

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«РЯЗАНСКИЙ ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»

ОТЧЁТ

ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ
ПО МОДУЛЮ ПМ.04

«СОПРОВОЖДЕНИЕ И АДМИНИСТРИРОВАНИЕ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ»

СПЕЦИАЛЬНОСТИ 09.02.07 - ИНФОРМАЦИОННЫЕ СИСТЕМЫ И
ПРОГРАММИРОВАНИЕ

Выполнил: студент 4 курса группы ИСП-19

Марьяшин Дмитрий Александрович

Сроки прохождения:

Дата начала практики: 03.04.2023

Дата окончания практики: 19.04.2023

Место прохождения практики:

МДОУ «Колокольчик»

Руководитель практики от учебного заведения:

Васильева А.С.

Руководитель практики от организации:

Молостова Н.А.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ОПИСАНИЕ ОРГАНИЗАЦИИ	4
2 ОПИСАНИЕ КОМПЬЮТЕРНОЙ ИНФРАСТРУКТУРЫ ОРГАНИЗАЦИИ	7
2.1 Аппаратная часть	7
2.2 Программная часть.....	8
3 ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОГО ЗАДАНИЯ.....	9
3.1 Постановка задачи.....	9
3.2 Проектирование.....	11
3.3 Разработка информационной системы.....	12
3.3.1 Разработка концептуальной модели базы данных	12
3.3.2 Разработка физической модели базы данных.....	20
3.3.3 Разработка пользовательского интерфейса	23
3.3.4 Защита информационной системы	35
3.4 Тестирование	36
3.5 Внедрение	45
4 ОБЗОР РАБОТЫ РАЗРАБОТАННОЙ ИС	47
4.1 Инсталляция.....	47
4.1.1 Файловая система.....	47
4.1.2 Клиент - Серверная система.....	50
4.2 Запуск	58
4.3 Функции информационной системы.....	59
4.3.1 Работа со справочниками	59
4.3.2 Регистрация и учет детей.....	60
4.3.3 Работа с табелем.....	62
ЗАКЛЮЧЕНИЕ.....	65
СПИСОК ИСТОЧНИКОВ	66

ВВЕДЕНИЕ

Прохождение производственной практики является важным элементом учебного процесса по подготовке специалиста в определенной области. Во время её прохождения студент применяет полученные в процессе обучения знания, умения и навыки на практике.

Производственная практика – это попытка соединить теоретическую подготовку с формированием практических навыков, обратную связь со стороны компаний и организаций, принимающих студентов на практику, о качестве обучения, а также получение дополнительной информации о том, над чем нужно поработать, чтобы соответствовать современным требованиям рынка труда.

Цель производственной практики – закрепление и обновление знаний, овладение практическими навыками, умениями и технологиями практической деятельности, а также формирование профессиональной компетентности, развитие деловых, организаторских и личностных качеств студентов для последующей эффективной работы на предприятиях, организациях и учреждениях в условиях рыночных отношений.

Благодаря производственной практики:

1. Можно получить достоверную информацию о том, какие компетенции по выбранной профессии в цене у работодателя;
2. Можно определить, в организации какой формы для вас предпочтительнее работать;
3. Можно определить, в каком направлении следует развивать свои знания и навыки;

Основой эффективности практики является самостоятельная и индивидуальная работа студентов. Практика предусматривает решением определенных задач и выполнение индивидуального задания.

Выполнение индивидуального задания является важнейшим этапом прохождения практики, развивающим самостоятельность в работе, расширяющим кругозор и позволяющим применить полученные в учебном заведении теоретические знания для решения конкретных задач.

Мое индивидуальное задание на эту практику заключается в разработке информационной системы, которая позволит автоматизировать процесс в организации, который в данный момент выполняется вручную или занимает много рабочего времени.

В этом году я проходил практику в МДОУ Новодеревенский детский сад «Колокольчик» в период с 03.04.2023 по 19.04.2023.

1 ОПИСАНИЕ ОРГАНИЗАЦИИ

Полное наименование: Муниципальное дошкольное образовательное учреждение Новодеревенский детский сад "Колокольчик".

Сокращенное наименование: Новодеревенский детский сад "Колокольчик".

Дата создания организации: 1991 год.

Предметом деятельности организации является воспитание, обучение, присмотр, уход и оздоровление детей в возрасте от 2 месяцев до 8 лет.

Муниципальное дошкольное образовательное учреждение Новодеревенский детский сад "Колокольчик" учреждено на основании решения Касимовского районного совета народных депутатов от 20.08.1991 года №199\1. Детский сад открыл свои двери для детей 16 сентября 1991 года.

Учредители: Учредителем Новодеревенского детского сада "Колокольчик" является муниципальное образование - Касимовский муниципальный район Рязанской области.

Функции и полномочия Учредителя и собственника имущества осуществляет управление образования и молодежной политики администрации Касимовского муниципального района Рязанской области.

Вышестоящей организацией является управление образования и молодежной политики администрации Касимовского муниципального района Рязанской области (адрес местонахождения: 391300, г. Касимов, ул. 50 лет СССР, д. 24, тел: 2-24-06, 2-46-37).

Под организационной структурой организации понимаются состав, соподчиненность, взаимодействие и распределение работ по подразделениям и органам управления, между которыми устанавливаются определенные отношения по поводу реализации властных полномочий, потоков команд и информации.

Организационная структура МДОУ Новодеревенского детского сада «Колокольчик» представлена на рисунке 1.

Управление Муниципальным дошкольным образовательным учреждением Новодеревенским детским садом "Колокольчик" строится на принципах единоначалия и самоуправления, обеспечивающих государственно-общественный характер управления ДООУ.

Непосредственное руководство и управление ДООУ осуществляет прошедший соответствующую аттестацию заведующий Молостова Наталья Анатольевна.

Управление представлено в виде 2 основных структур: административного и коллегиального управления. Деятельность структур регламентируется Уставом ДООУ и соответствующими положениями. Организационная структура управления детским садом представляет собой совокупность всех его органов с присущими им функциями.



Рисунок 1 - Структура организации

Управление представлено в виде 2 основных структур: административного и коллегиального управления. Деятельность структур регламентируется Уставом ДОУ и соответствующими положениями. Организационная структура управления детским садом представляет собой совокупность всех его органов с присущими им функциями.

I структура – коллегиальные органы управления:

Общее собрание трудового коллектива осуществляет общее руководство ДОУ, представляет полномочия трудового коллектива, содействует осуществлению управленческих начал, реализует право на самостоятельность ДОУ в решении вопросов, способствующих оптимальной организации образовательного процесса и финансово-хозяйственной деятельности.

Педагогический совет осуществляет управление педагогической деятельностью. Вопросы его компетенции определены Уставом и Положение о педагогическом совете. В соответствии со ст. 63 – 64 Федерального закона от 29.12.2013 № 273-ФЗ «Об образовании в Российской Федерации» дошкольное образование является уровнем общего образования и направлено на формирование общей культуры, развитие физических, интеллектуальных, нравственных, эстетических и личностных качеств, формирование предпосылок учебной деятельности, сохранение и укрепление здоровья детей дошкольного возраста.

Родительский комитет- коллегиальный орган самоуправления учреждения, действующий в целях развития и совершенствования образовательного и воспитательного

процесса, взаимодействия родительской общественности и Учреждения. Положение о родительском комитете.

II структура:

1. заведующие филиалами
2. заместитель заведующего по АХЧ.

Новодеревенский детский сад "Колокольчик" имеет филиалы:

Филиал Ахматовский детский сад. Местонахождение: Рязанская область Касимовский район д. Ахматово ул. Калинина д.7.

Непосредственное руководство деятельностью филиала Ахматовский детский сад осуществляет назначенный на должность заведующий филиалом Цинкина Ирина Викторовна.

Филиал Крюковский детский сад. Местонахождение: Рязанская область Касимовский район д. Крюково д.137.

Непосредственное руководство деятельностью филиала Крюковский детский сад осуществляет назначенный на должность заведующий филиалом Бородачева Надежда Васильевна.

Один из основных процессов, который происходит в данной организации, - это наблюдение за детьми в течение рабочего дня. Этот процесс включает в себя несколько этапов, включая:

1. Утренний прием детей в детский сад;
2. Обеспечение детей комфортным и безопасным пребыванием в детском саду;
3. Организация питания, сна и других необходимых мероприятий;
4. Проведение занятий и игр, направленных на развитие различных навыков и умений у детей.
5. Выдача детей родителям после окончания рабочего дня.

Важный аспект в наблюдении является учет посещаемости детей. Регулярный и точный учет посещаемости позволяет работникам детского сада контролировать количество детей, находящихся в группе, и принимать соответствующие меры для обеспечения их безопасности. Учет посещаемости также является важным фактором для организации питания в детском саду. Точность и своевременность учета позволяет работникам детского сада правильно спланировать количество продуктов и блюд, необходимых для приготовления питания на каждый день.

2 ОПИСАНИЕ КОМПЬЮТЕРНОЙ ИНФРАСТРУКТУРЫ ОРГАНИЗАЦИИ

2.1 Аппаратная часть

В детском саду "Колокольчик" используются технические инфокоммуникационные устройства для обеспечения эффективной работы, обеспечения безопасности и комфорта детей.

Аппаратная часть состоит из компьютеров, ноутбуков, периферийного оборудования, мультимедийных проекторов и других устройств, которые используются для проведения занятий с детьми, просмотра образовательных материалов и контроля за безопасностью детей.

Большая часть аппаратная средств закупалась за безналичный расчёт у компании «Элеком».

Основные аппаратные средства, которые используются в детском саду «Колокольчик»:

1. 3 Персональных компьютера на базе Windows 10. Имеют средние характеристики, которые позволяют решать поставленные задачи. Находятся в распоряжении Заведующей, Делопроизводителя и Завхоза.
2. 6 Ноутбуков на базе Windows 10. В распоряжении Педагогов и Воспитателей. Служат для проведения занятий и ведения документации.
3. 1 Сервер Supermicro. Использовался до перехода на информационную систему БАРС.Web-Образование. Возможности сервера ограничиваются ведением небольшой базы данных, так как он имеет 16 гб оперативной памяти и два процессора Intel Xeon 5530 с частотой 2.40 GHz.
4. Коммутатор PoE HikVision DS-3E0109P-E/M на 8 камер. Используется для объединения компьютеров в локальную сеть.

Проанализировав основные аппаратные средства можно сделать вывод:

1. Компьютеры, используемые в учреждении, в основном имеют средние характеристики и используются для ведения документации и работы с веб-приложениями.
2. Нерациональное использование ресурсов. Сервер Supermicro может использоваться для ведения небольших баз данных, которые могут улучшить деятельность учреждения.

2.2 Программная часть

В детском саду используется специальное программное обеспечение, которое помогает сотрудникам решать различные задачи и автоматизировать процессы, связанные с учетом, организацией питания, воспитанием и обучением детей. Это программное обеспечение значительно облегчает работу сотрудников детского сада и позволяет им более эффективно управлять рутинными задачами.

Основные программные средства и сайты, которые используются в детском саду «Колокольчик»:

1. zakupki.gov.ru – Это единое информационное пространство всей сферы государственных закупок в России. Используется для закупки товаров и услуг за счет бюджетных средств;
2. БАРС.Web-Образование – Это информационная система, которая автоматизирует процессы управления дошкольными образовательными учреждениями (ДОУ) и является важнейшим инструментом учета их комплектования;
3. ФГИС «Меркурий» – Это АИС предназначена для электронной сертификации и обеспечения прослеживаемости поднадзорных государственному ветеринарному надзору грузов при их производстве, обороте и перемещении по территории Российской Федерации. Если дошкольное учреждение закупает продукты, оно обязано требовать у поставщика ветеринарный документ и гасить его при поступлении продовольствия, подлежащего сертификации;
4. ГИС «Энергоэффективность» – государственная информационная система в области энергосбережения и повышения энергетической эффективности;
5. Программный продукт «Вижен-Софт:Питание - предназначен для автоматизации планирования и формирования рационального питания детей. (в филиалах для планирования и формирования рационального питания, а также для учета продуктов используются «Умные таблицы», разработанные в программе Microsoft Excel);
6. Офисный пакет Microsoft Office;
7. Различные утилиты.

3 ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОГО ЗАДАНИЯ

3.1 Постановка задачи

Одной из главных задач, осуществляемых в ДООУ «Колокольчик» является учет посещаемости детей и заполнение табеля.

Процесс «Заполнение и обработка табеля посещаемости детей» представлен на рисунке 2.

