форму «Индивидуальные карты развития». Она показана на рисунке 38.

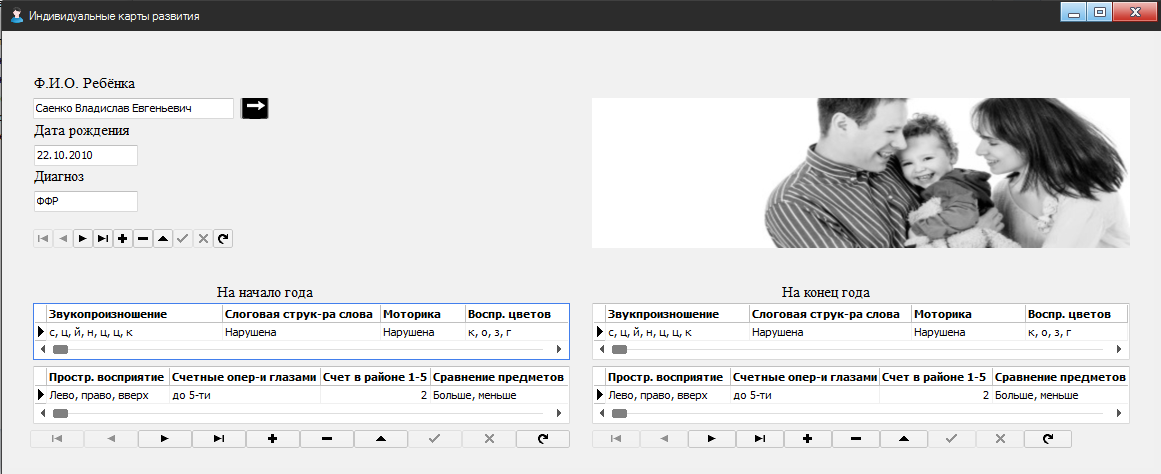


Рисунок 38 – Индивидуальные карты развития

Аналогично предыдущей форме, на этой тоже расположено поле ФИО из таблицы «Список зачисленнях детей», а также два дополнительных поля из той же таблицы. Присутствует кнопка со стрелочкой для просмотра и выбора всех детей. Ниже расположены две таблицы, «Индивидуальная карта развития на начало года», и «Индивидуальная карта развития на конец года». В начале ученого года вносится изначальная информация, а в конце, то что было дополнительно наработано с ребёнком за год. Ниже всех таблиц находяться навигаторы для удобства пользователя. Вернувшись в главное меню, мы перейдём на следующую форму, которая называется «Карты речового развития». Она показана на рисунке 39.

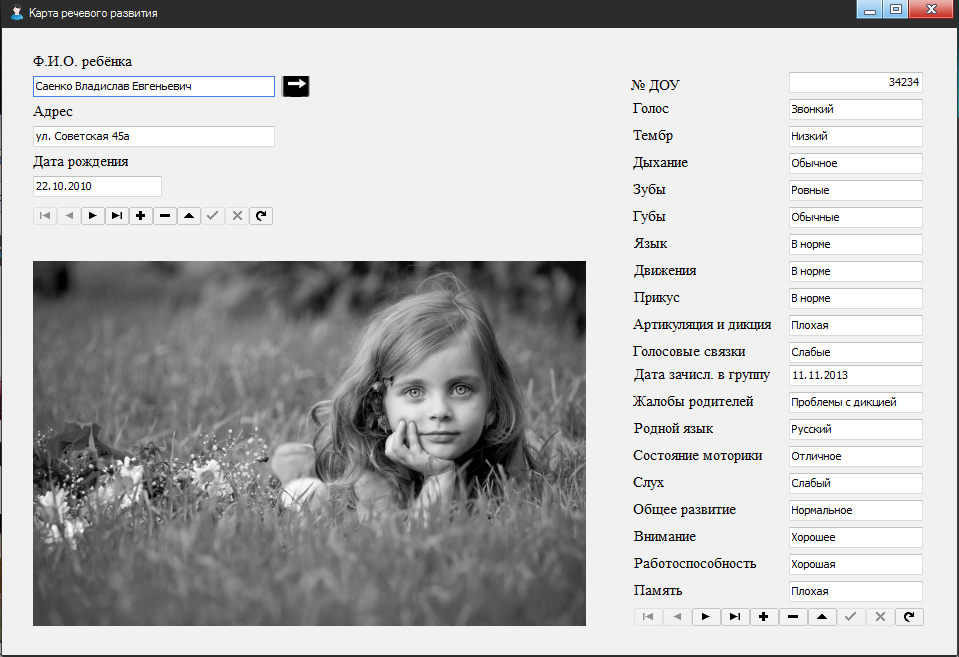


Рисунок 39 – Карта речового развития

На данной форме, аналогічно предыдущей, расположено поле ФИО с таблицы «Список зачисленнях детей», а также два дополнительных поля с той же таблицы. Присутствует кнопка со стрелочкой для просмотра и выбора всех детей. В правой части формы расположены 20 полей из таблицы «Карта речового развития». Ниже всех таблиц расположены навигаторы, чтобы пользователю было удобней работать с формой. Вернувшись в главное меню, выберем следующую форму с названием «Речевой экран» которая показана на рисунке 40.

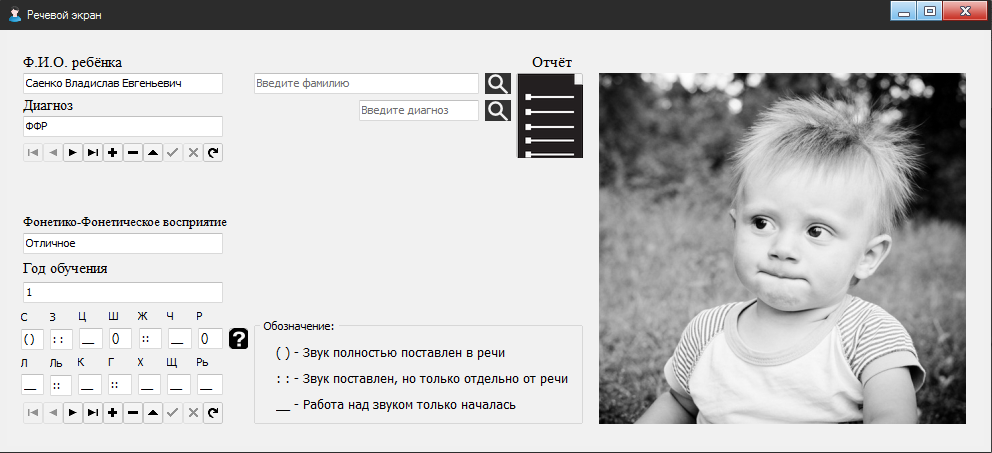


Рисунок 40 – Речевой экран

На данной форме, аналогічно предыдущей, мы видим поле ФИО из таблицы «Список зачисленнях детей», а также поле Диагноз из той же таблицы. Правее мы видим поле для поиска фамилии, при вводе необходимой фамилии (не обязательно полностью) и нажатии на кнопку поиска, программа найдёт в таблице не обходи мого ребёнка. Поиск показан на рисунке 41.

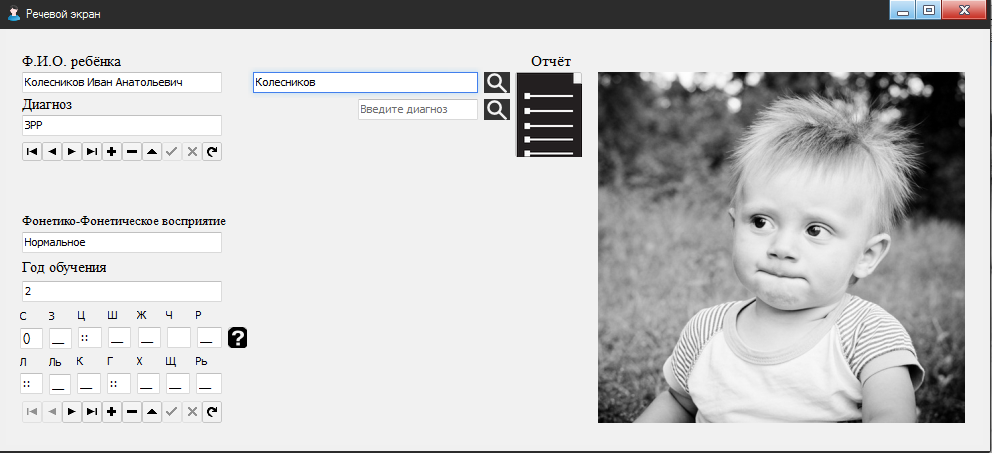


Рисунок 41 – Поиск по фамилии на форме «Речевой экран»

Также ниже данного поиска расположена ещё одно поле, предназначенное для поиска по диагнозу. В нижней части формы расположены 16 полей из таблицы «Речевой экран». Правее этих полей расположена кнопка помощи, обозначающая, что значат данные символы. Пояснения показаны на рисунке 42.

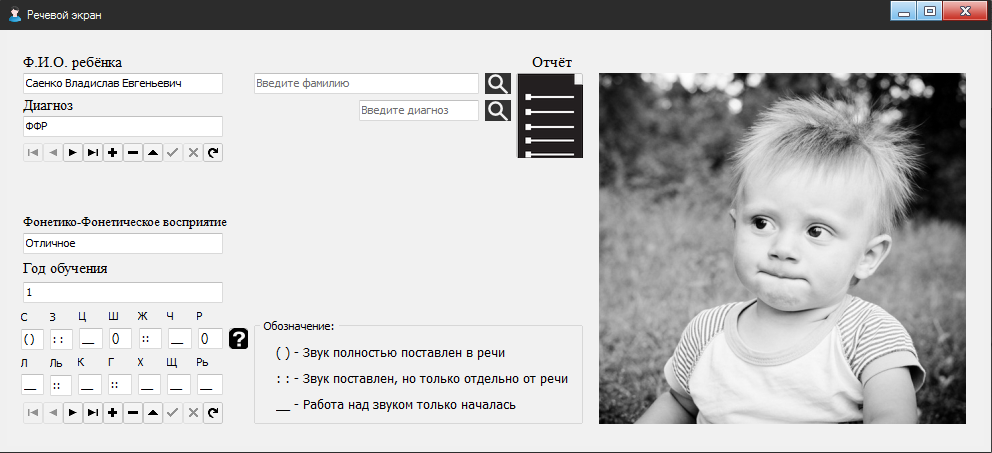


Рисунок 42 – Обозначение символов на форме «Речевой экран»

Как мы видим что символ «( )» обозначает что звук полностью поставлен в речи, «: :» - что звук постав лен лиш отдельно, а «\_\_» - работа над звуком только началась. Если одно из полей (звуков) пустое, это означает что работа над звуком ещё не начиналась. В правой верхней части формы мы видим кнопку отчета, при нажатии на которую мы видим уже известное пользователю окно «Output options» где нам предлагают дальнейшие действия с отчётом. Сам отчёт речевого экрана показан на рисунке 43.