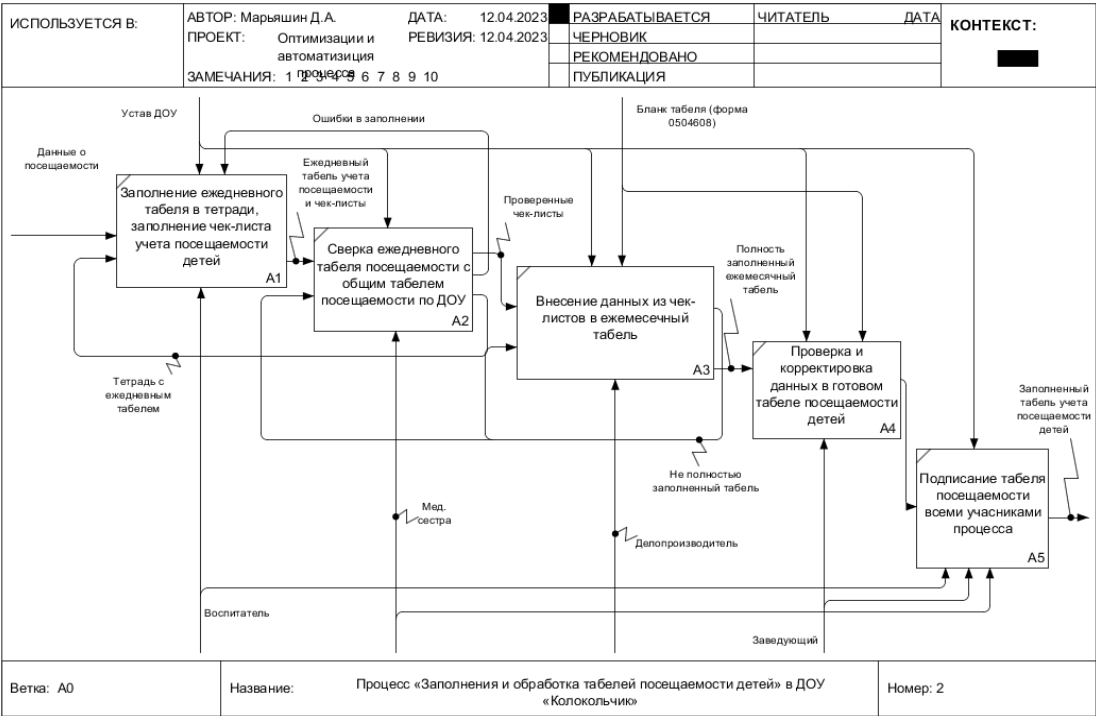


Рисунок 2 - IDEF0 диаграмма процесса

Проанализировав все этапы можно составить сводные таблицы.

На таблице 1 представлены данные о времени, затрачиваемом на каждом этапе процесса.

Таблица 1

Процесс (Из IDEF0 диаграммы)	Затрачиваемое время (мин)
A1	5-10
A2	10-20
A3	10
A4	40-60
A5	5-35
Итого:	70 - 135

Таблица 2

Проблема	Способ решения	Экономия времени(мин)
Допущение неточностей при заполнении ежедневного табеля посещаемости	Создать алгоритм работы по заполнению табеля посещаемости в оцифрованной форме	5 – 10
Длительный период заполнения данных в таблице посещаемости	Создать алгоритм работы по заполнению табеля посещаемости в оцифрованной форме	15 - 35
Утеря или повреждения табеле посещаемости	Создать базу данных для хранения табеля посещения	35-60

Задача: разработать программное обеспечение для оптимизации и автоматизации процесса «Заполнение и обработка табеля посещаемости детей» (Рис. 2) в ДОУ «Колокольчик».

Обоснование: ручное ведение табеля (Рис. 3) посещаемости детей в детском саду может быть трудоемким и неточным процессом. Кроме того, при использовании бумажного табеля может возникать риск его утери или повреждения. Автоматизированный процесс учета посещаемости поможет избежать этих проблем, а также существенно ускорить и упростить процесс формирования табеля посещаемости для педагогических работников.

Утверждена приказом Минфина России от 30.03.2015 N 52н

**ТАБЕЛЬ
УЧЕТА ПОСЕЩАЕМОСТИ ДЕТЕЙ**

на 20 апреля 20 г.

Учреждение Муниципальное дошкольное образовательное учреждение Новодеревянский детский сад "Коловояныч"

Структурное подразделение группа "Гномы" (от 4 до 7 лет)

Вид расчета родительская плата

Режим работы 7.00 - 19.00.

Форма по ОКУД 0504608
Дата 30.08.2021.
по ОКПО 24279550

№	Фамилия, имя ребенка	Номер счета	Плата по ставке	Дни посещения																															Пропущено дней всего	в том числе защита засчитываемых	Дни посещения, подлежащие оплате	Причины неисполнения (основания)
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	36	37	38	39				
1				В	В						В	В			В	В								В	В					В	В							
2				В	В						В	В			В	В								В	В					В	В							
3				В	В						В	В			В	В								В	В					В	В							
4				В	В						В	В			В	В								В	В					В	В							
5				В	В						В	В			В	В								В	В					В	В							
6				В	В						В	В			В	В								В	В					В	В							
			Всего присутствует детей	В	В						В	В			В	В								В	В					В	В							
			Всего отсутствует детей	В	В						В	В			В	В								В	В					В	В							

Руководитель учреждения заведующий Н.А. Молостова Ответственный медицинская сестра

(ответственный исполнитель) (подпись) (расшифровка подписи) исполнитель (должность)

Воспитатель (подпись) (расшифровка подписи) " 30 " апреля 20 23 г. Пяцова В.В.

(подпись) (расшифровка подписи) (подпись) (расшифровка подписи)

Рисунок 3 - Формат табеля (ОКУД 0504608) используемый в учреждении

Требования к программному решению:

- Регистрация детей в системе;
- Учет посещаемости детей по дням;
- Возможность отметить причины отсутствия ребенка;
- Генерация табеля посещаемости в форме ОКУД 0504608 на определенный период времени;
- Возможность просмотра и редактирования данных о посещаемости.

Ожидаемый результат разработки:

- Эффективное и удобное в использовании программное обеспечение для учета посещаемости детей;
- Автоматизированный процесс формирования табеля посещаемости, что позволит существенно сократить время на его ведение;
- Надежное хранение данных о посещаемости детей;
- Возможность просмотра и анализа данных о посещаемости для принятия управленческих решений.

3.2 Проектирование

Для реализации поставленной задачи была выбрана платформа 1С:Предприятие 8.3 (8.3.19.1264).

1С: Предприятие – это специализированная платформа для разработки бизнес-приложений, которая позволяет создавать программы для автоматизации различных процессов.

Помимо множества преимуществ «1С: Предприятие», есть такие, которые стали ключевыми в выборе платформы.

1. Кроссплатформенность. В случае перехода всех бюджетных организаций на «отечественные операционные системы» разработанное программное решение останется актуальным;
2. Возможность внесения изменений на стадии эксплуатации. Если после внедрения возникнет потребность в расширении системы автоматизации, то «1С:Предприятие» позволит добавить новые модули в любом количестве;
3. Быстрота и лёгкость разработки. Платформа предоставляет широкий выбор готовых шаблонов и инструментов, что позволит качественно решить поставленную задачу в период прохождения практики.

3.3 Разработка информационной системы

3.3.1 Разработка концептуальной модели базы данных

3.3.1.1 Определение сущностей

Основным объектом поставленной задачи является табеля посещаемости детей.

Проанализировав его можно выделить следующие сущности:

- Дети;
- Группы;
- Воспитатели;
- Список дней;
- Табели.

3.3.1.2 Определение связей между сущностями

Информационная модель задачи представлена в виде диаграммы «Сущность – связь» на рисунке 4.

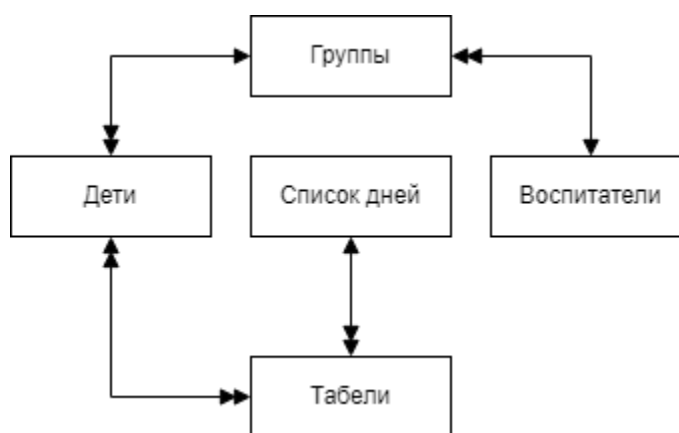


Рисунок 4 - Диаграмма «Сущность – связь»

В данной диаграмме используются следующие связи:

Между сущностями «Дети» и «Табель» используется связь «многие-ко-многим». Это означает, что один ребенок может присутствовать в нескольких табелях, и один табель может содержать несколько детей.

Между сущностями «Список дней» и «Табель» используется связь «один-ко-многим». Это означает, что один список может присутствовать в нескольких табелях, но один табель может содержать только один список.

Между сущностями «Дети» и «Группы» используется связь «один-ко-многим». Это означает, что один ребенок может быть зачислен только в одну группу, но одна группа может включать множество детей.

Между сущностями «Группы» и «Воспитатели» используется связь «один-ко-многим». Это означает, что у группы может быть только один воспитатель, но один воспитатель может руководить несколькими группами.

3.3.1.3 Разработка логической модели базы данных

Концептуальная модель данных состоит из ряда компонентов: сущностей, связей, атрибутов. При переходе к реляционной схеме базы данных каждый из этих компонентов должен быть проанализирован и, если это окажется необходимым, то даже и преобразован. Изменения, вносимые в процессе преобразования, должны быть такими, чтобы их результат полностью отвечал требованиям, выдвигаемым реляционной моделью данных.

Таким образом, фаза логического проектирования предполагает следующие действия:

- преобразование концептуальной модели данных в логическую модель, в результате которого будет определена схема реляционной модели данных;
- проверка модели с помощью концепций последовательной нормализации.

3.3.1.4 Преобразование сущностей

Каждой сущности ставится в соответствие отношение.

Каждый атрибут сущности становится атрибутом отношения, которому приписывают тип данных и свойство допустимости/недопустимости для него значения NULL (не определен).

Первичный ключ сущности становится первичным отношения. Атрибуты, входящие в первичный ключ, получают свойство обязательности и уникальности.

Концептуальная модель содержит пять сущностей. В виду отсутствия связи «один-к-одному» в реляционной модели будет по крайней мере пять отношений.

3.3.1.5 Преобразование связей

Преобразование связи производится согласно значениям ее характеристик: типа связи по мощности (один-ко-многим, один-к-одному, многие-ко-многим) и класса принадлежности.

Класс принадлежности — показатель обязательности участия каждого экземпляра сущности в связи.

Обязательный класс принадлежности связи показывает, что каждый экземпляр сущности ДОЛЖЕН участвовать в связи. Необязательный класс принадлежности допускает наличие экземпляров сущности, не участвующих в связи.

Проанализируем связи между сущностями.

Дети – Табели N:M

Связь N:M будет преобразована за счет введения дополнительной сущности СПИСОК ПРОПУСКОВ, которая должна содержать первичные ключи сущностей ДЕТИ и ТАБЕЛИ (они будут внешними ключами).

Список дней – Табели 1:M

Каждый список обязательно должен быть прикреплен к табелю, т.е. многосвязная сущность Список дней характеризуется обязательным классом принадлежности. Поэтому две сущности будут представлены двумя отношениями.

Дети – Группы 1:M

Каждый ребенок обязательно должен быть оформлен в группу, т.е. многосвязная сущность Дети характеризуется обязательным классом принадлежности. Поэтому две сущности будут представлены двумя отношениями.

Воспитатели – Группы 1:M

На каждую группу обязательно должен быть прикреплен воспитатель, т.е. многосвязная сущность Воспитатели характеризуется обязательным классом принадлежности. Поэтому две сущности будут представлены двумя отношениями.

Таким образом, получим эквивалентный вариант диаграммы «сущность-связь» (рис. 5).

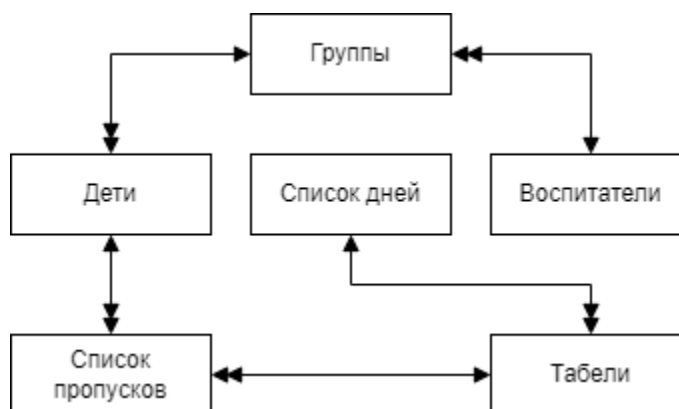


Рисунок 5 - Диаграмма «Сущность – связь»

3.3.1.6 Задание первичных ключей, определение атрибутов сущностей

Для каждой сущности определим атрибуты, которые будут храниться в базе данных.