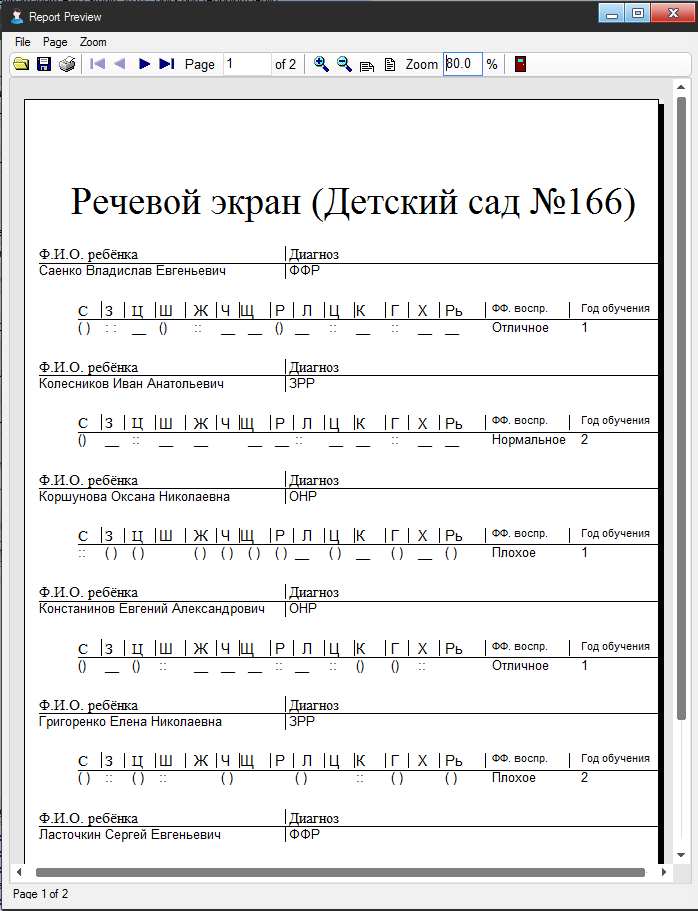


Рисунок 43 – Отчёт речового экрана

Аналогично предыдущему отчету, здесь мы тоже можем распечатать, сохранить или открыть отчет. Вернувшись в главное меню выберем следующую форму «Картотека игр» которая показана на рисунке 44.

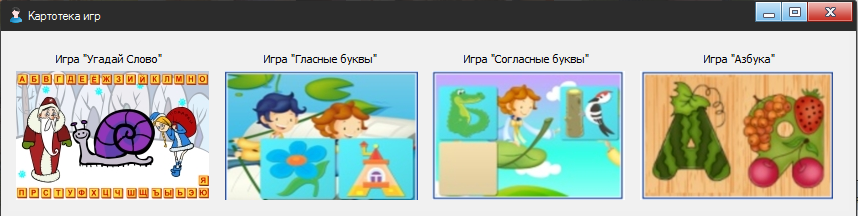


Рисунок 44 – Картотека игр

На форме присутствуют 4 изображения которые являются средством перехода между формами. Нажав изображение с названим Игра «Угадай Слово» мы перейдём на соответствующую форму которая показана на рисунке 45.



Рисунок 45 – Игра «Угадай слово»

На данной форме расположена Flash игра «Угадай слово», где необходимо нажать первую букву того животного что изображено на экране. Вернувшись на форму «Картотека игр» мы выберем следующую форму, которая называется Игра «Гласные буквы». Она показана на рисунке 46.



Рисунок 46 – Игра «Гласные буквы»

На данной форме расположена Flash игра «Гласные буквы», где необходимо в алфавитном порядке соединить буквы, чтобы получить изображение. Вернувшись на форму «Картотека игр» мы выберем следующую форму, которая называется Игра «Согласные буквы». Она показана на рисунке 47.



Рисунок 47 – Игра «Согласные буквы»

На данной форме расположена Flash игра «Согласные буквы», где необходимо найти каждой букве пару. Вернувшись на форму «Картотека игр» мы выберем следующую форму, которая называется Игра «Азбука». Она показана на рисунке 48.

.

Рисунок 48 – Игра «Азбука»

На данной форме расположена Flash игра «Азбука», где необходимо перетащить цветную букву на её место.

2 Производственная и экологическая безопасность при разработке автоматизированной информационной подсистемы учёта работы учителя-логопеда Мариупольской психолого-медико-педагогической консультации при дошкольном заведении №166

2.1.1 Факторы, определяющие условия труда пользователя работающего с персональным компьютером

К работам оператором, программистом, инженером и техником ПК, пользователем ПК допускаются:

* прошедшие вводный инструктаж по охране труда;
* прошедшие обучение безопасным приемам и методам труда по программе, утвержденной руководителем предприятия, разработанной на основе Типовой программы, и прошедшие проверку знаний, в том числе по электробезопасности с присвоением 1-й квалификационной группы по электробезопасности;
* прошедшие курс обучения принципам работы с вычислительной техникой, а также по работе на персональном компьютере на конкретном программном обеспечении;
* инструктаж по охране труда на конкретном рабочем месте по данной инструкции.

Помещение с установленной вычислительной техникой перед началом и после окончания работы, должно быть проветрено, что обеспечивает улучшение качественного состава воздуха, в том числе и аэроионный режим.

Влажная уборка кабинета с вычислительной техникой должна производиться обязательно один раз в день, а при наличии больше пяти единиц – два раза.

С целью нормализации содержания аэроионов в воздухе рабочей зоны рекомендуется устанавливать в помещениях с ПК ионизаторы воздуха.

Предельно допустимое значение вредного производственного фактора - это предельное значение величины вредного производственного фактора, воздействие которого при ежедневной регламентированной продолжительности в течение всего трудового стажа не приводит к снижению работоспособности и заболеванию, как в период трудовой деятельности, так и к заболевания в последующий период жизни, а также не оказывает неблагоприятного влияния на здоровье потомства.

При пользовании средствами вычислительной техники и периферийным оборудованием каждый работник должен внимательно и осторожно обращаться с электропроводкой, приборами и аппаратами и всегда помнить, что пренебрежение правилами безопасности угрожает и здоровью, и жизни человека

Во избежание поражения электрическим током необходимо твердо знать и выполнять следующие правила безопасного пользования электроэнергией:

- необходимо постоянно следить на своем рабочем месте за исправным состоянием электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, и заземления. При обнаружении неисправности немедленно обесточить электрооборудование, оповестить администрацию. Продолжение работы возможно только после устранения неисправности.

- во избежание повреждения изоляции проводов и возникновения коротких замыканий не разрешается:

- вешать что-либо на провода;

- закрашивать и белить шнуры и провода;

- закладывать провода и шнуры за газовые и водопроводные трубы, за батареи отопительной системы;

- выдергивать штепсельную вилку из розетки за шнур, усилие должно быть приложено к корпусу вилки.

-для исключения поражения электрическим током запрещается:

- часто включать и выключать компьютер без необходимости;

- прикасаться к экрану и к тыльной стороне блоков компьютера;

- работать на средствах вычислительной техники и периферийном оборудовании мокрыми руками;

- работать на средствах вычислительной техники и периферийном оборудовании, имеющих нарушения целостности корпуса, нарушения изоляции проводов, неисправную индикацию включения питания, с признаками электрического напряжения на корпусе

- класть на средства вычислительной техники и периферийном оборудовании посторонние предметы.

-запрещается под напряжением очищать от пыли и загрязнения электрооборудование.

-запрещается проверять работоспособность электрооборудования в неприспособленных для эксплуатации помещениях с токопроводящими полами, сырых, не позволяющих заземлить доступные металлические части.

- ремонт электроаппаратуры производится только специалистами-техниками с соблюдением необходимых технических требований.

- недопустимо под напряжением проводить ремонт средств вычислительной техники и периферийного оборудования.

- во избежание поражения электрическим током, при пользовании электроприборами нельзя касаться одновременно каких-либо трубопроводов, батарей отопления, металлических конструкций , соединенных с землей.

- при пользовании электроэнергии в сырых помещениях соблюдать особую осторожность.

- при обнаружении оборвавшегося провода необходимо немедленно сообщить об этом администрации, принять меры по исключению контакта с ним людей. Прикосновение к проводу опасно для жизни.

-спасение пострадавшего при поражении электрическим током главным образом зависит от быстроты освобождения его от действия током.

Во всех случаях поражения человека электрическим током немедленно вызывают врача. До прибытия врача нужно, не теряя времени, приступить к оказанию первой помощи пострадавшему.

Необходимо немедленно начать производить искусственное дыхание, наиболее эффективным из которых является метод рот в рот или рот в нос, а также наружный массаж сердца.

Искусственное дыхание пораженному электрическим током производится вплоть до прибытия врача.

Требования по обеспечению пожарной безопасности.На рабочем месте запрещается иметь огнеопасные вещества

В помещениях запрещается:

- зажигать огонь;

- включать электрооборудование, если в помещении пахнет газом;

- курить;

- сушить что-либо на отопительных приборах;

- закрывать вентиляционные отверстия в электроаппаратуре

Источниками воспламенения являются:

- искра при разряде статического электричества

- искры от электрооборудования

- искры от удара и трения

- открытое пламя

При возникновении пожароопасной ситуации или пожара персонал должен немедленно принять необходимые меры для его ликвидации, одновременно оповестить о пожаре администрацию.

Помещения с электрооборудованием должны быть оснащены огнетушителями типа ОУ-2 или ОУБ-3.

Общие требования безопасности - Настоящая инструкция распространяется на персонал, эксплуатирующий средства вычислительной техники и периферийное оборудование. Инструкция содержит общие указания по безопасному применению электрооборудования в учреждении. Требования настоящей инструкции являются обязательными, отступления от нее не допускаются. К самостоятельной эксплуатации электроаппаратуры допускается только специально обученный персонал не моложе 18 лет, пригодный по состоянию здоровья и квалификации к выполнению указанных работ.

Требования безопасности перед началом работы - Перед началом работы следует убедиться в исправности электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, наличии заземления компьютера, его работоспособности.

Требования безопасности во время работы - Для снижения или предотвращения влияния опасных и вредных факторов необходимо соблюдать санитарные правила и нормы. гигиенические требования к видео дисплейным терминалам, персональным электронно-вычислительным машинам и организации работы, и Приложение 1,2

Во избежание повреждения изоляции проводов и возникновения коротких замыканий не разрешается: вешать что-либо на провода, закрашивать и белить шнуры и провода, закладывать провода и шнуры за газовые и водопроводные трубы, за батареи отопительной системы, выдергивать штепсельную вилку из розетки за шнур, усилие должно быть приложено к корпусу вилки.

Для исключения поражения электрическим током запрещается: часто включать и выключать компьютер без необходимости, прикасаться к экрану и к тыльной стороне блоков компьютера, работать на средствах вычислительной техники и периферийном оборудовании мокрыми руками, работать на средствах вычислительной техники и периферийном оборудовании, имеющих нарушения целостности корпуса, нарушения изоляции проводов, неисправную индикацию включения питания, с признаками электрического напряжения на корпусе, класть на средства вычислительной техники и периферийном оборудовании посторонние предметы.

Запрещается под напряжением очищать от пыли и загрязнения электрооборудование.

Запрещается проверять работоспособность электрооборудования в неприспособленных для эксплуатации помещениях с токопроводящими полами, сырых, не позволяющих заземлить доступные металлические части.

Недопустимо под напряжением проводить ремонт средств вычислительной техники и периферийного оборудования. Ремонт электроаппаратуры производится только специалистами-техниками с соблюдением необходимых технических требований.

Во избежание поражения электрическим током, при пользовании электроприборами нельзя касаться одновременно каких-либо трубопроводов, батарей отопления, металлических конструкций , соединенных с землей.

При пользовании электроэнергией в сырых помещениях соблюдать особую осторожность.

Требования безопасности в аварийных ситуациях - При обнаружении неисправности немедленно обесточить электрооборудование, оповестить администрацию. Продолжение работы возможно только после устранения неисправности.

При обнаружении оборвавшегося провода необходимо немедленно сообщить об этом администрации, принять меры по исключению контакта с ним людей. Прикосновение к проводу опасно для жизни.

Во всех случаях поражения человека электрическим током немедленно вызывают врача. До прибытия врача нужно, не теряя времени, приступить к оказанию первой помощи пострадавшему.

Необходимо немедленно начать производить искусственное дыхание, наиболее эффективным из которых является метод ?рот в рот¦ или ?рот в нос¦, а также наружный массаж сердца.

Искусственное дыхание пораженному электрическим током производится вплоть до прибытия врача.

На рабочем месте запрещается иметь огнеопасные вещества

В помещениях запрещается:

- зажигать огонь;

- включать электрооборудование, если в помещении пахнет газом;

- курить;

- сушить что-либо на отопительных приборах;

- закрывать вентиляционные отверстия в электроаппаратуре.

Источниками воспламенения являются:

- искра при разряде статического электричества;

- искры от электрооборудования;

- искры от удара и трения;

- открытое пламя.

При возникновении пожароопасной ситуации или пожара персонал должен немедленно принять необходимые меры для его ликвидации, одновременно оповестить о пожаре администрацию.

Помещения с электрооборудованием должны быть оснащены огнетушителями типа ОУ-2 или ОУБ-3.

Требования безопасности по окончании работы

После окончания работы необходимо обесточить все средства вычислительной техники и периферийное оборудование. В случае непрерывного производственного процесса необходимо оставить включенными только необходимое оборудование.

Инструкция по технике безопасности при работе на компьютере

Введение

Настоящая инструкция предназначена для предотвращения неблагоприятного воздействия на человека вредных факторов, сопровождающих работы со средствами вычислительной техники и периферийным оборудованием.

Настоящая инструкция подлежит обязательному и безусловному выполнению. За нарушение инструкции виновные несут ответственность в административном и судебном порядке в зависимости от характера последствий нарушения.

Соблюдение правил безопасной работы является необходимым условием предупреждения производственного травматизма.

Общие положения.

Область распространения и порядок применения инструкции:

Настоящая инструкция распространяется на персонал, эксплуатирующий средства вычислительной техники и периферийное оборудование. Инструкция содержит общие указания по безопасному применению электрооборудования в учреждении. Требования настоящей инструкции являются обязательными, отступления от нее не допускаются.

Требования к персоналу, эксплуатирующему средства вычислительной техники и периферийное оборудование:

К самостоятельной эксплуатации электроаппаратуры допускается только специально обученный персонал не моложе 18 лет, пригодный по состоянию здоровья и квалификации к выполнению указанных работ.

Перед допуском к работе персонал должен пройти вводный и первичный инструктаж по технике безопасности с показом безопасных и рациональных приёмов работы. Затем не реже одного раза в 6 мес. проводится повторный инструктаж, возможно, с группой сотрудников одинаковой профессии в составе не более 20 человек. Внеплановый инструктаж проводится при изменении правил по охране труда, при обнаружении нарушений персоналом инструкции по технике безопасности, изменении характера работы персонала.

В помещениях, в которых постоянно эксплуатируется электрооборудование, должны быть вывешены в доступном для персонала месте инструкции по технике безопасности, в которых также должны быть определены действия персонала в случае возникновения аварий, пожаров, электротравм.

Руководители структурных подразделений несут ответственность за организацию правильной и безопасной эксплуатации средств вычислительной техники и периферийного оборудования, эффективность их использования; осуществляют контрольза выполнением персоналом требований настоящей инструкции по технике безопасности.

2.2 Обеспечение пожаро-и взрывобезопасности

Главная причина пожаров на предприятии - нарушение ТП. Мероприятия по пожарной профилактике подразделяются на организационные, технические и эксплуатационные. Организационные мероприятия предусматривают правильную эксплуатацию машин, правильное содержание зданий и противопожарный инструктаж рабочих и служащих. К техническим мероприятиям относятся соблюдение противопожарных норм, правил при проектировании зданий, при устройстве электропроводки, отопления, вентиляции и освещения. Мероприятия режимного характера - запрещение курения в неустановленных местах, производство сварных и огнеопасных работ в пожароопасных помещениях. Эксплуатационные мероприятия - профилактические осмотры, ремонт и испытания технологического оборудования.

При эксплуатации ЭВМ не исключена опасность различного рода возгораний. В современных компьютерах очень высока плотность размещения элементов электронных систем, в непосредственной близости друг от друга располагаются соединительные провода, коммуникационные кабели. При протекании по ним электрического тока выделяется значительное количество теплоты, что может привести к повышению температуры отдельных узлов до 80-100 "С. При этом возможны оплавление изоляции соединительных проводов, их оголение и, как следствие, короткое замыкание, сопровождаемое искрением, которое ведет к недопустимым перегрузкам элементов электронных схем. Перенагреваясь, они сгорают с разбрызгиванием искр.

Для отвода избыточного тепла от ЭВМ служат системы вентиляции и кондиционирования воздуха. Однако эти системы также представляют дополнительную пожарную опасность для машинного зала и других помещений, так как, с одной стороны, воздуховоды обеспечивают подачу кислорода, являющегося окислителем, во все помещения, а с другой - при возникновении пожара быстро распространяют огонь и продукты горения по всем помещениям и устройствам, с которыми они связаны.

Питание к электроустановкам подается по кабельным линиям, которые представляют особую пожарную опасность. Наличие горючего изоляционного материала, вероятных источников зажигания в виде электрических искр и дуг, разветвленность и труднодоступность делают кабельные линии местами наиболее вероятного возникновения и развития пожара.

Эксплуатация ЭВМ связана с необходимостью проведения обслуживающих, ремонтных и профилактических работ. При этом используют различные смазочные материалы, легковоспламеняющиеся жидкости, прокладывают временные электропроводки, ведут пайку и чистку отдельных узлов и деталей. Возникает дополнительная пожарная опасность, требующая соответствующих мер пожарной профилактики.Для предупреждения возгорания все виды кабелей следует прокладывать в металлических газонаполненных трубах. В машинных залах кабельные линии прокладывают под технологическими съемными полами, которые выполняют из негорючих или трудногорючих материалов с пределом огнестойкости не менее 0,5 ч.

В помещениях вычислительного центра пожарные краны устанавливают в коридорах, на площадках лестничных клеток, у входов. Ручные углекислотные огнетушители устанавливают в помещениях из расчета один огнетушитель на 40-50 м2.

В случае пожара срабатывает находящаяся в помещениях автоматическая установка пожаротушения (АУП). Чаще всего применяются газовые АУП. Они снабжены световой и звуковой сигнализацией.

Для предотвращения распространения огня во время пожара из одной части здания на другую устраивают противопожарные преграды в виде противопожарных стен, перегородок, перекрытий, зон, тамбуров-шлюзов, дверей, окон, люков, клапанов.

В здании на случай возникновения пожара предусматривается не менее двух эвакуационных выходов; но через машинный зал, имеющий также не менее двух выходов, не должны проходить пути эвакуации сотрудников, работающих в других подразделениях. В других производственных помещениях допускается проектировать один выход, если расстояние от наиболее удаленного места до выхода не превышает 25 м, а количество работающих в смене не более 25 человек. Проходы, коридоры и рабочие места не следует загромождать архивными материалами, бумагой. На эвакуационных путях устанавливают как естественное, так и искусственное аварийное освещение.

Для хранения носителей информации используются несгораемые металлические шкафы, двери в хранилище также должны быть несгораемыми.

Комплекс организационных и технических мероприятий пожарной профилактики позволяет предотвратить пожар, а в случае его возникновения обеспечить безопасность людей, ограничить распространение огня, а также создать условия для успешного тушения пожара.