Результат представлен в табл. 3.1.

Таблица 3.1

Сущность	Первичный ключ	Атрибуты
Дети	Код ребенка	Код ребенка ФИО ребенка Код группы Номер счета Ставка Дата зачисления Дата отчисления
Группы	Код группы	Код группы Наименование группы Возраст Код воспитателя Режим работы
Воспитатели	Код воспитателя	Код воспитателя ФИО воспитателя
Табели	Код табеля	Код табеля Код группы Начало месяца Конец месяца Вид расчета
Список пропусков	Код списка пропусков	Код списка пропусков Код табеля Код ребенка Тип наличия Причина Дата пропуска
Список дней	Код списка дней	Код списка дней Код табеля Дата дня Статус дня

3.3.1.7 Приведение модели базы данных к первой нормальной форме

Модель базы данных находится в первой нормальной форме, если для каждого ее отношения выполняются следующие условия:

- в отношении отсутствуют повторяющиеся атрибуты;

- каждый атрибут отношения неделим.

Согласно требованиям первой нормальной формы, необходимо преобразовать атрибуты «ФИО ребенка», «ФИО воспитателя» в сущностях «Дети», «Воспитатели» чтобы получить неделимые атрибуты. В дальнейшем это позволит реализовать такие функции, например, как поиск информации о детях по фамилии и т.д.

Приведем таблицу сущностей и атрибутов в первой нормальной форме (табл. 3.2).

Таблица 3.2.

Сущность	Первичный ключ	Атрибуты
Дети	Код ребенка	Код ребенка Фамилия ребенка Имя ребенка Отчество ребенка Код группы Номер счета Ставка Дата зачисления Дата отчисления
Группы	Код группы	Код группы Наименование группы Возраст Код воспитателя Режим работы
Воспитатели	Код воспитателя	Код воспитателя Фамилия воспитателя Имя воспитателя Отчество воспитателя
Табели	Код табеля	Код табеля Код группы Начало месяца Конец месяца Вид расчета

Список пропусков	Код списка пропусков	Код списка пропусков Код табеля Код ребенка Тип наличия Причина Дата пропуска
Список дней	Код списка дней	Код списка дней Код табеля Дата дня Статус дня

3.3.1.8 Приведение модели базы данных ко второй нормальной форме

Модель базы данных находится во второй нормальной форме, если для каждого ее отношения выполняются следующие условия:

- отношение находится в первой нормальной формы;
- все поля записи отношения функционально полно зависят от первичного ключа.

Функционально-полная зависимость - это зависимость атрибута от всего составного первичного ключа, а не от его части.

Атрибут «Ставка» в сущности «Дети», атрибут «Режим работы» в сущности «Группы», атрибут «Тип наличия» в сущности «Список пропусков», атрибут «Статус дня» в сущности «Список дней», атрибут «Вид расчета» в сущности «Табель» необходимо выделить в отдельные сущности.

Так же в сущности «Тип наличия» будет атрибут «Тип присутствия», который тоже необходимо выделить в отдельную сущность.

Приведем таблицу сущностей и атрибутов во второй нормальной форме.

Результат представлен в таблице 3.3.

Таблица 3.3

Сущность	Первичный ключ	Атрибуты
Дети	Код ребенка	Код ребенка Фамилия ребенка Имя ребенка Отчество ребенка Код группы Номер счета Ставка Дата зачисления Дата отчисления
Группы	Код группы	Код группы Наименование группы Возраст Код воспитателя Режим работы
Воспитатели	Код воспитателя	Код воспитателя Фамилия воспитателя Имя воспитателя Отчество воспитателя
Табели	Код табеля	Код табеля Код группы Начало месяца Конец месяца Вид расчета
Список пропусков	Код списка пропусков	Код списка пропусков Код табеля Код ребенка Тип наличия Причина Дата пропуска
Список дней	Код списка дней	Код списка дней Код табеля Дата дня Статус дня

Ставка	Код ставки	Код ставки Наименование ставки Сумма ставки
Режим работы	Код режима работы	Код режима работы Наименование режима работы Период режима
Тип наличия	Код типа наличия	Код типа наличия Наименование типа нал Полное наименование Код типа присутствия Оплачивается
Тип присутствия	Код типа присутствия	Код типа присутствия
Статус дня	Код статуса дня	Код статуса дня Наименование статуса дня

3.3.1.9 Приведение модели базы данных к третьей нормальной форме

Условия третьей нормальной формы:

- должны выполняться условия второй нормальной формы;
- внутри каждой сущности должны отсутствовать транзитивные связи.

Разработанная модель уже отвечает требованиям третьей нормальной формы.

Диаграмма взаимосвязей между атрибутами сущностей в третьей нормальной форме представлена на рисунке 6.

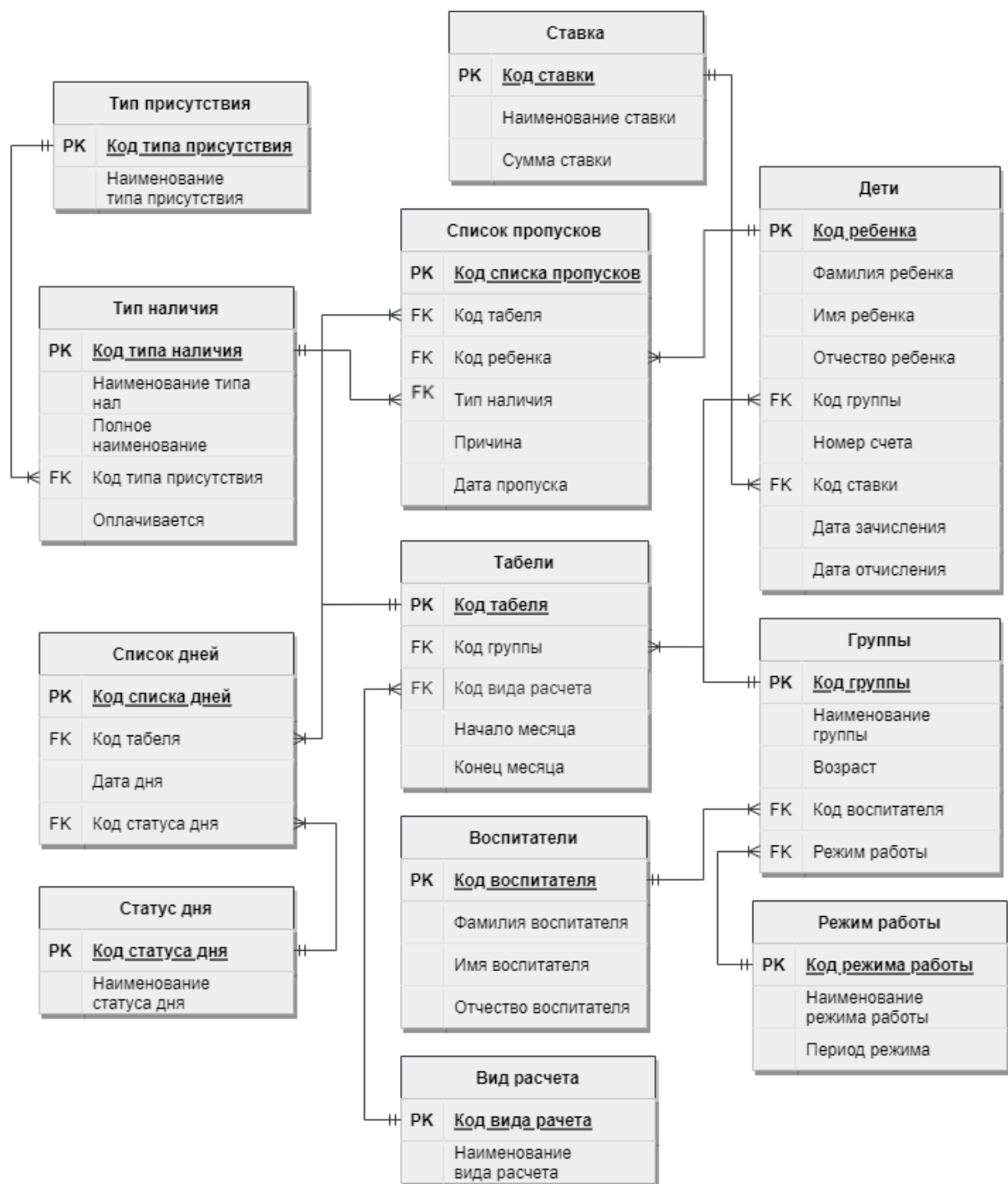


Рисунок 6 - Диаграмма взаимосвязей между атрибутами сущностей в третьей нормальной форме

3.3.2 Разработка физической модели базы данных

Физическая модель базы данных содержит все детали, необходимые конкретной СУБД для создания базы:

- наименования таблиц и полей,
- типы данных,
- определения первичных и внешних ключей,
- ограничения на значения полей.

Физическая модель строится на основе логической с учетом ограничений, накладываемых возможностями выбранной СУБД (в данном случае – 1С: Предприятие).

Для реализации конфигурации необходимо разработать следующие объекты:

1. Справочники;
2. Документы;
3. Перечисления;
4. Регистры сведений.

Модель данных объектов представлена в таблице 4.

Таблица 4

ТипПрисутствия (Перечисление)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодТипаПрисутствия	Строка	Обязательно	—
	Наименование	Строка	Обязательно	—
СтатусДня (Перечисление)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодСтатусаДня	Строка	Обязательно	—
	Наименование	Строка	Обязательно	—
Воспитатели (Справочник)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодВоспитателя	Строка	Обязательно	—
	ФамилияВоспитателя	Строка	Обязательно	—
	ИмяВоспитателя	Строка	Обязательно	—
	ОтчествоВоспитателя	Строка	Обязательно	—
ТипНаличия (Справочник)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодТипаНаличия	Строка	Обязательно	—
	Наименование	Строка	Обязательно	—
	ПолноеНаименование	Строка	Обязательно	—
Внешний	ТипПрисутствия	ПеречислениеСсылка. ТипПрисутствия	Обязательно	—
	Оплачивается	Булево	Необязательно	—
ВидРасчета (Справочник)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодВидаРасчета	Строка	Обязательно	—
	Наименование	Строка	Обязательно	—
Группы (Справочник)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодГруппы	Строка	Обязательно	—
	Наименование	Строка	Обязательно	—
	Возраст	Строка	Обязательно	—
Внешний	Воспитатель	СправочникСсылка. Воспитатели	Обязательно	—
Внешний	РежимРаботы	СправочникСсылка. РежимыРаботы	Обязательно	—

СписокДней (ТабличнаяЧасть)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодСпискаДней	Строка	Обязательно	—
Внешний	ТабельУчетаПосещаемос.	ДокументСсылка. ТабельУчетаПосещаем.	Обязательно	—
	ДатаДня	Дата	Обязательно	—
Внешний	СтатусДня	ПеречислениеСсылка. СтатусДня	Обязательно	—
СтавкаОплаты (Справочник)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодСтавкиОплаты	Строка	Обязательно	—
	Наименование	Строка	Обязательно	—
	Сумма	Число	Обязательно	—
СписокПропусков(ТабличнаяЧасть)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодСпискаПропусков	Строка	Обязательно	—
Внешний	ТабельУчетаПосещаемос.	ДокументСсылка. ТабельУчетаПосещаем.	Обязательно	—
Внешний	Ребенок	СправочникСсылка. Дети	Обязательно	—
Внешний	ТипНаличия	СправочникСсылка. ТипНаличия	Обязательно	—
	ДатаПропуска	Дата	Обязательно	—
	Причина	Строка	Необязательно	—
Дети (Справочник)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодРебенка	Строка	Обязательно	—
	ФамилияРебенка	Строка	Обязательно	—
	ИмяРебенка	Строка	Обязательно	—
	ОтчествоРебенка	Строка	Обязательно	—
Внешний	Группа	СправочникСсылка. Группы	Обязательно	—
	НомерСчета	Строка	Обязательно	—
Внешний	СтавкаОплаты	СправочникСсылка. СтавкаОплаты	Обязательно	—
	ДатаЗачисления	Дата	Обязательно	—
	ДатаОтчисления	Дата	Необязательно	—
РежимРаботы (Справочник)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	КодРежимаРаботы	Строка	Обязательно	—
	Наименование	Строка	Обязательно	—
	Период	Строка	Обязательно	—

Табель Учета Посещаемости (Документ)				
Ключ	Наименование	Тип данных	Обязательность	Комментарий
Первичный	Код Документа	Строка	Обязательно	—
	Номер	Строка	Обязательно	—
Внешний	Группа	Справочник Ссылка. Группы	Обязательно	—
Внешний	Вид Расчета	Справочник Ссылка. Вид Расчета	Обязательно	—
	Начало Месяца	Дата	Обязательно	—
	Конец Месяца	Дата	Обязательно	—

3.3.3 Разработка пользовательского интерфейса

3.3.3.1 Разработка входных форм

Для взаимодействия с базой данных, необходимо разработать входные формы, посредством которых пользователь сможет вносить, изменять и просматривать данные.