3 Организационно-экономическая часть

3.1 Введение

Неотъемлемой частью современного материального производства общества стал технологический подход, который выступает как развитие, внедрение научных достижений в социальные процессы, а также в образование. Современный период развития общества характеризуется сильным влиянием на него компьютерных технологий, проникающих во все сферы человеческой деятельности, обеспечивающих распространение информационных потоков, образуя глобальное информационное пространство. Неотъемлемой и важной частью этих процессов является компьютеризация образования.

На современном этапе под компьютеризацией образования понимается процесс внедрения новых информационных технологий во все виды и формы образовательной деятельности и на этой основе формирование новых образовательных моделей. Очевидно, что достижение этих целей возможно при наличии современной компьютерной техники, соответствующего уровня компьютерной подготовки администрации школ, педагогов, совершенствования учебно-методических комплексов на основе разноуровневого подхода в обучении детей, создания индустрии производства программной продукции и системы ее поставки, проката.

Под компьютерными технологиями мы понимаем дидактическую сущность компьютерных технологий обучения и их соотношение с педагогическими технологиями, а также их место в структуре организации учебного процесса в вузе. Для понимания компьютерных технологий в образовании необходимо разобраться с сутью этого понятия.

Итак, современные технологии основаны на достижениях научно-технического прогресса и ориентированы на производство продукта: материальная технология создаёт материальный продукт, информационная технология (ИТ) — информационный продукт. Технология это также научная дисциплина, разрабатывающая и совершенствующая способы и инструменты производства. В быту технологией принято называть описание производственных процессов, инструкции по их выполнению, технологические требования и т.д. Технологией или технологическим процессом часто называют также сами операции добычи, транспортировки и переработки, являющихся основой производственного процесса.

3.2 Оценка затрат на создание нового программного продукта

Автоматизированная информационная подсистема учёта работы учителя-логопеда психолого-медико-педагогической консультации разрабатывалась в течении28 рабочих дней, одним программистом. В месяце 21 рабочий день.

Текущие затраты включают в себя заработную плату исполнителя, затраты на единое социальное страхование, затраты на материалы, затраты на электроэнергию, амортизационные отчисления, непрямые затраты, плановые начисления.

Заработная плата разработчика ПО -составляет основной фонд заработной платы

 , грн (1)

где Т – месячный оклад программиста, грн

t– фактическое время, дни

- число рабочих дней в месяце

(грн)

Дополнительная заработная плата определяется по формуле

Здоп =Зпр +Зотп +Зну, грн (2)

где Зпр–премия, грн,

Зотп – оплата отпуска, грн,

Зну – оплата неявок по уважительной причине, грн,

Определение суммы премии.

Премиальные доплаты определяются по формуле:

Зпр = Зт ∙ , грн (3)

где П – процент премии, %

Зпр = 3600,00 ∙ = 288,00 (грн)

Так как рассматривается разовое исполнение работы на заказ, оплата отпуска Зотпне предусмотрена.

Оплата за неявки по уважительным причинам

Зну= Зт∙, грн (4)

где Н1 - процент неявок по уважительной причине

Зну= 3600,00 ∙ = 36,00 (грн)

Здоп =288 + 0 +36 = 324 (грн)

Общая заработная плата исполнителя определяется по формуле:

Зобщ = Зосн + Здоп, грн (5)

Зобщ = 3600 + 324 = 3924 (грн)

Отчисления на социальное страхование проводится в соответствии с Налоговым кодексом Украины из общего фонда заработной платы.

Процент отчислений на социальное страхование для дошкольного образования.

Отчисление на ЕВСС рассчитываются по формуле :

|  |  |  |
| --- | --- | --- |
|  | ЗЕВСС = Зобщ ∙грн | (6) |

где Н2- отчисление на ЕВСС,%

ЗЕВСС = 3924 ∙ = 1443,25 (грн)

Расходы на затратные материалы включают затраты на канцелярские принадлежности, затратные материалы для принтера, бумага, носители информации и т.д., расчет производится в таблице 2.

Таблица 2 – Расходы на затратные материалы

|  |  |  |
| --- | --- | --- |
| Наименование | Кол-во | Сумма, грн |
| Канцелярские принадлежности:  Ручка,  Тетрадь | 10 | 5,5  12 |
| Бумага для печати (A4) | 1 | 39 |
| Накопитель информации | 1 | 10 |
| картридж на принтер (Philips LaserMFD 6135d) | 4 | 201,5 |
| Итого (М) |  | 268 |

Затраты на электроэнергию определяются по формуле

|  |  |  |  |
| --- | --- | --- | --- |
| Зэл =,грн (7) | | | |
| где |  | – | потребляемая мощность ЭВМ и освещения соответственно (W1, кВт – мощность выбранного компьютера и монитора,  W2 , кВт – мощность освещения) |
|  | - | | время использования ЭВМ и освещения соответственно, дни |
|  | – | | количество рабочих часов в день |
|  | – | | цена 1кВт∙час электроэнергии, = 0,84грн  Зэл ==24,46 (грн) |

Во время работы используется как электроосвещение, так и ЭВМ, которая работает от электроэнергии.

Амортизационные отчисления включают в себя амортизацию оборудования, амортизацию программного обеспечения, амортизацию здания.

Амортизация оборудования рассчитывается по формуле

|  |  |  |
| --- | --- | --- |
|  | , грн | (8) |

где - первичная стоимость оборудования, грн (расчет в таблице 3.2)

– норма амортизации оборудования, %

– календарный час, дни

1. – количество календарных дней в году, дни

Для определения первичной стоимости оборудования составляется таблица 3.

Таблица 3 – Калькуляция оборудования

|  |  |
| --- | --- |
| Наименование | Сумма, грн |
| Acer Aspire XC3110G(DT.SLJME.032)Intel Pentium G620 (2.3 ГГц)/RAM 2 GB/HDD 260 GB/Intel HD Graphics | 3502 |
| Монитор (Samsung SyncMaster 2243nw 22”) | 1200 |
| Клавиатура (4TECH LCD-720) | 45 |
| Мышь(4TECH Mouse720) | 25 |
| Принтер/Факс (Philips LaserMFD 6135d) | 647 |
| Итого затраты | 5329 |

= 204,4 (грн)

Амортизация программного обеспечения определяем по формуле

|  |  |  |
| --- | --- | --- |
|  | , грн | (9) |

где  - первичная стоимость программного обеспечения, грн

 - норма амортизации программного обеспечения, %

365 – количество календарных дней в году, дни

Норма амортизации программного обеспечения принимается в соответствии с Налоговым кодексом Украины для программного обеспечения  = 50%

Для определения первичной стоимости программного обеспечения составляется таблица 4.

Таблица 4 – Калькуляция программного обеспечения

|  |  |
| --- | --- |
| Наименование | Сумма, грн |
| Windows7 | 703,03 |
| MS Office | 1351 |
| Итого | 2054,03 |

 = 105,48 (грн)

Общую амортизацию определяем по формуле:

|  |  |  |
| --- | --- | --- |
|  | , грн | (10) |

 = 309,88 (грн)

В непроизводственные расходы () входит содержание и ремонт здания, оборудования, содержание административного персонала. Эти затраты составляют от 1,5% до 3% от основной заработной платы и рассчитываются по формуле:

Знепр =Зосн∙ ,грн (11)

Знепр =3600∙ = 108 (грн)

Оценка затрат на создание нового программного продукта определяется по формуле:

|  |  |  |
| --- | --- | --- |
|  | Sі =Зобщ + ЗЕВСС  +М + Зэл + А + Знепр, грн | (12) |
|  | Sі =3924 + 1443,25 +268 + 24,46 + 309,88 + 108 = 6077,59 (грн) |  |

Плановые накопления составляют от 8% до 10% от себестоимости затрат. И вычисляются по формуле:

Знак =Sі∙ , грн (13)

Знак =6077,59∙  = 607,75 (грн)

Полученные данные сводятся в таблицу плановой калькуляции затрат (Таблица 5).

Таблица 5 – Плановая калькуляция затрат

|  |  |
| --- | --- |
| Наименование статей | Сумма, грн |
| Заработная плата программиста | 3924 |
| Отчисления на социальное страхование | 1443,25 |
| Расходы на затратные материалы | 268 |
| Расходы на электроэнергию | 24,46 |
| Амортизационные отчисления | 309,88 |
| Непроизводственныерасходы | 108 |
| Общая себестоимость | 6077,59 |
| Плановые накопления | 607,75 |
| Общиезатраты(S) | 6685,34 |

3.3 Оценка экономической эффективности

При проведении маркетингового исследования найдены аналогичные программмные продукты.

Программный продукт 1 ("Играем и учимся") почти удовлетворяет поставленной задаче. Однако функциональность данного программного обеспечения не является максимальной, и множество функций, которые необходимы сейчас для учёта и работы с детьми - отсутствуют. К тому же распространение данного программного продукта является коммерческим.

Программный продукт 2 («Ваш логопед») не полностью соответствует требованиям поставленной задачи. Является конечным вариантом, что говорит о невозможности добавление необходимых функций. Так же данное приложение является устаревшим и не поддерживает современные ОС (Windows 7,8). Данный программный продукт распространяется на коммерческой основе.

Результаты сведены в таблице 6.

Таблица 6 – Сводные данные маркетингового исследования

|  |  |  |
| --- | --- | --- |
| Наименование ПО | Процент соответствия поставленной задаче | Стоимость |
| Программный продукт 1: «Играем и учимся»; | 80 % | 5000,00 |
| Программный продукт 2: «Ваш логопед» | 60 % | 4000,00 |
| Разрабатываемый программный продукт (АИС учёта работы учителя-логопеда ПМПК при детском саде №166) | 100 % | 6685,34 |

В результате проведённого исследования можно заключить, что разработка нового программного обеспечения целесообразна, но данный продукт достаточно дорог. Для полноты исследования необходимо провести оценку экономической эффективности.

Экономическую эффективность просчитаем на основании допущения уменьшения время работы и затрат на протяжении рабочего дня.