Важной особенностью платформы 1С:Предприятие является механизм автогенерируемых форм. Этот механизм освободит от необходимости создания всех возможных форм для каждого из объектов конфигурации.

Необходимо разработать только те выходные формы конфигурации, которые будут содержать программный код для автоматизации процессов.

1. Форма элемента «Дети» (Рис. 7).

Скриншот формы «Дети» в 1С:Предприятии. Вверху заголовок «Смирнов Виктор Владимирович (...)». Кнопки: «Записать и закрыть», «Записать», «Еще...». Поля ввода: Код (000000002), ФИО (Смирнов Виктор Владимирович), Номер счета (X000008344), Плата по ставке (Основная), Дата зачисления (01.04.2022), Дата отчисления (..), Группа (Зайки), Действует с: (23.11.2022). Кнопка «Обновить». Таблица:

Группа	Дата зачисления	Дата отчисления
Зайки	23.11.2022	
Гномы	01.09.2022	23.11.2022

Рисунок 7 - Форма элемента «Дети»

Содержит данные о ребенке, необходимые для заполнения табеля посещаемости.

Поле «ФИО» проходит обязательную проверку перед записью. Проверка осуществляется с помощью кода (Рис. 8).

```

&НаКлиенте
□ Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)
    //Если проверка неудачная отказ в записи
    Массив = Новый Массив();
    Массив.Вставить(0,Объект.Наименование);
    Отказ = ПроверкаФИО(Массив);

    Если Отказ = Истина и Не ЗначениеЗаполнено(Группа.Группа) Тогда //Остановка процедуры
        Возврат;
    КонецЕсли;
    // Заполнение ФИО
    Структура = РазделитьФИО(Объект.Наименование);
    Объект.Наименование = Структура.Фамилия + " " + Структура.Имя + " " + Структура.Отчество;
    //Запись в группу
    ЗаполнитьГруппы(Объект.Ссылка,Булево);
КонецПроцедуры

```

Рисунок 8 - Код проверки ФИО

В проверке участвуют функции из общего модуля (Рис. 9).

```

// Проверка на пустоту
Если НЕ ЗначениеЗаполнено(СтрокаПроверки) Тогда
    Возврат Истина;
КонецЕсли;
// Допустимые символы
КодыДопустимыхСимволов = Новый Массив;
КодыДопустимыхСимволов.Добавить(1105); // "а"
КодыДопустимыхСимволов.Добавить(1025); // "ё"
//Проверка каждого символа в строке на кириллицу
Для Индекс = 1 По СтрДлина(СтрокаПроверки) Цикл
    КодСимвола = КодСимвола(Сред(СтрокаПроверки, Индекс, 1));
    Если ((КодСимвола < 1040) Или (КодСимвола > 1103)) // 1040(А) – 1103(я)
        И (КодыДопустимыхСимволов.Найти(КодСимвола) = Неопределено) Тогда
            Возврат Ложь;
        КонецЕсли;
    КонецЦикла;
Возврат Истина;
КонецФункции

□ функция ПроверкаФИО(ФИО) Экспорт
    Проверка = Ложь;

    Для каждого СтрокаФИО из ФИО Цикл
        // ФИО в массиве из 3 строк
        ЧастиФИО = РазделитьФИО(СтрокаФИО);
        //Проверка строки на кириллицу и пустоту
        Для каждого Строка из ЧастиФИО Цикл
            Если Не ПроверкаКириллица(Строка.Значение) Тогда
                Сообщить("Что то не так с ФИО");
                Возврат Истина;
            КонецЕсли;
        КонецЦикла;
    КонецЦикла;
КонецФункции

```

Рисунок 9 - Функции для проверки ФИО в общем модуле

Поле «Группа» является ссылкой на РегистрСведений «СоставГрупп». При его изменении создается строка в регистре «В пределах дня», что позволяет отслеживать изменения перевода в другую группу. Реализовано с помощью кода, представленного на рисунке 10.


```

&НаСервере
□ Процедура ЗаполнитьГруппы(Ссылка, Булево)
    //На форме существует поле,
    //которое принимает значение истина
    //после изменения поля "Группа"
    Если Булево = Истина Тогда
        //Создание записи в регистре сведений
        МенеджерЗаписи = РегистрыСведений.СоставГрупп.СоздатьМенеджерЗаписи();
        МенеджерЗаписи.Ребенок = Объект.Ссылка;
        МенеджерЗаписи.Группа = Группа.Группа;
        МенеджерЗаписи.Период = ДействуетС;
        МенеджерЗаписи.Записать();
    КонецЕсли;
КонецПроцедуры
&НаКлиенте
□ Процедура ПриОткрытии(Отказ)
    //Заполнение группы при открытии формы
    Структура = ПолучитьГруппу(Объект.Ссылка);
    Если НЕ Структура = Ложь Тогда
        Группа.Группа = Структура.Группа;
        ДействуетС = Структура.Дата;
    КонецЕсли;
КонецПроцедуры
&НаСервере
□ функция ПолучитьГруппу(Ссылка)
    //Запрос на получении данных о группе
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ ПЕРВЫЕ 1
        | СоставГруппСрезПоследних.Группа КАК Группа,
        | СоставГруппСрезПоследних.Период КАК Период
        | ИЗ
        | РегистрСведений.СоставГрупп.СрезПоследних КАК СоставГруппСрезПоследних
        | ГДЕ
        | СоставГруппСрезПоследних.Ребенок.Ссылка = &Ссылка";

    Запрос.УстановитьПараметр("Ссылка", Ссылка);
    РезультатЗапроса = Запрос.Выполнить();
    Структура = Новый Структура("Группа, Дата");
    Если НЕ РезультатЗапроса.Пустой() Тогда

```

Рисунок 10 - Реализованный код для отслеживания группы

Также присутствует Таблица значений «История», которая позволяет узнать историю переводов ребёнка. Код загрузки истории переводов представлен на рисунке 11.

```

&НаСервере
Процедура ЗаполнитьИсторию()
//Очистка таблицы
История.Очистить();
//Запрос Истории переводов
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| СоставГрупп.Группа КАК Группа,
| СоставГрупп.Период КАК Период
| ИЗ
| РегистрСведений.СоставГрупп КАК СоставГрупп
| ГДЕ
| СоставГрупп.Ребенок = &Ребенок
|
| УПОРЯДОЧИТЬ ПО
| Период УБЫВ";

Запрос.УстановитьПараметр("Ребенок", Объект.Ссылка);
РезультатЗапроса = Запрос.Выполнить();
ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
ДатаОтчисл = "";
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
//Датой отчисления является дата перевода в другую группу
НоваяСтрока = История.Добавить();
НоваяСтрока.Группа = ВыборкаДетальныеЗаписи.Группа;
НоваяСтрока.ДатаЗачисления = Строка(Формат(ВыборкаДетальныеЗаписи.Период, "Д#dd.MM.yyyy"));
НоваяСтрока.ДатаОтчисления = Строка(Формат(ДатаОтчисл, "Д#dd.MM.yyyy"));
ДатаОтчисл = ВыборкаДетальныеЗаписи.Период;
КонецЦикла;
КонецПроцедуры
&НаКлиенте
Процедура Обновить(Команда)
ЗаполнитьИсторию();
КонецПроцедуры

```

Рисунок 11 - Код загрузки истории переводов

2. Форма списка «Дети» (Рис. 12).

← →

☆ Дети

🔗 ⓘ ✕

Создать 📄

Поиск (Ctrl+F) ✕ 🔍 Еще ▾

Актуально на дату:

10.04.2023 📅

☑

ФИО	Группа	Номер счета	Дата зачисления	Дата отчисления	Код
Белов Александр Иванович	Зайки	X000003648	30.03.2022		000000011
Григорьева Ирина Анатольевна	Зайки	X000002958	01.04.2022		000000012
Иванова Марина Андреевна	Зайки	X000008454	01.04.2022		000000013
Калинина Любовь Васильевна	Осминожки	X000005961	01.04.2022		000000010
Козлова Ольга Александровна	Гномы	X000008355	01.04.2022		000000003
Лебедева Ксения Сергеевна	Осминожки	X000009841	01.04.2022		000000009
Маркова Анна Сергеевна		X000008321	01.04.2022		000000001
Никитина Нина Алексеевна	Осминожки	X000008336	01.04.2022	17.05.2023	000000006
Попов Денис Андреевич	Гномы	X000008784	16.03.2023		000000004
Романова Анастасия Петровна	Зайки	X000008456	01.04.2022		000000015
Смирнов Виктор Владимирович	Гномы	X000008344	01.04.2022		000000002
Соколова Екатерина Владимировна	Гномы	X000008994	01.09.2022	21.04.2023	000000005
Тихонов Андрей Геннадьевич	Зайки	X000004465	01.04.2022		000000014
Федотова Алина Игоревна	Осминожки	X000008345	01.04.2022		000000008
Шестаков Даниил Викторович	Осминожки	X000008347	01.04.2022		000000007

Рисунок 12 - Форма списка «Дети»

Содержит список всех детей, добавленных в базу данных. Присутствует функция отбора на определённую дату. Условия отбора представлены на рисунке 13. Запрос динамического списка представлен на рисунке 14.

```

&НаКлиенте
□ Процедура Отбор()
    //Проверка параметров отбора
    Если Булево = Истина и Не НаДату = Дата(1,1,1) Тогда
        //Первая группа сортировки
        ПерваяГруппа = Список.Отбор.Элементы.Добавить(Тип("ГруппаЭлементовОтбораКомпоновкиДанных"));
        ПерваяГруппа.ТипГруппы = ТипГруппыЭлементовОтбораКомпоновкиДанных.ГруппаИ;
        //Вторая группа
        ВтораяГруппа = ПерваяГруппа.Элементы.Добавить(Тип("ГруппаЭлементовОтбораКомпоновкиДанных"));
        ВтораяГруппа.ТипГруппы = ТипГруппыЭлементовОтбораКомпоновкиДанных.ГруппаИли;

        //Параметры отбора
        ЭлементОтбора = ПерваяГруппа.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
        ЭлементОтбора.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("ДатаЗачисления");
        ЭлементОтбора.ВидСравнения = ВидСравненияКомпоновкиДанных.МеньшеИлиРавно;
        ЭлементОтбора.Использование = Истина;
        ЭлементОтбора.ПравоеЗначение = НаДату;

        ЭлементОтбора = ВтораяГруппа.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
        ЭлементОтбора.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("ДатаОтчисления");
        ЭлементОтбора.ВидСравнения = ВидСравненияКомпоновкиДанных.БольшеИлиРавно;
        ЭлементОтбора.Использование = Истина;
        ЭлементОтбора.ПравоеЗначение = НаДату;

        ЭлементОтбора = ВтораяГруппа.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
        ЭлементОтбора.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("ДатаОтчисления");
        ЭлементОтбора.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;
        ЭлементОтбора.Использование = Истина;
        ЭлементОтбора.ПравоеЗначение = Дата(1,1,1);

        //В запросе используется регистр сведений СоставГрупп
        //Если ребенок на дату отбора не числится ни в одной группе, то поле "Группа" пустое
        Список.Параметры.УстановитьЗначениеПараметра("Дата", НаДату);
    Иначе
        Список.Параметры.УстановитьЗначениеПараметра("Дата", ТекущаяДата());
        Список.Отбор.Элементы.Очистить();
    КонецЕсли;
КонецПроцедуры

```

Рисунок 13 - Условия отбора

```

ВЫБРАТЬ
    Дети.Ссылка КАК Ссылка,
    Дети.ВерсияДанных КАК ВерсияДанных,
    Дети.ПометкаУдаления КАК ПометкаУдаления,
    Дети.Код КАК Код,
    Дети.Наименование КАК Наименование,
    Дети.НомерСчета КАК НомерСчета,
    Дети.ПлатаПоСтавке КАК ПлатаПоСтавке,
    Дети.ДатаЗачисления КАК ДатаЗачисления,
    Дети.ДатаОтчисления КАК ДатаОтчисления,
    Дети.Предопределенный КАК Предопределенный,
    Дети.ИмяПредопределенныхДанных КАК ИмяПредопределенныхДанных,
    Дети.Представление КАК Представление,
    СоставГруппСрезПоследних.Группа КАК Группа,
    СоставГруппСрезПоследних.Период КАК Период
ИЗ
    РегистрСведений.СоставГрупп.СрезПоследних(&Дата) КАК СоставГруппСрезПоследних
    ПОЛНОЕ СОЕДИНЕНИЕ Справочник.Дети КАК Дети
    ПО СоставГруппСрезПоследних.Ребенок = Дети.Ссылка

```

Рисунок 14 - Запрос динамического списка

3. Форма элемента «Группы» (Рис. 15)

Зачисление	Ребенок
10.04.2023	Смирнов Виктор Владимирович
10.04.2023	Козлова Ольга Александровна
10.04.2023	Попов Денис Андреевич
10.04.2023	Соколова Екатерина Владимировна
13.04.2023	Маркова Анна Сергеевна

Рисунок 15 - Форма элемента «Группы»

Содержит данные о группе и ее составе. Список детей является Динамическим списком и заполняется на основе запроса (Рис. 16).

```
ВЫБРАТЬ
    СоставГруппСрезПоследних.Период КАК Период,
    СоставГруппСрезПоследних.Ребенок КАК Ребенок
ИЗ
    РегистрСведений.СоставГрупп.СрезПоследних КАК СоставГруппСрезПоследних
ГДЕ
    СоставГруппСрезПоследних.Группа = &Группа
```

Рисунок 16 - Запрос динамического списка "Список детей"

4. Форма констант «Общие сведения» (Рис. 17)

Рисунок 17 - Форма констант «Общие сведения»

Содержит данные об учреждении необходимые для формирования табеля.

Поля, содержащие ФИО, проходят обязательную проверку на полноту.

5. Форма документа «Табель Учета Посещаемости» (Рис. 18)

← → ☆ Табель учета посещаемости 000000001 от 10.04.2023 23:53:19

Провести и закрыть Записать Провести Печать

Номер: 000000001 Дата: 10.04.2023 23:53:19 Группа: Зайки

Период: Апрель 2023 г. Всего дней: 30 Рабочих: 19

Заполнить Настроить выходные

Ребенок	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Причины непосещения
Смирнов Виктор	В	В	В					В	В						В	В						В	В						В	В	
Белов Александр	В	В	В					В	В						В	В						В	В						В	В	
Тихонов Андрей	В	В	В					В	В						В	В						В	В						В	В	
Романова Анастасия	В	В	В					В	В						В	В						В	В						В	В	
Григорьева Ирина	В	В	В	х				В	В						В	В						В	В						В	В	
Иванова Марина	В	В	В	х	х	х	х	В	В						В	В						В	В						В	В	
Маркова Анна	В	В	В	х	х	х	х	В	В	х	х		НБ	НБ	В	В	НБ					В	В					В	В	13-17 Простуда	

НУ - Невяка по уважительной причине
 НЯ - Невяка без уважительной причины
 НБ - Невяка по болезни
 В - Выходной
 х - Отчислен

Рисунок 18 - Форма документа «ТабельУчетаПосещаемости»

Это главный документ информационной системы.

Содержит в себе 2 табличные части:

1. Список дней (Рис. 19) – табличная часть, которая содержит список дат и статусы дней.
2. Список пропусков (Рис. 20) – табличная часть, которая содержит пропуски детей с указанием причины.

Таблица значений «Табель» формируется с помощью кнопки «Заполнить». Для изменения периода табеля разработана общая форма «ИзменениеПериода» (Рис.21).

При создании таблицы колонки выходных дней заполняются автоматически, в зависимости от дня недели даты. При необходимости у пользователя есть возможность настроить выходные дни по кнопке «Настроить выходные». После нажатия открывается общая форма «Изменение выходных» (Рис. 22).

N	Дата пропуска	Ребенок	Тип наличия	Причина
1	13.04.2023	Маркова Анна Сергеевна	НБ	13-17 Простуда
2	14.04.2023	Маркова Анна Сергеевна	НБ	13-17 Простуда
3	17.04.2023	Маркова Анна Сергеевна	НБ	13-17 Простуда

Рисунок 19 - Табличная часть Список пропусков

N	Дата	Статус дня
1	01.04.2023	Выходной
2	02.04.2023	Выходной
3	03.04.2023	Выходной
4	04.04.2023	Рабочий
5	05.04.2023	Рабочий
6	06.04.2023	Рабочий
7	07.04.2023	Рабочий
8	08.04.2023	Выходной
9	09.04.2023	Выходной
10	10.04.2023	Рабочий
11	11.04.2023	Рабочий
12	12.04.2023	Рабочий
13	13.04.2023	Рабочий
14	14.04.2023	Рабочий
15	15.04.2023	Рабочий

Рисунок 20 - Табличная часть Список дней

Изменение периода

Год:
Месяц:

Вы

Январь

Февраль

Март

Апрель

Май

Июнь

Июль

Август

Сентябрь

Октябрь

Ноябрь

Декабрь

Рисунок 21 - Общая форма «ИзменениеПериода»

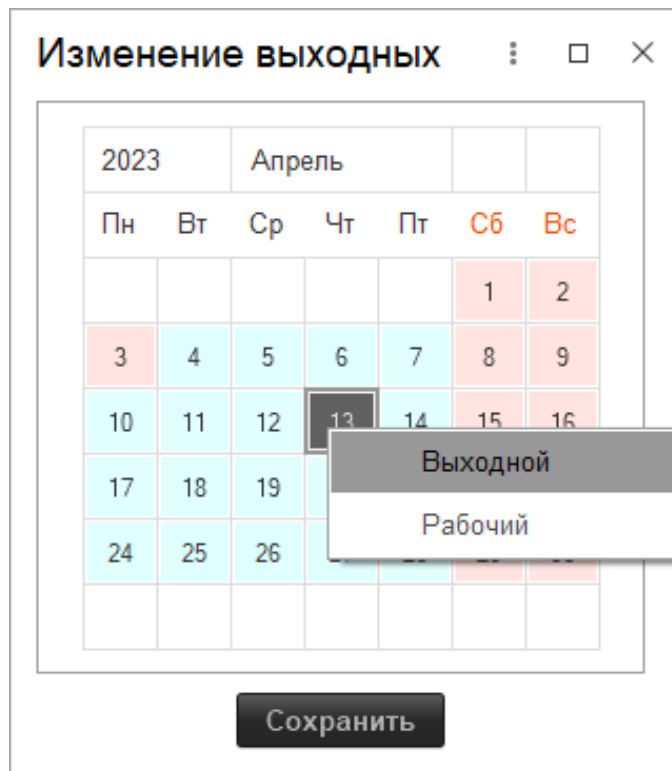


Рисунок 22 - Общая форма «Изменение выходных»

Таблица «Табель» заполняется поэтапно:

1. Настройка количества дней (Рис. 23). Заполняются в зависимости от месяца и года.

```

НаСервере
□ Процедура ЗаполнитьТабельДни(ВсеГодней, Дата)
    Элементы["ТабельРебенок"].ТолькоПросмотр = Истина;
    // 31 - количество столбцов (Дней) в таблице
    Для День = 1 По 31 Цикл
        //Всего дней - количество дней в месяце
        //Если день не содержится в месяце, то его не видно
        Если День > ВсеГодней Тогда
            Элементы.Табель.ПодчиненныеЭлементы["ТабельДень"+День].Видимость = Ложь;
        Иначе
            Элементы.Табель.ПодчиненныеЭлементы["ТабельДень"+День].Видимость = Истина;
            //формирование даты на день
            ДатаДня = Дата (Год(Дата), Месяц(Дата), День);
            //Получение статуса дня
            Для каждого Строка из Объект.СписокДней Цикл
                Если ДатаДня = Строка.Дата Тогда
                    СтатусДня = Строка.СтатусДня;
                КонецЕсли;
            КонецЦикла;
            // Настройка столбцов и списков выбора
            Если СтатусДня = Перечисления.СтатусДня.Рабочий Тогда
                Элементы["ТабельДень"+День].ТолькоПросмотр = Ложь;
                Элементы["ТабельДень"+День].ЦветТекстаЗаголовка = ЦветаСтиля.ЦветТекстаПоля;
                Элементы["ТабельДень"+День].СписокВыбора.Очистить();
                ВыборкаТипыОтсутствия = ПолучитьТипыОтсутствия();

                Индекс = 0;
                Пока ВыборкаТипыОтсутствия.Следующий() Цикл
                    Элементы["ТабельДень"+День].СписокВыбора.Вставить(Индекс, ВыборкаТипыОтсутствия.Ссылка,
                        Строка(ВыборкаТипыОтсутствия.Ссылка) + " (" + Строка(ВыборкаТипыОтсутствия.ПолноеНаименование) + ")");
                    Индекс = Индекс+1;
                КонецЦикла;
            Иначе
                Элементы["ТабельДень"+День].ТолькоПросмотр = Истина;
                Элементы["ТабельДень"+День].ЦветТекстаЗаголовка = ЦветаСтиля.Выходной;
                Элементы["ТабельДень"+День].СписокВыбора.Очистить();
                Выходной = ПолучитьВыходной();
                Элементы["ТабельДень"+День].СписокВыбора.Вставить(0, Выходной);
                Элементы["ТабельДень"+День].ЦветТекста = ЦветаСтиля.Выходной;
            КонецЕсли;
        КонецЕсли;
    КонецЦикла;
КонецПроцедуры

```

Рисунок 23 - Настройка количества дней

2. Заполнение списка детей (Рис. 24). Заполняются с определенным отбором, который позволяет получить список детей, состоящих в группе на каждый день месяца.

```

□ Процедура ЗаполнитьТабельДети(Группа, НачалоМ, КонечМ)
    Для День = 1 По ВсегоДней Цикл
        ДатаДня = Дата(Год(НачалоМ), Месяц(НачалоМ), День);
        //СрезПоследних (Список детей группы на дату дня)
        Запрос = Новый Запрос;
        Запрос.Текст =
            "ВЫБРАТЬ РАЗЛИЧНЫЕ
             | Дети.Ссылка КАК Ссылка
             |
             | ИЗ
             |     РегистрСведений.СоставГрупп.СрезПоследних(&Дата, ) КАК СоставГруппСрезПоследних
             |     ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Дети КАК Дети
             |     ПО СоставГруппСрезПоследних.Ребенок = Дети.Ссылка
             |
             | ГДЕ
             |     Дети.ДатаЗачисления <= &ДатаКонца
             |     И (Дети.ДатаОтчисления > &ДатаНачала
             |         ИЛИ Дети.ДатаОтчисления = &Пусто)
             |     И СоставГруппСрезПоследних.Группа = &Группа";
        Запрос.УстановитьПараметр("Группа", Группа);
        Запрос.УстановитьПараметр("ДатаКонца", ДатаДня);
        Запрос.УстановитьПараметр("ДатаНачала", ДатаДня);
        Запрос.УстановитьПараметр("Пусто", Дата(1,1,1));
        Запрос.УстановитьПараметр("Дата", ДатаДня);
        РезультатЗапроса = Запрос.Выполнить();

        ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

        Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
            //Условия, которые позволяют не дублировать строки
            Если Не Табель.Количество() = 0 Тогда
                Строки = Табель.НайтиСтроки( Новый Структура("Ребенок", ВыборкаДетальныеЗаписи.Ссылка));
                Если Строки.Количество() = 0 Тогда
                    ДобавитьСтроку(Табель, ВыборкаДетальныеЗаписи);
                КонечЕсли;
            Иначе
                ДобавитьСтроку(Табель, ВыборкаДетальныеЗаписи);
            КонечЕсли;
        КонечЦикла;
    КонечЦикла;
КонецПроцедуры