Рассмотрим рабочий день учителя-логопеда: продолжительность смены 8 часов, для составления отчётов и работы с документацией, такими как: список зачисленных детей, речевой экран, анализ результативности, индивидуальные карты, планы работы.необходимо в среднем 4 часа. В то время как сразрабатываемым ПО это время сокращается в 5 раз, то есть необходимо48 минут (0,8 часа).

Определить затраты на электроэнергию при работе вручную по формуле:

Зэлруч= W2 ∙ tсм ∙ Сэл, грн (14)

гдеW2- потребляемая мощность освещения, кВт∙час

tсм- продолжительность рабочего дня, час

Сэл- стоимость электроэнергии, грн

Зэлруч= 0.04 ∙ 4 ∙ 0,84 = 0,13 (грн)

Произвести такую же оценку при работе с использованием программы.

Определим затраты на электроэнергию при использовании ЭВМ по формуле:

Зэлавт= (W1+W2) ∙t4 ∙ Сэл, грн (15)

где t4 - время, которое потратит работник с использованием компьютера на выполнение объема работы, час

W1- потребляемая мощность ЭВМ, кВт∙час

Зэлавт= (0.1+0.04) ∙0,8 ∙ 0,84 = 0,09 (грн)

Полученные данные свести в таблицу 7.

Таблица 7 – Сводная таблица

|  |  |  |
| --- | --- | --- |
| Показатель | Операции | |
| Вручную | Программно |
| Выполнениевсехдолжностныйинструкций за день, мин | 240 | 48 |
| Количествоопераций за рабочий день | 1 | 5 |
| Затраты на выполнение обязанностей, грн:   * на электроэнергию * на затратные материалы | 0.13  13.00 | 0.09  0,5 |
| Итого затрат, грн | 13.13 | 0.59 |

Годовую экономию определяем по формуле:

Ег = (Свр – Спр) ∙tмес (1 + Нс)∙ 12, грн*,* (16)

где Нс- нормативный коэффициент (0,52)

Свр - стоимость при работе вручную, грн

Спр - стоимость с использованием программы, грн

- число рабочих дней в месяце (= 28дн)

Ег = (13,15–0,59) ∙ 22 (1 + 0,52) ∙ 12 = 5040,07 (грн)

Срок окупаемости вложенных затрат при разработке программы определяется по формуле

|  |  |  |
| --- | --- | --- |
|  | Токуп =, год | (17) |

где S - общие затраты, грн (определены в таблице 3.4,)

Ег - сумма годовой экономии, грн

Токуп = = 1,33 (г.)

На основании проведенного экономического анализа, можно сделать вывод, что создание нового программного продукта актуально, и он на современном рынке будет конкурентоспособным.

Вывод

В ходе выполнения дипломного проекта была достигнута основная цель работы – разработана автоматизированная информационная подсистема учёта работы учителя-логопеда.

Для достижения поставленной цели были проведены изучение и анализ уже существующей структуры автоматизированных систем. При этом был выявлен ряд недостатков существующей системы, которые возможно устранить за счет разработки и внедрения новой АИС.

Для осуществления разработки были использованы следующие технологии:

* для создания базы данных, хранящей необходимую для функционирования АИС, информацию о детях в логопедических пунктах, и последующей работы с ней была выбрана СУБД MySQL. MySQL отличатся хорошей скоростью работы, надежностью, гибкостью и при этом распространяется совершенно бесплатно;
* в качестве языка программирования был выбран С++, предоставляющий возможность создания различного рода приложений, управляемых базами данных, и поддерживающий СУБД MySQL;
* в качестве интегрированной среды разработки была выбрана Embarcadero RAD Studio, которая обладает простотой, комфортом и удобностью при разработке быстрых и безопасных приложений на языке программирования С++ и Delphi.

Общие затраты составляют 6685,34 грн. Исходя из проведенного экономического анализа, программа окупается за 1,33 года. Создание нового программного продукта актуально, и он на современном рынке будет конкурентоспособным.

Перечень использованных источников

1. Элджер Дж. C++: библиотека программиста. Пер. с англ. - СПб.: ЗАО "Издательство Питер", 1999. - 320 с.
2. Мейерс С. Наиболее эффективное использование C++. 35 новых рекомен-даций по улучшению ваших программ и проектов: Пер. с англ. - М.: ДМК Пресс, 2000. - 304 с
3. MySQLHighAvailability / Обеспечение высокой доступности систем на основе MySQL, halmann L., Kindahl M., Bell C., БХВ-Петербург, 624стр, 2010 г.
4. Х. Дейтел, П. Дейтел. Как программировать на C++: Пер. с англ. - Москва: ЗАО "Издательство БИНОМ", 1998. 1024с.
5. У. Сэвитч. C++ в примерах: Пер. с англ. - Москва: ЭКОМ, 1997. 736с.
6. К. Джамса. Учимся программировать на языке C++: Пер. с англ. - Москва: Мир, 1997. 320с.

Приложение А

Листинг

Unit1.h

#ifndefUnit1H

#define Unit1H

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <DBCtrls.hpp>

#include <DBGrids.hpp>

#include <ExtCtrls.hpp>

#include <Grids.hpp>

#include <jpeg.hpp>

#include <pngimage.hpp>

#include <GIFImg.hpp>

#include "sSkinManager.hpp"

//---------------------------------------------------------------------------

class TForm1 : public TForm

{

\_\_published: // IDE-managed Components

TImage \*Image1;

TImage \*Image2;

TLabel \*Label2;

TLabel \*Label1;

TImage \*Image3;

TLabel \*Label3;

TImage \*Image4;

TLabel \*Label4;

TImage \*Image5;

TLabel \*Label5;

TImage \*Image6;

TLabel \*Label6;

TImage \*Image7;

TLabel \*Label7;

TImage \*Image8;

TLabel \*Label8;

TsSkinManager \*sSkinManager1;

void \_\_fastcall Image2DblClick(TObject \*Sender);

void \_\_fastcall Image1DblClick(TObject \*Sender);

void \_\_fastcall Image4DblClick(TObject \*Sender);

void \_\_fastcall Image5DblClick(TObject \*Sender);

void \_\_fastcall Image6DblClick(TObject \*Sender);

void \_\_fastcall Image7DblClick(TObject \*Sender);

void \_\_fastcall Image8DblClick(TObject \*Sender);

void \_\_fastcall Image3DblClick(TObject \*Sender);

void \_\_fastcall FormClose(TObject \*Sender, TCloseAction &Action);

private: // User declarations

public: // User declarations

\_\_fastcall TForm1(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm1 \*Form1;

//---------------------------------------------------------------------------

#endif

Unit1.cpp

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit3.h"

#include "Unit4.h"

#include "Unit5.h"

#include "Unit6.h"

#include "Unit7.h"

#include "Unit8.h"

#include "Unit9.h"

#include "Unit14.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "sSkinManager"

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Image2DblClick(TObject \*Sender)

{

Form4->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Image1DblClick(TObject \*Sender)

{

Form3->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Image4DblClick(TObject \*Sender)

{

Form5->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Image5DblClick(TObject \*Sender)

{

Form6->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Image6DblClick(TObject \*Sender)

{

Form7->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Image7DblClick(TObject \*Sender)

{

Form8->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Image8DblClick(TObject \*Sender)

{

Form9->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Image3DblClick(TObject \*Sender)

{

Form14->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::FormClose(TObject \*Sender, TCloseAction &Action)

{

Application->MainForm->Close();

}

//---------------------------------------------------------------------------

Unit2.h

//---------------------------------------------------------------------------

#ifndef Unit2H

#define Unit2H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <ADODB.hpp>

#include <DB.hpp>

//---------------------------------------------------------------------------

class TDM : public TDataModule

{

\_\_published: // IDE-managed Components

TADOConnection \*ADOConnection1;

TADOTable \*analiz\_rezultativnosti;

TDataSource \*dsanaliz;

TADOTable \*grafik\_raboti;

TDataSource \*dsgrafik;

TADOTable \*Ind\_karta\_razv\_K;

TDataSource \*dsind\_k\_k;

TADOTable \*Ind\_karta\_razv;

TDataSource \*dsind\_k;

TADOTable \*karta\_re4\_razv;

TDataSource \*dskarta\_re4\_razv;

TADOTable \*plan\_raboti;

TDataSource \*dsPlan\_roboti;

TADOTable \*re4\_ekran;

TDataSource \*dsRe4\_ekran;

TADOTable \*spisok\_detey;

TDataSource \*dsSpisok\_detey;

TAutoIncField \*spisok\_deteyUK\_spiska;

TWideStringField \*spisok\_deteyFIO\_rebenka;

TDateField \*spisok\_deteyData\_rozhd;

TWideStringField \*spisok\_deteyAdres;

TDateField \*spisok\_deteyData\_pmpk;

TWideStringField \*spisok\_deteydiagnoz;

TIntegerField \*spisok\_deteyn\_protokola;

TIntegerField \*spisok\_deteyn\_gruppi;

TAutoIncField \*analiz\_rezultativnostiUK\_analiza;

TIntegerField \*analiz\_rezultativnostiUK\_spiska;

TDateField \*analiz\_rezultativnostiData\_opisania;

TWideStringField \*analiz\_rezultativnostiOpisanie;

TAutoIncField \*grafik\_rabotiUK\_grafik;

TWideStringField \*grafik\_rabotiponed;

TWideStringField \*grafik\_rabotivtor;

TWideStringField \*grafik\_rabotisreda;

TWideStringField \*grafik\_rabotichetverg;

TWideStringField \*grafik\_rabotipyat;

TAutoIncField \*Ind\_karta\_razv\_KUK\_karti\_k;

TIntegerField \*Ind\_karta\_razv\_KUK\_spiska;

TWideStringField \*Ind\_karta\_razv\_Kzvukoproizn;

TWideStringField \*Ind\_karta\_razv\_Kslog\_struk;

TWideStringField \*Ind\_karta\_razv\_Kmotorika;

TWideStringField \*Ind\_karta\_razv\_Kvospr\_cvetov;

TWideStringField \*Ind\_karta\_razv\_Kprostr\_vospr;

TWideStringField \*Ind\_karta\_razv\_Kschet\_oper;

TSmallintField \*Ind\_karta\_razv\_Kschet15;

TWideStringField \*Ind\_karta\_razv\_Ksravn\_predmetov;

TAutoIncField \*Ind\_karta\_razvUK\_karti\_nachalo;