```

Рисунок 24 - Заполнение списка детей

3. Заполнение данных об отчисленных или еще не зачисленных детях в детский сад (Рис. 25).

```

&НаСервере
□ Процедура ЗаполнитьТабельСтолбцыОтчисление(Дата)
    Если Не Табель.Количество() = 0 Тогда
        Для День = 1 По ВсегоДней Цикл
            ДеньСтолбец = "День" + День;
            ДатаДня = Дата(Год(Дата), Месяц(Дата), День);
            Для каждого Строка из Табель Цикл
                Запрос = Новый Запрос;
                Запрос.Текст =
                    "ВЫБРАТЬ
                     | Дети.ДатаЗачисления КАК ДатаЗачисления,
                     | Дети.ДатаОтчисления КАК ДатаОтчисления,
                     | Дети.Ссылка КАК Ссылка
                     |
                     | ИЗ
                     |     Справочник.Дети КАК Дети
                     |
                     | ГДЕ
                     |     Дети.Ссылка = &Ссылка";
                Запрос.УстановитьПараметр("Ссылка", Строка.Ребенок);
                //Если в данный день ребенок не зачислен или отчислен то
                //Ставится тип присутствия "Отчислен"
                ВыборкаРезультатаЗапроса = Запрос.Выполнить().Выбрать();
                ВыборкаРезультатаЗапроса.Следующий();
                Если Не ВыборкаРезультатаЗапроса.ДатаЗачисления = Дата(1,1,1) Тогда
                    Если ВыборкаРезультатаЗапроса.ДатаЗачисления >= ДатаДня Тогда
                        Строка[ДеньСтолбец] = Справочники.ТипНаличия.НайтиПоРеквизиту
                            ("ТипПрисутствия", Перечисления.ТипПрисутствия.Отчислен);
                    КонечЕсли;
                КонечЕсли;
                Если Не ВыборкаРезультатаЗапроса.ДатаОтчисления = Дата(1,1,1) Тогда
                    Если ВыборкаРезультатаЗапроса.ДатаОтчисления <= ДатаДня Тогда
                        Строка[ДеньСтолбец] = Справочники.ТипНаличия.НайтиПоРеквизиту
                            ("ТипПрисутствия", Перечисления.ТипПрисутствия.Отчислен);
                    КонечЕсли;
                КонечЕсли;
            КонечЦикла;
        КонечЦикла;
    КонечЕсли;
КонецПроцедуры

```

Рисунок 25 - Заполнение отчисленных

4. Заполнение данных об отчисленных или еще не зачисленных детях в группу (Рис. 25).

```

НаСервере
Процедура ЗаполнитьТабельСтолбцыГруппаНеЧисл(Дата)
Если Не Табель.Количество() = 0 Тогда
    Для День = 1 По ВсегоДней Цикл
        ДеньСтолбец = "День" + День;
        ДатаДня = Дата(Год(Дата), Месяц(Дата), День);
        Для каждого Строка из Табель Цикл
            Запрос = Новый Запрос;
            Запрос.Текст =
                "ВЫБРАТЬ
                 | СоставГруппСрезПоследних.Ребенок КАК Ребенок
                 | ИЗ
                 | РегистрСведений.СоставГрупп.СрезПоследних(&ДатаСреза, )
                 | КАК СоставГруппСрезПоследних
                 | ГДЕ
                 | СоставГруппСрезПоследних.Ребенок.Ссылка = &Ссылка
                 | И СоставГруппСрезПоследних.Группа = &Группа";

            Запрос.УстановитьПараметр("Ссылка", Строка.Ребенок);
            Запрос.УстановитьПараметр("ДатаСреза", ДатаДня);
            Запрос.УстановитьПараметр("Группа", Объект.Группа);
            Результат = Запрос.Выполнить();
            //Если на данный день ребенок не числится, тогда ТипНаличия Отчислен
            Если Результат.Пустой() Тогда
                Строка[ДеньСтолбец] = Справочники.ТипНаличия.НайтиПоРеквизиту
                (["ТипПрисутствия", Перечисления.ТипПрисутствия.Отчислен]);
            КонецЕсли;
        КонецЦикла;
    КонецЦикла;
КонецЕсли;
КонецПроцедуры

```

Рисунок 26 - Заполнение не зачисленных в группу

5. Заполнение выходных дней с учетом изменений (Рис. 27).

```

НаСервере
Процедура ЗаполнитьТабельСтолбцыВыходные(Дата)
Если Не Табель.Количество() = 0 Тогда
    Для День = 1 По ВсегоДней Цикл
        ДеньСтолбец = "День" + День;
        ДатаДня = Дата(Год(Дата), Месяц(Дата), День);
        //Проверка статуса дня
        Для каждого Строка из Объект.СписокДней Цикл
            Если ДатаДня = Строка.Дата Тогда
                СтатусДня = Строка.СтатусДня;
            КонецЕсли;
        КонецЦикла;
        //Установка выходного
        Для каждого Строка из Табель Цикл
            Если СтатусДня = Перечисления.СтатусДня.Выходной Тогда
                Строка[ДеньСтолбец] = ПолучитьВыходной();
            КонецЕсли;
        КонецЦикла;
    КонецЦикла;
КонецЕсли;
КонецПроцедуры

```

Рисунок 27 - Заполнение выходных дней

6. Заполнение пропусков из таблицы «Список пропусков» (Рис. 28).

```

НаСервере
Процедура ЗаполнитьТабельСтолбцыПропуски(Дата)
Если Не Табель.Количество() = 0 и Не Объект.СписокПропусков.Количество() = 0 Тогда
    Для День = 1 По ВсегоДней Цикл
        ДеньСтолбец = "День" + День;
        ДатаДня = Дата(Год(Дата), Месяц(Дата), День);
        Пропуск = ПолучитьПустуюСсылку();
        Для каждого СтрокаТабель из Табель Цикл
            Для каждого СтрокаПропуски из Объект.СписокПропусков Цикл;
                Если СтрокаПропуски.ДатаПропуска = ДатаДня и
                СтрокаПропуски.Ребенок = СтрокаТабель.Ребенок Тогда
                    СтрокаТабель[ДеньСтолбец] = СтрокаПропуски.ТипНаличия;
                    СтрокаТабель.ПричиныНепосещения = СтрокаПропуски.Причина;
                КонецЕсли;
            КонецЦикла;
        КонецЦикла;
    КонецЦикла;
КонецЕсли;
КонецПроцедуры

```

Рисунок 28 - Заполнение пропусков

Поле «Причина отсутствия» позволяет записывать в документ причины пропусков (Рис. 29).

```

НаКлиенте
Процедура ТабельПричиныНепосещенияНачалоВыбора(Элемент, ДанныеВыбора, СтандартнаяОбработка)
//Отключение стандартной обработки
СтандартнаяОбработка = Ложь;
ТекущиеДанные = Элементы.Табель.ТекущиеДанные;
СтрокиПропусков = Объект.СписокПропусков.НайтиСтроки
(Новый Структура("Ребенок", ТекущиеДанные.Ребенок));
//Если нет пропусков
Если СтрокиПропусков.Количество() = 0 Тогда
Сообщить("Нет данных о пропусках ребенка!");
Возврат;
КонецЕсли;
Причины = ТекущиеДанные.ПричиныНепосещения;
//параметры для записи данных
ДопПараметры = Новый Структура;
ДопПараметры.Вставить("Причины", Причины);
ДопПараметры.Вставить("Ребенок", ТекущиеДанные.Ребенок);
Оповещение = Новый ОписаниеОповещения("ПослеВводаПропусков", ЭтотОбъект, ДопПараметры);
//Открытие окна редактирования (Многострочного)
ПоказатьВводСтроки(Оповещение, Причины, "Введите причины пропусков", Истина);
КонецПроцедуры

НаКлиенте
Процедура ПослеВводаПропусков(Результат, Параметры) Экспорт
Если Результат = Неопределено Тогда
Возврат;
КонецЕсли;
СтрокиПропусковРебенка = Объект.СписокПропусков.НайтиСтроки
(Новый Структура("Ребенок", Параметры.Ребенок));
//Для удобства извлечения данные записываются во все строки
Для каждого Строка из СтрокиПропусковРебенка Цикл
Строка.Причина = Результат;
КонецЦикла;
//Заполнение строки табеля
ТекущиеДанные = Элементы.Табель.ТекущиеДанные;
ТекущиеДанные.ПричиныНепосещения = Результат;
КонецПроцедуры

```

Рисунок 29 - Процедуры записи причины отсутствия

Ячейки, которые содержат тип Выходной и Отчислен (Рис. 30) недоступны для изменения.

```

НаКлиенте
Процедура ТабельПередНачаломИзменения(Элемент, Отказ)
//Получение имени колонки
ИмяКолонки = Элементы.Табель.ТекущийЭлемент.Имя;
//Получение реквизита ячейки
ИмяРеквизита = СтрЗаменить(ИмяКолонки, Элементы.Табель.Имя, "");
//Получение значения
ЗначениеЯчейки = Элементы.Табель.ТекущиеДанные[ИмяРеквизита];
Если ЗначениеЯчейки = ПолучитьОчисл() Тогда
Сообщить("На данную дату ребенок в группе не числится");
Отказ = Истина;
КонецЕсли;
КонецПроцедуры

```

Рисунок 30 - Обработка запрета изменения

3.3.3.2 Разработка выходных форм

Основной выходной формой разработанной системы является «Табель учета посещаемости». Это печатная форма документа «ТабельУчетаПосещаемости», которая формируется на кнопку «Печать».

Макет печатной формы представлен на рисунке 31. Он разработан на основе того, что мне предоставило учреждение и полностью соответствует форме ОКУД 0504608.

несанкционированного доступа к информации, уничтожения или изменения данных, а также для обеспечения надежности и стабильности работы ИС.

Разработанная информационная система содержит множество конфиденциальных данных о детях, таких как их персональные данные, информация об их пребывании в детском саду и другие сведения. Поэтому, в случае отсутствия необходимых мер по защите системы, она может стать уязвимой для несанкционированного доступа, утечки данных или вредоносных атак.

Для защиты данных в разработанной системе присутствует ряд функций:

1. Разграничение доступа. В конфигурации реализованы 2 роли, такие как администратор и пользователь. Каждая из ролей имеет свой набор прав доступа к информации;
2. Резервное копирование данных. Встроенная функция 1С «Выгрузка/Загрузка информационной базы» позволяет в любой момент времени пользователю с правами администратора сделать резервную копию данных;
3. Мониторинг изменений данных. Встроенная функция 1С «Журнал регистраций» позволяет просматривать все действия программы, не только те, которые совершают пользователи, но и все, что сделала сама программа в фоновом режиме, пока пользователи работали;
4. Мониторинг производительности. Встроенная функция 1С «Замер производительности» помогает отслеживать производительность конфигурации, чтобы предотвратить возможные проблемы и отказы.

3.4 Тестирование

Тестирование – это процесс проверки ее функциональности, производительности, надежности и безопасности. Цель тестирования информационной системы состоит в том, чтобы выявить ошибки и недостатки в работе системы и убедиться в ее соответствии заданным требованиям.

Тестирование неотъемлемая часть разработки, потому что любая система, будь то программное обеспечение, веб-приложение или база данных, может содержать ошибки и недостатки, которые могут привести к сбоям, потере данных или даже к критическим нарушениям безопасности. Тестирование позволяет выявить эти проблемы и исправить их до того, как система будет запущена в реальной эксплуатации.

Разработанная конфигурация будет подвергаться 3 видам тестирования:

1. Тестирование и исправление.

Встроенная функция 1С, которая позволяет провести автоматическое тестирование и исправление информационной базы.

Для проведения тестирования необходимо перейти во вкладку Администратор => Тестирование и исправление. Настроить режим (Рис. 33) и «Выполнить».

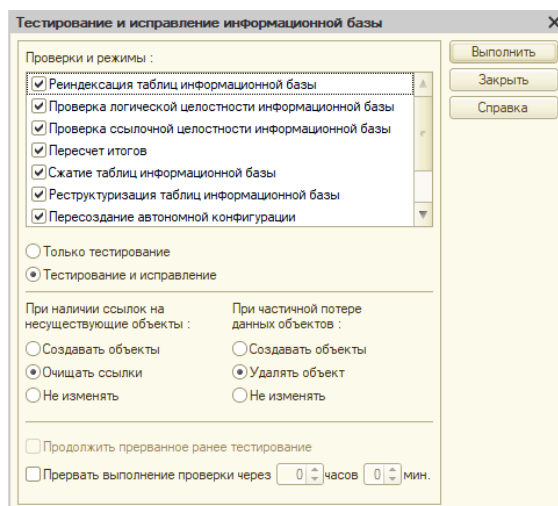


Рисунок 33 - Настройки тестирования

Тестирование прошло успешно (Рис. 34), программа не обнаружила нарушений.

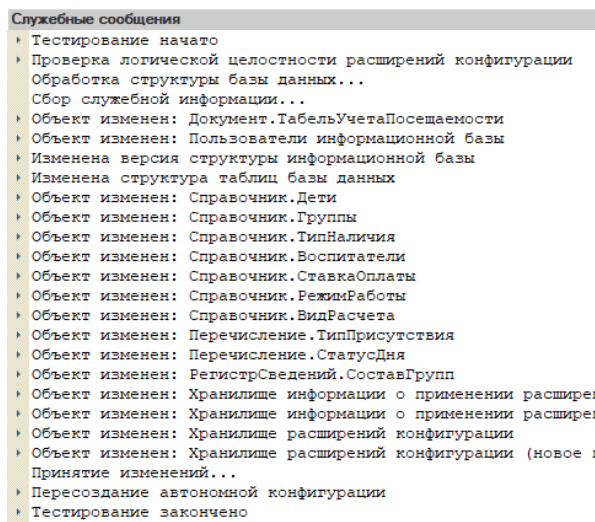


Рисунок 34- Результат тестирования

2. Юнит-тестирование.

Юнит-тестирование - это процесс тестирования программного обеспечения, в котором отдельные компоненты (юниты) тестируются на корректность их работы в изоляции от других компонентов.

Тестируемым модулем был выбран модуль печати документа «ТабельУчетаПосещаемости», поскольку он выполняет важную функцию в системе.

Для тестирования будет использоваться прикладная библиотека OneScript, так как она позволяет создавать и выполнять текстовые сценарии, написанные на языке 1С: Предприятие.

Для использования 1Script необходимо установить следующее:

1. OneScript - Консольное приложение oscript.exe, являющееся основным компонентом системы.
2. Vanessa Runner, Плагин ADD – набор внешних обработок. Помогает быстрее и меньшим кодом писать тесты.

Первым этапом будет создание внешней обработки 1С, в которой будет располагаться юнит – тест. Для этого необходимо на главной панели открыть вкладку Главная => Новый, выбрать Внешняя обработка и нажать «ОК».

Назовем созданную обработку ТестТабельПечать и перейдем в модуль объекта (Рис. 35).

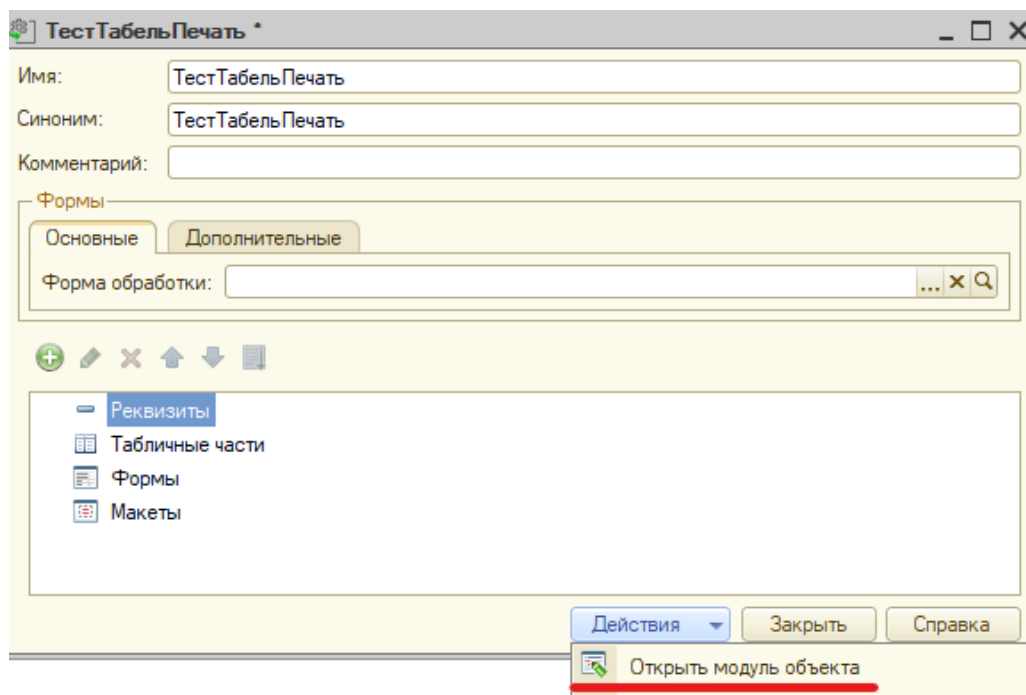


Рисунок 35 - Внешняя обработка для теста

На втором этапе для правильной работы OneScript необходимо создать ряд процедур и переменных, представленных на рисунке 36. Они являются шаблоном для проведения теста.

```

Перем КонтекстЯдра; // обработка из которой запускаются тесты

□ Процедура Инициализация(КонтекстЯдраПараметр) Экспорт
    //Подключение внешних обработок(Планигов) из библиотек Vanessa и ADD
    //Каталог: C:\Program Files (x86)\OneScript\lib\add\plugins
    КонтекстЯдра = КонтекстЯдраПараметр;
КонецПроцедуры

□ Процедура ЗаполнитьНаборТестов(НаборТестов) Экспорт
    //Здесь происходит загрузка тестовых процедур
    НаборТестов.НачатьГруппу("Выполнение теста печати табеля");
КонецПроцедуры

□ Процедура ПередЗапускомТеста() Экспорт
    //Перед запуском начинается транзакция
    //действия с базой данных, выполняемые в рамках транзакции
    //или выполняются целиком, или целиком откатываются
    НачатьТранзакцию();
КонецПроцедуры

□ Процедура ПослеЗапускаТеста() Экспорт
    //Откат всех действий
    Если ТранзакцияАктивна() Тогда
        ОтменитьТранзакцию();
    КонецЕсли;
КонецПроцедуры

```

Рисунок 36 - Шаблон для проведения теста

Для создания юнит – теста на проверку модуля «Печать» потребуются процедура «ПроверитьРавенствоТабличныхДокументовТолькоПоЗначениям» (Рис. 37) из плагина УтвержденияПроверкаТаблиц (Vanessa - ADD).

```

□ Процедура ПроверитьРавенствоТабличныхДокументовТолькоПоЗначениям(ТабДок1, ТабДок2,
    //Проверка таблиц на равенство данных
    УчтыватьТолькоВидимыеКолонкиИлиДопСообщениеОшибки = Ложь, УчтыватьТолькоВидимыеСтрокиИлиДопСообщениеОшибки = Ложь,
    Знач ДопСообщениеОшибки = "", Знач ДопПараметры = Неопределено) Экспорт
    Если ТипЗнч(ТабДок1) <> Тип("ТабличныйДокумент") Тогда
        ВызватьИсключение "ПроверитьРавенствоТабличныхДокументовТолькоПоЗначениям:
        |Первый параметр-таблица не является табличным документов";
    КонецЕсли;
    Если ТипЗнч(ТабДок2) <> Тип("ТабличныйДокумент") Тогда
        ВызватьИсключение "ПроверитьРавенствоТабличныхДокументовТолькоПоЗначениям:
        |Второй параметр-таблица не является табличным документов";
    КонецЕсли;
    УчтыватьТолькоВидимыеКолонки = Ложь;
    Если ТипЗнч(УчтыватьТолькоВидимыеКолонкиИлиДопСообщениеОшибки) = Тип("Булево") Тогда
        УчтыватьТолькоВидимыеКолонки = УчтыватьТолькоВидимыеКолонкиИлиДопСообщениеОшибки;
    ИначеЕсли ТипЗнч(УчтыватьТолькоВидимыеКолонкиИлиДопСообщениеОшибки) = Тип("Строка") Тогда
        ДопСообщениеОшибки = УчтыватьТолькоВидимыеКолонкиИлиДопСообщениеОшибки;
    КонецЕсли;
    УчтыватьТолькоВидимыеСтроки = Ложь;
    Если ТипЗнч(УчтыватьТолькоВидимыеСтрокиИлиДопСообщениеОшибки) = Тип("Булево") Тогда
        УчтыватьТолькоВидимыеСтроки = УчтыватьТолькоВидимыеСтрокиИлиДопСообщениеОшибки;
    ИначеЕсли ТипЗнч(УчтыватьТолькоВидимыеСтрокиИлиДопСообщениеОшибки) = Тип("Строка") Тогда
        ДопСообщениеОшибки = УчтыватьТолькоВидимыеСтрокиИлиДопСообщениеОшибки;
    КонецЕсли;
    Таб1 = ПолучитьТаблицуЗначенийИзТабличногоДокумента(ТабДок1, УчтыватьТолькоВидимыеКолонки, УчтыватьТолькоВидимыеСтроки);
    Таб2 = ПолучитьТаблицуЗначенийИзТабличногоДокумента(ТабДок2, УчтыватьТолькоВидимыеКолонки, УчтыватьТолькоВидимыеСтроки);
    ПроверитьРавенствоТаблиц(Таб1, Таб2, ДопСообщениеОшибки, ДопПараметры);
КонецПроцедуры

```

Рисунок 37 - Функция «СоздатьДанныеПоМакетам»

Данная процедура позволит проверить макет на правильность заполнения ячеек.

← →

☆

Табель учета посещаемости 000000001 от 10.04.2023 23:53:19

🔗 ⓘ ✕

Провести и закрыть

Записать

Провести

Печать

Номер: 000000001

Дата: 10.04.2023 23:53:19

Группа: Зайки

Вид расчета: Родительская плата

Период: Апрель 2023 г.

Всего дней: 30

Рабочих: 19

▶ Заполнить

⚙ Настроить выходные

Ребенок	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Причины непосещеия
Смирнов Виктор	В	В	В					В	В						В	В						В	В						В	В	
Белов Александр	В	В	В					В	В						В	В						В	В						В	В	
Тихонов Андрей	В	В	В					В	В						В	В						В	В						В	В	
Романова Анаст...	В	В	В					В	В						В	В						В	В						В	В	
Григорьева Ирина	В	В	В	х				В	В						В	В						В	В						В	В	
Иванова Марина	В	В	В	х	х	х	х	В	В						В	В						В	В						В	В	
Маркова Анна	В	В	В	х	х	х	х	В	В	х	х		НБ	НБ	В	В	НБ					В	В						В	В	13-17 Простуда

НУ - Невяка по уважительной причине

НБ - Невяка без уважительной причины

НБ - Невяка по болезни

В - Выходной

х - Отчислен

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	Утверждена
2																																													
3																																													
4																																													
5																																													
6																																													
7																																													
8																																													
9																																													
10																																													
11																																													
12																																													
13																																													
14																																													
15																																													
16																																													
17																																													
18																																													
19																																													
20																																													
21																																													
22																																													
23																																													
24																																													
25																																													
26																																													
27																																													
28																																													
29																																													
30																																													
31																																													
32																																													
33																																													
34																																													
35																																													
36																																													
37																																													
38																																													
39																																													
40																																													
41																																													
42																																													
43																																													
44																																													
45																																													

В результате получился макет с ожидаемым результатом. Подсчёты проводились вручную.

40


```

Перем КонтекстЯдра; // обработка из которой запускаются тесты
Перем ПроверкаТаблиц; // модуль проверки таблиц (УтвержденияПроверкаТаблиц)

Процедура Инициализация(КонтекстЯдраПараметр) Экспорт
//Подключение внешних обработок(Планигов) из библиотек Vanessa и ADD
//Каталог: C:\Program Files (x86)\OneScript\lib\add\plugins
КонтекстЯдра = КонтекстЯдраПараметр;
ПроверкаТаблиц = КонтекстЯдра.Плагин("УтвержденияПроверкаТаблиц");
КонецПроцедуры

Процедура ЗаполнитьНаборТестов(НаборТестов) Экспорт
//Здесь происходит загрузка тестовых процедур
НаборТестов.НачатьГруппу("Выполнение теста печати табеля");
НаборТестов.Добавить("Тест_ПечатьТабель"); // Экспортный метод
КонецПроцедуры

//Юнит - Тест
Процедура Тест_ПечатьТабель() Экспорт
// ТабДок заполняется внутри экспортного метода
ТабДок = Новый ТабличныйДокумент;
//Печать документа
Документы.ТабельУчетаПосещаемости.Печать
(ТабДок, Документы.ТабельУчетаПосещаемости.НайтиПоНомеру("000000001").Ссылка);
//Получение макета ожидаемого результата
Эталон = ПолучитьМакет("ОжидаемыйРезультат");
//Сравнение макетов
ПроверкаТаблиц.ПроверитьРавенствоТабличныхДокументовТолькоПоЗначениям
(ТабДок, Эталон,,, "Макеты не равны");
КонецПроцедуры

```

Рисунок 40 - Юнит-Тест

Для запуска теста необходимо запустить внешнюю обработку «xddTestRunner.epf» из каталога «C:\Program Files (x86)\OneScript\lib\add» (Рис. 41). Нажать Загрузить => Тесты из файлов, выбрать внешнюю обработку «ТестТабельПечать».