TIntegerField \*Ind\_karta\_razvUK\_spiska;

TWideStringField \*Ind\_karta\_razvZvukoproiznoshenie;

TWideStringField \*Ind\_karta\_razvSlog\_strukt\_slova;

TWideStringField \*Ind\_karta\_razvMotorika;

TWideStringField \*Ind\_karta\_razvvospr\_cvetov;

TWideStringField \*Ind\_karta\_razvprostr\_vospr;

TWideStringField \*Ind\_karta\_razvschet\_oper\_glazami;

TSmallintField \*Ind\_karta\_razvschet15;

TWideStringField \*Ind\_karta\_razvsrav\_predmetov;

TAutoIncField \*karta\_re4\_razvUK\_karti;

TIntegerField \*karta\_re4\_razvUK\_spiska;

TIntegerField \*karta\_re4\_razvn\_dou;

TDateField \*karta\_re4\_razvData\_zachisleniya\_v\_gruppu;

TWideStringField \*karta\_re4\_razvzhalobi;

TWideStringField \*karta\_re4\_razvrodnoy\_yazik;

TWideStringField \*karta\_re4\_razvsostoyanie\_motoriki;

TWideStringField \*karta\_re4\_razvSluh;

TWideStringField \*karta\_re4\_razvObshee\_razvitie;

TWideStringField \*karta\_re4\_razvVnimanie;

TWideStringField \*karta\_re4\_razvRabotosposobnost;

TWideStringField \*karta\_re4\_razvPamyat;

TWideStringField \*karta\_re4\_razvGolos;

TWideStringField \*karta\_re4\_razvTembr;

TWideStringField \*karta\_re4\_razvDihanie;

TWideStringField \*karta\_re4\_razvArtik\_i\_dic;

TWideStringField \*karta\_re4\_razvGolosovie\_svyazki;

TWideStringField \*karta\_re4\_razvZubi;

TWideStringField \*karta\_re4\_razvgubi;

TWideStringField \*karta\_re4\_razvyazik;

TWideStringField \*karta\_re4\_razvdvizheniya;

TWideStringField \*karta\_re4\_razvprikus;

TAutoIncField \*plan\_rabotiuk\_plana;

TIntegerField \*plan\_rabotiuk\_spiska;

TWideStringField \*plan\_rabotisvist;

TWideStringField \*plan\_rabotiship;

TWideStringField \*plan\_rabotisonori;

TWideStringField \*plan\_rabotiafrikat;

TWideStringField \*plan\_rabotidrug;

TSmallintField \*plan\_rabotis;

TSmallintField \*plan\_rabotiz;

TSmallintField \*plan\_rabotishc;

TSmallintField \*plan\_rabotizh;

TSmallintField \*plan\_rabotisc;

TSmallintField \*plan\_rabotizc;

TSmallintField \*plan\_rabotilc;

TSmallintField \*plan\_rabotic;

TSmallintField \*plan\_rabotiy;

TSmallintField \*plan\_rabotich;

TSmallintField \*plan\_rabotish;

TSmallintField \*plan\_rabotil;

TSmallintField \*plan\_rabotir;

TSmallintField \*plan\_rabotirc;

TSmallintField \*plan\_rabotis\_z;

TSmallintField \*plan\_rabotis\_sc;

TSmallintField \*plan\_rabotis\_c;

TSmallintField \*plan\_rabotis\_sh;

TSmallintField \*plan\_rabotizh\_z;

TSmallintField \*plan\_rabotizh\_sh;

TSmallintField \*plan\_rabotich\_sc;

TSmallintField \*plan\_rabotich\_tc;

TSmallintField \*plan\_rabotilc\_l;

TSmallintField \*plan\_rabotich\_sch;

TSmallintField \*plan\_rabotishc\_sc;

TSmallintField \*plan\_rabotishc\_tc;

TSmallintField \*plan\_rabotishc\_ch;

TSmallintField \*plan\_rabotir\_l;

TSmallintField \*plan\_rabotir\_rc;

TSmallintField \*plan\_rabotirc\_lc;

TSmallintField \*plan\_rabotirc\_y;

TAutoIncField \*re4\_ekranuk\_ekrana;

TIntegerField \*re4\_ekranuk\_spiska;

TStringField \*re4\_ekrans;

TStringField \*re4\_ekranz;

TStringField \*re4\_ekranc;

TStringField \*re4\_ekransh;

TStringField \*re4\_ekranzh;

TStringField \*re4\_ekranch;

TStringField \*re4\_ekranshc;

TStringField \*re4\_ekranr;

TStringField \*re4\_ekranl;

TStringField \*re4\_ekranlc;

TStringField \*re4\_ekrank;

TStringField \*re4\_ekrang;

TStringField \*re4\_ekranh;

TWideStringField \*re4\_ekranrc;

TStringField \*re4\_ekranff\_fospr;

TStringField \*re4\_ekrangod\_obu4;

TADOTable \*rosp\_zanyatii;

TDataSource \*dsRosp\_zany;

TAutoIncField \*rosp\_zanyatiiuk\_rosp;

TWideStringField \*rosp\_zanyatiiponed;

TWideStringField \*rosp\_zanyatiivtor;

TWideStringField \*rosp\_zanyatiisreda;

TWideStringField \*rosp\_zanyatiichetverg;

TWideStringField \*rosp\_zanyatiipyat;

void \_\_fastcall analiz\_rezultativnostiAfterInsert(TDataSet \*DataSet);

private: // User declarations

public: // User declarations

\_\_fastcall TDM(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TDM \*DM;

//---------------------------------------------------------------------------

#endif

Unit2.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TDM \*DM;

//---------------------------------------------------------------------------

\_\_fastcall TDM::TDM(TComponent\* Owner)

: TDataModule(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TDM::analiz\_rezultativnostiAfterInsert(TDataSet \*DataSet)

{

DM->analiz\_rezultativnostiData\_opisania->Value=Now();

}

//---------------------------------------------------------------------------

Unit3.h

//---------------------------------------------------------------------------

#ifndef Unit3H

#define Unit3H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <DBCtrls.hpp>

#include <DBGrids.hpp>

#include <ExtCtrls.hpp>

#include <Grids.hpp>

#include <ComCtrls.hpp>

#include "RpCon.hpp"

#include "RpConDS.hpp"

#include "RpDefine.hpp"

#include "RpRave.hpp"

#include <Mask.hpp>

#include <Buttons.hpp>

#include <jpeg.hpp>

//---------------------------------------------------------------------------

class TForm3 : public TForm

{

\_\_published: // IDE-managed Components

TDBGrid \*DBGridSpisok\_detey;

TDBNavigator \*DBNavigator1;

TEdit \*Edit1;

TDateTimePicker \*DateTimePicker1;

TRvProject \*RvProject1;

TRvDataSetConnection \*RvDataSetConnection1;

TDateTimePicker \*DateTimePicker2;

TLabel \*Label1;

TDBEdit \*DBEdit1;

TBitBtn \*BitBtn1;

TBitBtn \*BitBtn2;

TBitBtn \*BitBtn3;

TBitBtn \*BitBtn4;

TLabel \*Label2;

TImage \*Image1;

TLabel \*Label3;

TLabel \*Label4;

void \_\_fastcall BitBtn1Click(TObject \*Sender);

void \_\_fastcall BitBtn2Click(TObject \*Sender);

void \_\_fastcall BitBtn3Click(TObject \*Sender);

void \_\_fastcall BitBtn4Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm3(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm3 \*Form3;

//---------------------------------------------------------------------------

#endif

Unit3.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit3.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "RpCon"

#pragma link "RpConDS"

#pragma link "RpDefine"

#pragma link "RpRave"

#pragma resource "\*.dfm"

TForm3 \*Form3;

//---------------------------------------------------------------------------

\_\_fastcall TForm3::TForm3(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::BitBtn1Click(TObject \*Sender)

{

TLocateOptions SearchOptions;

DM->spisok\_detey->Locate("FIO\_rebenka",this->Edit1->Text, SearchOptions<<loPartialKey<<loCaseInsensitive);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::BitBtn2Click(TObject \*Sender)

{

unsigned short y,d,m;

this->DateTimePicker1->Date.DecodeDate(&y,&m,&d);

TLocateOptions SearchOptions;

DM->spisok\_detey->Locate("Data\_rozhd",EncodeDate(y,m,d), SearchOptions<<loPartialKey<<loCaseInsensitive);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::BitBtn3Click(TObject \*Sender)

{

unsigned short y,d,m;

this->DateTimePicker1->Date.DecodeDate(&y,&m,&d);

TLocateOptions SearchOptions;

DM->spisok\_detey->Locate("Data\_pmpk",EncodeDate(y,m,d), SearchOptions<<loPartialKey<<loCaseInsensitive);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::BitBtn4Click(TObject \*Sender)

{

this->RvProject1->Execute();

}

//---------------------------------------------------------------------------

Unit4.h

//---------------------------------------------------------------------------

#ifndef Unit4H

#define Unit4H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <DBGrids.hpp>

#include <Grids.hpp>

#include <DBCtrls.hpp>

#include <Mask.hpp>

#include <Buttons.hpp>

#include <ComCtrls.hpp>

#include <ExtCtrls.hpp>

#include <jpeg.hpp>

//---------------------------------------------------------------------------

class TForm4 : public TForm

{

\_\_published: // IDE-managed Components

TLabel \*Label1;

TDBEdit \*DBEdit1;

TLabel \*Label2;

TDBEdit \*DBEdit2;

TLabel \*Label3;

TDBEdit \*DBEdit3;

TLabel \*Label4;

TDBEdit \*DBEdit4;

TLabel \*Label5;

TDBEdit \*DBEdit5;

TMonthCalendar \*MonthCalendar1;

TBitBtn \*BitBtn1;

TImage \*Image1;

void \_\_fastcall BitBtn1Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm4(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm4 \*Form4;

//---------------------------------------------------------------------------

#endif

Unit4.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit4.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm4 \*Form4;

//---------------------------------------------------------------------------

\_\_fastcall TForm4::TForm4(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::BitBtn1Click(TObject \*Sender)

{

if (MonthCalendar1->Visible) {

this->MonthCalendar1->Visible=False;

} else {

this->MonthCalendar1->Visible=True;

}

}

//---------------------------------------------------------------------------

Unit5.h

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit4.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm4 \*Form4;

//---------------------------------------------------------------------------

\_\_fastcall TForm4::TForm4(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::BitBtn1Click(TObject \*Sender)

{

if (MonthCalendar1->Visible) {

this->MonthCalendar1->Visible=False;

} else {

this->MonthCalendar1->Visible=True;

}

}

//---------------------------------------------------------------------------

Unit5.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit5.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm5 \*Form5;

//---------------------------------------------------------------------------

\_\_fastcall TForm5::TForm5(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm5::BitBtn1Click(TObject \*Sender)

{

if (GroupBox1->Visible) {

this->GroupBox1->Visible=False;

} else {

this->GroupBox1->Visible=True;

}

}

//---------------------------------------------------------------------------

Unit6.h

//---------------------------------------------------------------------------

#ifndef Unit6H

#define Unit6H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <DBGrids.hpp>

#include <Grids.hpp>

#include <DBCtrls.hpp>

#include <Buttons.hpp>

#include <ExtCtrls.hpp>

#include <Mask.hpp>

#include <jpeg.hpp>

//---------------------------------------------------------------------------

class TForm6 : public TForm

{

\_\_published: // IDE-managed Components

TDBGrid \*DBGrid2;

TDBGrid \*DBGrid3;

TLabel \*Label1;

TDBGrid \*DBGrid5;

TLabel \*Label2;

TLabel \*Label3;

TDBEdit \*DBEdit1;

TLabel \*Label4;

TDBEdit \*DBEdit2;

TLabel \*Label5;

TDBEdit \*DBEdit3;

TDBNavigator \*DBNavigator1;

TDBNavigator \*DBNavigator2;

TDBNavigator \*DBNavigator3;

TDBGrid \*DBGrid1;

TBitBtn \*BitBtn1;

TGroupBox \*GroupBox1;

TDBGrid \*DBGrid4;

TImage \*Image1;

void \_\_fastcall BitBtn1Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm6(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm6 \*Form6;

//---------------------------------------------------------------------------

#endif

Unit6.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit6.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm6 \*Form6;

//---------------------------------------------------------------------------

\_\_fastcall TForm6::TForm6(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm6::BitBtn1Click(TObject \*Sender)

{

if (GroupBox1->Visible) {

this->GroupBox1->Visible=False;

} else {

this->GroupBox1->Visible=True;

}

}

//---------------------------------------------------------------------------

Unit7.h

//---------------------------------------------------------------------------

#ifndef Unit7H

#define Unit7H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <DBGrids.hpp>

#include <Grids.hpp>

#include <DBCtrls.hpp>

#include <Mask.hpp>

#include <ExtCtrls.hpp>

#include <Buttons.hpp>

#include <jpeg.hpp>

//---------------------------------------------------------------------------

class TForm7 : public TForm

{

\_\_published: // IDE-managed Components

TLabel \*Label1;

TDBEdit \*DBEdit1;

TLabel \*Label2;

TDBEdit \*DBEdit2;

TLabel \*Label3;

TDBEdit \*DBEdit3;

TDBNavigator \*DBNavigator1;

TLabel \*Label4;

TDBEdit \*DBEdit4;

TLabel \*Label5;

TDBEdit \*DBEdit5;

TLabel \*Label6;

TDBEdit \*DBEdit6;

TLabel \*Label7;

TDBEdit \*DBEdit7;

TLabel \*Label8;

TDBEdit \*DBEdit8;

TLabel \*Label9;

TDBEdit \*DBEdit9;

TLabel \*Label10;

TDBEdit \*DBEdit10;

TLabel \*Label11;

TDBEdit \*DBEdit11;

TLabel \*Label12;

TDBEdit \*DBEdit12;

TLabel \*Label13;

TDBEdit \*DBEdit13;

TLabel \*Label14;

TDBEdit \*DBEdit14;

TLabel \*Label15;

TDBEdit \*DBEdit15;

TLabel \*Label16;

TDBEdit \*DBEdit16;

TLabel \*Label17;

TDBEdit \*DBEdit17;

TLabel \*Label18;

TDBEdit \*DBEdit18;

TLabel \*Label19;

TDBEdit \*DBEdit19;

TLabel \*Label20;

TDBEdit \*DBEdit20;

TLabel \*Label21;

TDBEdit \*DBEdit21;

TLabel \*Label22;

TDBEdit \*DBEdit22;

TLabel \*Label23;

TDBEdit \*DBEdit23;

TDBNavigator \*DBNavigator2;

TBitBtn \*BitBtn1;

TGroupBox \*GroupBox1;

TDBGrid \*DBGrid4;

TImage \*Image1;

void \_\_fastcall BitBtn1Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm7(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm7 \*Form7;

//---------------------------------------------------------------------------

#endif

Unit7.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit7.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm7 \*Form7;

//---------------------------------------------------------------------------

\_\_fastcall TForm7::TForm7(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm7::BitBtn1Click(TObject \*Sender)

{

if (GroupBox1->Visible) {

this->GroupBox1->Visible=False;

} else {

this->GroupBox1->Visible=True;

}

}

//--------------------------------------------------------------------------

Unit8.h

//---------------------------------------------------------------------------

#ifndef Unit8H

#define Unit8H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <DBGrids.hpp>

#include <Grids.hpp>

#include <DBCtrls.hpp>

#include <Mask.hpp>

#include <ExtCtrls.hpp>

#include "RpCon.hpp"

#include "RpConDS.hpp"

#include "RpDefine.hpp"

#include "RpRave.hpp"

#include <Buttons.hpp>

#include <jpeg.hpp>

//---------------------------------------------------------------------------

class TForm8 : public TForm

{

\_\_published: // IDE-managed Components

TLabel \*Label1;

TDBEdit \*DBEdit1;

TLabel \*Label2;

TDBEdit \*DBEdit2;

TLabel \*Label3;

TDBEdit \*DBEdit3;

TLabel \*Label4;

TDBEdit \*DBEdit4;

TLabel \*Label5;

TDBEdit \*DBEdit5;

TLabel \*Label6;

TDBEdit \*DBEdit6;

TLabel \*Label7;

TDBEdit \*DBEdit7;

TLabel \*Label8;

TDBEdit \*DBEdit8;

TLabel \*Label9;

TDBEdit \*DBEdit9;

TLabel \*Label10;

TDBEdit \*DBEdit10;

TLabel \*Label11;

TDBEdit \*DBEdit11;

TLabel \*Label12;

TDBEdit \*DBEdit12;

TLabel \*Label13;

TDBEdit \*DBEdit13;

TLabel \*Label14;

TDBEdit \*DBEdit14;

TLabel \*Label15;

TDBEdit \*DBEdit15;

TLabel \*Label16;

TDBEdit \*DBEdit16;

TDBNavigator \*DBNavigator1;

TDBNavigator \*DBNavigator2;

TGroupBox \*GroupBox1;

TStaticText \*StaticText1;

TStaticText \*StaticText4;

TLabel \*Label17;

TDBEdit \*DBEdit17;

TStaticText \*StaticText3;

TLabel \*Label18;

TDBEdit \*DBEdit18;

TBitBtn \*BitBtn1;

TEdit \*Edit1;

TBitBtn \*BitBtn2;

TEdit \*Edit2;

TBitBtn \*BitBtn3;

TBitBtn \*BitBtn4;

TLabel \*Label19;

TImage \*Image1;

TRvDataSetConnection \*RvDataSetConnection20;

TRvDataSetConnection \*RvDataSetConnection30;

TRvProject \*RvProject25;

void \_\_fastcall BitBtn1Click(TObject \*Sender);

void \_\_fastcall BitBtn2Click(TObject \*Sender);

void \_\_fastcall BitBtn3Click(TObject \*Sender);

void \_\_fastcall BitBtn4Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm8(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm8 \*Form8;

//---------------------------------------------------------------------------

#endif

Unit8.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit8.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "RpCon"

#pragma link "RpConDS"

#pragma link "RpDefine"

#pragma link "RpRave"

#pragma resource "\*.dfm"

TForm8 \*Form8;

//---------------------------------------------------------------------------

\_\_fastcall TForm8::TForm8(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm8::BitBtn1Click(TObject \*Sender)

{

if (GroupBox1->Visible) {

this->GroupBox1->Visible=False;

} else {

this->GroupBox1->Visible=True;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm8::BitBtn2Click(TObject \*Sender)

{

TLocateOptions SearchOptions;

DM->spisok\_detey->Locate("FIO\_rebenka",this->Edit1->Text, SearchOptions<<loPartialKey<<loCaseInsensitive);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm8::BitBtn3Click(TObject \*Sender)

{

TLocateOptions SearchOptions;

DM->spisok\_detey->Locate("diagnoz",this->Edit2->Text, SearchOptions<<loPartialKey<<loCaseInsensitive);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm8::BitBtn4Click(TObject \*Sender)

{

this->RvProject25->Execute();

}

//---------------------------------------------------------------------------

Unit9.h

//---------------------------------------------------------------------------

#ifndef Unit9H

#define Unit9H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <ExtCtrls.hpp>

#include <jpeg.hpp>

//---------------------------------------------------------------------------

class TForm9 : public TForm

{

\_\_published: // IDE-managed Components

TImage \*Image1;

TImage \*Image2;

TImage \*Image3;

TImage \*Image4;

TLabel \*Label1;

TLabel \*Label2;

TLabel \*Label3;

TLabel \*Label4;

void \_\_fastcall Image1DblClick(TObject \*Sender);

void \_\_fastcall Image2DblClick(TObject \*Sender);

void \_\_fastcall Image3DblClick(TObject \*Sender);

void \_\_fastcall Image4DblClick(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm9(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm9 \*Form9;

//---------------------------------------------------------------------------

#endif

Unit9.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit9.h"

#include "Unit10.h"

#include "Unit11.h"

#include "Unit12.h"

#include "Unit13.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm9 \*Form9;

//---------------------------------------------------------------------------

\_\_fastcall TForm9::TForm9(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm9::Image1DblClick(TObject \*Sender)

{

Form10->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm9::Image2DblClick(TObject \*Sender)

{

Form11->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm9::Image3DblClick(TObject \*Sender)

{

Form12->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm9::Image4DblClick(TObject \*Sender)

{

Form13->Show();

}

//---------------------------------------------------------------------------

Unit10.h

//---------------------------------------------------------------------------

#ifndef Unit10H

#define Unit10H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include "ShockwaveFlashObjects\_OCX.h"

#include <OleCtrls.hpp>

#include <Menus.hpp>

#include <OleServer.hpp>

//---------------------------------------------------------------------------

class TForm10 : public TForm

{

\_\_published: // IDE-managed Components

TShockwaveFlash \*ShockwaveFlash1;

void \_\_fastcall FormCreate(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm10(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm10 \*Form10;

//---------------------------------------------------------------------------

#endif

Unit10.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit10.h"

#include "ShockwaveFlashObjects\_TLB.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "ShockwaveFlashObjects\_OCX"

#pragma resource "\*.dfm"

TForm10 \*Form10;

//---------------------------------------------------------------------------

\_\_fastcall TForm10::TForm10(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm10::FormCreate(TObject \*Sender)

{

ShockwaveFlash1->Movie="C:/1.swf";

}

//---------------------------------------------------------------------------

Unit11.h

//---------------------------------------------------------------------------

#ifndef Unit11H

#define Unit11H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include "ShockwaveFlashObjects\_OCX.h"

#include <OleCtrls.hpp>

//---------------------------------------------------------------------------

class TForm11 : public TForm

{

\_\_published: // IDE-managed Components

TShockwaveFlash \*ShockwaveFlash1;

void \_\_fastcall FormCreate(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm11(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm11 \*Form11;

//---------------------------------------------------------------------------

#endif

Unit11.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit11.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "ShockwaveFlashObjects\_OCX"

#pragma resource "\*.dfm"

TForm11 \*Form11;

//---------------------------------------------------------------------------

\_\_fastcall TForm11::TForm11(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm11::FormCreate(TObject \*Sender)

{

ShockwaveFlash1->Movie="C:/2.swf";

}

//--------------------------------------------------------------------------

Unit12.h

//---------------------------------------------------------------------------

#ifndef Unit12H

#define Unit12H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include "ShockwaveFlashObjects\_OCX.h"

#include <OleCtrls.hpp>

//---------------------------------------------------------------------------

class TForm12 : public TForm

{

\_\_published: // IDE-managed Components

TShockwaveFlash \*ShockwaveFlash1;

void \_\_fastcall FormCreate(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm12(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm12 \*Form12;

//---------------------------------------------------------------------------

#endif

Unit12.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit12.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "ShockwaveFlashObjects\_OCX"

#pragma resource "\*.dfm"

TForm12 \*Form12;

//---------------------------------------------------------------------------

\_\_fastcall TForm12::TForm12(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm12::FormCreate(TObject \*Sender)

{

ShockwaveFlash1->Movie="C:/3.swf";

}

//---------------------------------------------------------------------------

Unit13.h

//---------------------------------------------------------------------------

#ifndef Unit13H

#define Unit13H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include "ShockwaveFlashObjects\_OCX.h"

#include <OleCtrls.hpp>

//---------------------------------------------------------------------------

class TForm13 : public TForm

{

\_\_published: // IDE-managed Components

TShockwaveFlash \*ShockwaveFlash1;

void \_\_fastcall FormCreate(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm13(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm13 \*Form13;

//---------------------------------------------------------------------------

#endif

Unit13.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit13.h"

#pragma package(smart\_init)

#pragma link "ShockwaveFlashObjects\_OCX"

#pragma resource "\*.dfm"

TForm13 \*Form13;

//---------------------------------------------------------------------------

\_\_fastcall TForm13::TForm13(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm13::FormCreate(TObject \*Sender)

{

ShockwaveFlash1->Movie="C:/4.swf";

}

Unit14.h

//---------------------------------------------------------------------------

#ifndef Unit14H

#define Unit14H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <ComCtrls.hpp>

#include <Menus.hpp>

#include <DBCtrls.hpp>

#include <Mask.hpp>

#include <Buttons.hpp>

#include <ExtCtrls.hpp>

#include <jpeg.hpp>

//---------------------------------------------------------------------------

class TForm14 : public TForm

{

\_\_published: // IDE-managed Components

TDBRichEdit \*DBRichEdit1;

TBitBtn \*BitBtn1;

TBitBtn \*BitBtn2;

TBitBtn \*BitBtn3;

TBitBtn \*BitBtn4;

TBitBtn \*BitBtn5;

TDBNavigator \*DBNavigator1;

TDBRichEdit \*DBRichEdit2;

TDBRichEdit \*DBRichEdit3;

TDBRichEdit \*DBRichEdit4;

TDBRichEdit \*DBRichEdit5;

TBitBtn \*BitBtn6;

TImage \*Image1;

void \_\_fastcall BitBtn1Click(TObject \*Sender);

void \_\_fastcall BitBtn2Click(TObject \*Sender);

void \_\_fastcall BitBtn3Click(TObject \*Sender);

void \_\_fastcall BitBtn4Click(TObject \*Sender);

void \_\_fastcall BitBtn5Click(TObject \*Sender);

void \_\_fastcall BitBtn6Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm14(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm14 \*Form14;

//---------------------------------------------------------------------------

#endif

Unit14.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit14.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm14 \*Form14;

//---------------------------------------------------------------------------

\_\_fastcall TForm14::TForm14(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm14::BitBtn1Click(TObject \*Sender)

{

if (DBRichEdit1->Visible) {

this->DBRichEdit1->Visible=False;

} else {

this->DBRichEdit1->Visible=True;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm14::BitBtn2Click(TObject \*Sender)

{

if (DBRichEdit2->Visible) {

this->DBRichEdit2->Visible=False;

} else {

this->DBRichEdit2->Visible=True;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm14::BitBtn3Click(TObject \*Sender)

{

if (DBRichEdit3->Visible) {

this->DBRichEdit3->Visible=False;

} else {

this->DBRichEdit3->Visible=True;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm14::BitBtn4Click(TObject \*Sender)

{

if (DBRichEdit4->Visible) {

this->DBRichEdit4->Visible=False;

} else {

this->DBRichEdit4->Visible=True;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm14::BitBtn5Click(TObject \*Sender)

{

if (DBRichEdit5->Visible) {

this->DBRichEdit5->Visible=False;

} else {

this->DBRichEdit5->Visible=True;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm14::BitBtn6Click(TObject \*Sender)

{

this->DBRichEdit1->Visible=False;

this->DBRichEdit2->Visible=False;

this->DBRichEdit3->Visible=False;

this->DBRichEdit4->Visible=False;

this->DBRichEdit5->Visible=False;

}

Unit15.h

#ifndef Unit15H

#define Unit15H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <Buttons.hpp>

#include <ExtCtrls.hpp>

#include "acPNG.hpp"

//---------------------------------------------------------------------------

class TForm15 : public TForm

{

\_\_published: // IDE-managed Components

TLabel \*Label1;

TLabel \*Label2;

TEdit \*Password;

TEdit \*Edit1;

TBitBtn \*BitBtn1;

TBitBtn \*BitBtn2;

TImage \*Image1;

void \_\_fastcall BitBtn1Click(TObject \*Sender);

void \_\_fastcall BitBtn2Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcallTForm15(TComponent\* Owner);

boolEnter; //Переменная, которая отвечает за текущее действие, производимое с основной формой (0 - создание нового пользователя, 1 - вход в программу)

boolisok; //Переменная, которая отвечает за "Правильный вход"

String Crypt(String Data); //Функция шифровки данных

String UnCrypt(String Crypt); //Функция расшифровки данных

int p; //Некое число

};

String TForm15::Crypt(String Data){ //Описание функции шифровки

StringrString; //Возвращаемое значение

for(inti=1;i<Data.Length()+1;i++) //Перебираем каждый символ заданного текста

{

rString=rString+String(IntToHex(int(Data[i]+p),2)); //Просто метод шифрования (символ заменяется на свою позицию в таблице символов и прибавляется при этом загадочное число p, а затем все это записывается в 16 сс)

}

returnrString; //Возвращаем закодированную строку

}

//----------Дешифровка---------------------------

String TForm15::UnCrypt(String Crypt){

StringrString; //Возвращаемое значение

for(inti=1;i<Crypt.Length()+1;i+=2) // Проходим циклом по каждому второму символу, так как во время шифрования каждому символу выделяется 2 байта

{

rString=rString+String( char( StrToInt("0x"+Crypt.SubString(i,2))-p ) ); //Несложная расшифровка

}

returnrString; //Возвращаем раскодированную строку

}

//---------------------------------------------------------------------------

extern PACKAGE TForm15 \*Form15;

//---------------------------------------------------------------------------

#endif

Unit15.cpp

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit15.h"

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "acPNG"

#pragma resource "\*.dfm"

TForm15 \*Form15;

TStringList\* DataList=new TStringList; //Здесь будет хранится информация о пользователях

//---------------------------------------------------------------------------

\_\_fastcall TForm15::TForm15(TComponent\* Owner)

: TForm(Owner)

{

Enter=1; //Меняем значение этой переменной на 1 (вход)

p=17;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm15::BitBtn1Click(TObject \*Sender)

{

if( (Edit1->Text!="") && (Password->Text!="")){ //Если требуемые поля заполнены

DataList->LoadFromFile("data.dat"); //Загружаем наши данные из файла

if(Enter){ //Если пользователь входи в программу (значение равно 1)

{

if(DataList->IndexOfName(Edit1->Text)!=-1){ //Если заданное имя существует

if(Password->Text==UnCrypt(DataList->Values[Edit1->Text]) ) // Если введенный текст в поле "Пароль" равен расшифрованному значению пароля

{

isok=1; //Все хорошо, вход выполнен по правилам

MessageBox(0,"Вход выполнен","Проверка пройдена",0); //Информируем

Form1->Show(); //Показываем вторую форму, где содержаться "секретные данные"

Hide(); //Скрываем эту

}

elseMessageBox(0,"Введён неверный пароль","Ошибка авторизации",0); //Если пароль введен не правильно

}

elseMessageBox(0,"Введён неверный логин","Ошибка авторизации",0); //Если заданный пользователь не зарегистрирован, то сообщаем об этом пользователю

}

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm15::BitBtn2Click(TObject \*Sender)

{

Close();

}