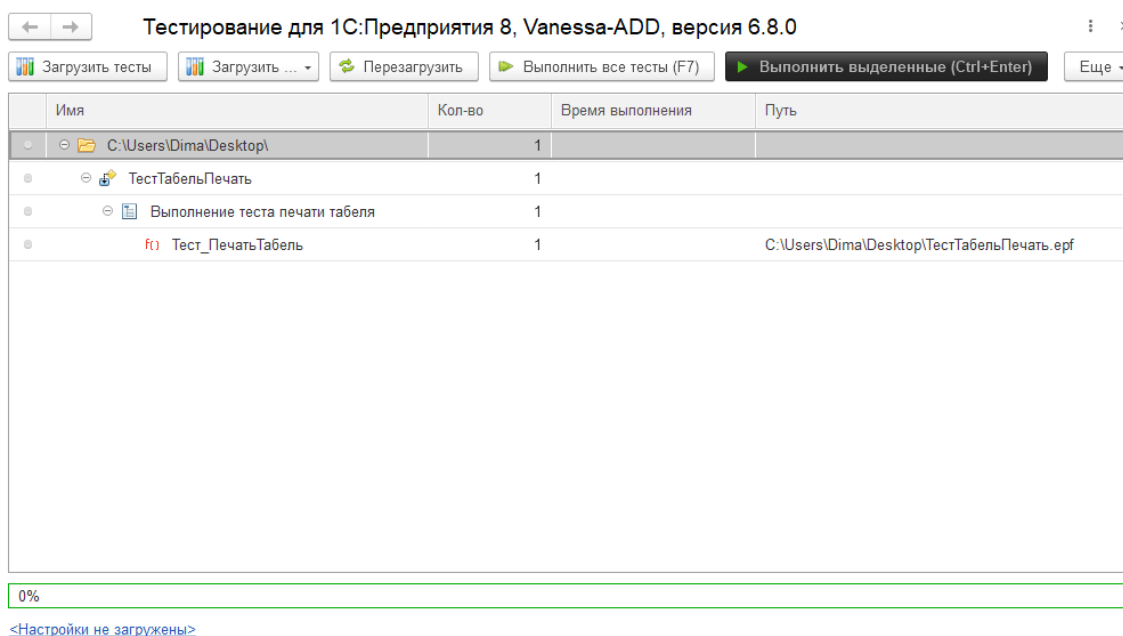


Рисунок 41 - Внешняя обработка для запуска тестов

Нажатие кнопки «Выполнить выделенные» запустит юнит – тест и выведет отчет проверки (Рис. 42).

Запущено тестов - 1, ошибочных тестов - 0, не прошло проверку - 1, пропущено - 0
Общее время выполнения: 0,516 (0:00:00 сек.)

C:\Users\Dima\Desktop\

ТестТабельПечать

Выполнение теста печати табеля

Тест_ПечатьТабель, путь C:\Users\Dima\Desktop\ТестТабельПечать.epf

[Failed] Таблицы должны совпадать, а они различны
Различия в таблицах:
Ожидали статус <ТаблицыСовпадают>, а получили <НеСовпадаютЗначенияВЯчейкеТаблицы>
Значение [K25:22]. Ожидали <7>, а получили <6>

Макеты не равны
{ВнешняяОбработка.УтвержденияПроверкаТаблиц.МодульОбъекта(446)}:ВызватьИсключение Префикс + " " + СообщениеОшибки;
{ВнешняяОбработка.УтвержденияПроверкаТаблиц.МодульОбъекта(42)}:ВызватьОшибкуПроверки("Таблицы должны совпадать, а они
{ВнешняяОбработка.УтвержденияПроверкаТаблиц.МодульОбъекта(75)}:ПроверитьРавенствоТаблиц(Таб1, Таб2, ДопСообщениеОшибки
{ВнешняяОбработка.ТестТабельПечать.МодульОбъекта(28)}:ПроверкаТаблиц.ПроверитьРавенствоТабличныхДокументовТолькоПоЗна
{1}):КонтекстВыполнения.Тест_ПечатьТабель();
{ВнешняяОбработка.xddTestRunner.МодульОбъекта(391)}:Выполнить(ИсполняемыйКод);
{ВнешняяОбработка.xddTestRunner.МодульОбъекта(282)}:ВыполнитьПроцедуруКонтекста(КонтекстВыполнения, ЭлементДереваТест
{ВнешняяОбработка.xddTestRunner.Форма.УправляемаяФорма.Форма(1049)}:Результат.РезультатТестирования = ОбъектНаСервере.

Рисунок 42 - Отчет выполненного юнит-теста

При заполнении макета я специально ошибся (Рис. 43) в ячейке [25:22]. Сделанная ошибка позволила проверить работоспособность разработанного теста.

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
14	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
15	В					В	В						В	В	
16	В					В	В						В	В	
17	В					В	В						В	В	
18	В					В	В						В	В	
19	В	х				В	В						В	В	
20	В	х	х	х	х	В	В						В	В	
21	В	х	х	х	х	В	В	х	х		НБ	НБ	В	В	НБ
22	В	4	5	5	5	В	В	6	6	6	6	6	В	В	6
23	В	3	2	2	2	В	В	1	1	0	1	1	В	В	1
24															

Рисунок 43 - Сделанная ошибка

После исправления ошибки тест был пройден удачно (Рис. 44).

Тестирование для 1С:Предприятия 8, Vanessa-ADD, версия 6.8.0

Загрузить тесты Загрузить ... Перезагрузить Выполнить все тесты (F7) Выполнить выделенные (Ctrl+Enter) Еще

Имя	Кол-во	Время выполнения	Путь
C:\Users\Dima\Desktop\	1	0,516	
ТестТабельПечать	1	0,516	
Выполнение теста печати табеля	1	0,516	
Тест_ПечатьТабель	1	0,516	C:\Users\Dima\Desktop\ТестТабельПечат...

100%

<Настройки не загружены>

Рисунок 44 - Удачный результат теста

3. Автоматизированное тестирование.

Это часть процесса тестирования, которая представляет собой имитацию действий пользователя, а также получение и оценку результатов этих действий.

Для реализации данного вида тестирования необходимо

1. Создать конфигурацию с типом запуска «Менеджер тестирования» (Рис. 45);

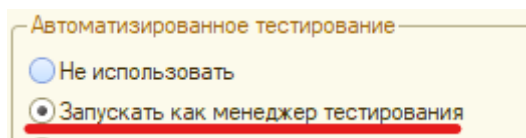


Рисунок 45 - Тип запуска конфигурации

2. Создать форму для проведения теста (Рис. 46);

Рисунок 46 - Форма тестирования

3. Реализовать код для подключения к клиенту тестирования (Рис. 47);

```

<НаКлиенте
[ Процедура ПодключениеКБазе ()
    Если ЗначениеЗаполнено (Платформа) и ЗначениеЗаполнено (ПутьКБазе) Тогда
        ЗапуститьСистему (Платформа + " ENTERPRISE /F " + ПутьКБазе + " /TESTCLIENT");
    Иначе
        Предупреждение ("Заполните все поля!");
    КонецЕсли;
КонецПроцедуры

<НаКлиенте
[ функция УстановкаСоединения ()
    ТестовоеПриложение = Новый ТестируемоеПриложение ();
    ВремяОкончанияОжидания = ТекущаяДата () + 60;
    Подключен = Ложь;
    ОписаниеОшибкиСоединения = "";
    Пока Не ТекущаяДата () >= ВремяОкончанияОжидания Цикл
        Попытка
            ТестовоеПриложение.УстановитьСоединение ();
            Подключен = Истина;
            Возврат ТестовоеПриложение;
            Прервать;
        Исклучение
            ОписаниеОшибкиСоединения = ОписаниеОшибки ();
        КонецПопытки;
    КонецЦикла;
    Если Не Подключен Тогда
        ТестовоеПриложение = Неопределено;
        Сообщить ("Не смогли установить соединение! " + Символы.ПС + ОписаниеОшибкиСоединения);
    КонецЕсли;
Конецфункции

```

Рисунок 47 - Методы подключения клиента тестирования

4. Сгенерировать тест с помощью записи действий пользователя и преобразовать в язык системы 1С. (Рис. 48);

```
НаКлиенте
Процедура ОкноПриложенияОсновноеКнопкаКомандногоИнтерфейсаТабельУчетаПосещаемостиНажать (ТестовоеПриложение)
    ОкноПриложенияОсновное = Неопределено;
    КлиентскиеОкнаТестируемогоПриложения = ТестовоеПриложение.ПолучитьПодчиненныеОбъекты();
    Для Каждого ТестируемоеОкно Из КлиентскиеОкнаТестируемогоПриложения Цикл
        Если ТестируемоеОкно.Основное Тогда
            ОкноПриложенияОсновное = ТестируемоеОкно;
            Прервать;
        КонечЕсли;
    КонечЦикл;
    ОкноПриложенияОсновноеCommandInterface = ОкноПриложенияОсновное.ПолучитьКомандныйИнтерфейс();
    КнопкаКомандногоИнтерфейсаЖурналУчета = ОкноПриложенияОсновноеCommandInterface.НайтиОбъект
        (Тип ("ТестируемаяКнопкаКомандногоИнтерфейса"), "Журнал учета");
    КнопкаКомандногоИнтерфейсаЖурналУчета.Нажать();

    КнопкаКомандногоИнтерфейсаТабельУчетаПосещаемости = ОкноПриложенияОсновноеCommandInterface.НайтиОбъект
        (Тип ("ТестируемаяКнопкаКомандногоИнтерфейса"), "Табель учета посещаемости");
    КнопкаКомандногоИнтерфейсаТабельУчетаПосещаемости.Нажать();
КонечПроцедуры

НаКлиенте
Процедура ОкноПриложенияТабельУчетаПосещаемостиКнопкаСоздатьНажать (ТестовоеПриложение)
    ОкноПриложенияТабельУчетаПосещаемости = ТестовоеПриложение.НайтиОбъект
        (Тип ("ТестируемоеОкноКлиентскогоПриложения"), "Табель учета посещаемости", , 30);
    ОкноПриложенияТабельУчетаПосещаемостиФормаТабельУчетаПосещаемости = ОкноПриложенияТабельУчетаПосещаемости.НайтиОбъект
        (Тип ("ТестируемаяФорма"), "Табель учета посещаемости");
    КнопкаСоздать = ОкноПриложенияТабельУчетаПосещаемостиФормаТабельУчетаПосещаемости.НайтиОбъект (Тип ("ТестируемаяКнопкаФормы"), "Создать");
    КнопкаСоздать.Нажать();
КонечПроцедуры
```

Рисунок 48 - Часть тестового сценария

Для проверки созданного тестового сценария необходимо в поле «Платформа» добавить путь к 1С: Предприятие, а в поле «Путь к базе» указать путь к каталогу клиента тестирования (Рис. 49).

Форма тестирования

Платформа: C:\Program Files\1cv8\8.3.19.1264\bin\1cv8.exe

Путь к базе: C:\Users\Dima\Documents\InfoBase15

Тест табель

Рисунок 49 - Заполненная форма тестирования

После запуска теста произойдет имитация действий пользователя по созданию и заполнению табеля (Рис. 50).

Сообщения:

— Тест пройден!

Рисунок 50 - Результат тестирования

3.5 Внедрение

Внедрение - это процесс настройки и запуска программного продукта разработанной системы в рамках определенного бизнес-процесса организации. Главной целью внедрения является автоматизация бизнес-процессов, увеличение эффективности и оптимизация работы учреждения.

Разработанная конфигурации может внедряться как файловым вариантом, так и клиент - серверным.

Для внедрения в ДООУ «Колокольчик» будет использоваться клиент – серверный вариант, что позволит защитить данные и использовать данную систему в многопользовательском режиме через локальную сеть учреждения.

В качестве системы управления базами данных будет использоваться Microsoft SQL Server, потому что она обладает рядом преимуществ:

- Высокая производительность за счет использования индексирования и секционирования таблиц в СУБД;
- Автоматическое задействование аппаратных ресурсов по мере роста нагрузки;
- Высокая степень надежности за счет технологий обеспечения отказоустойчивости и резервного копирования данных;
- Рациональное использование дискового пространства за счет возможности сжатия данных в базе SQL Server до 50% от исходного объема.

Разработка системы проводилась в файловом варианте, поэтому при внедрении необходимо перенести разработанную базу на сервер SQL.

Для внедрения системы потребуется:

1. Установить SQL Server, SSMS и Сервер 1С, настроить роли и загрузить базу данных;
2. Установить 1С: Предприятие на компьютеры пользователей;
3. Подключиться к 1С Серверу через 1С: Предприятие.
4. Обучить пользователей для работы с конфигурацией.

Для внедрения сформирован и настроен комплект поставки, который включает необходимые файлы для внедрения (Рис. 51).

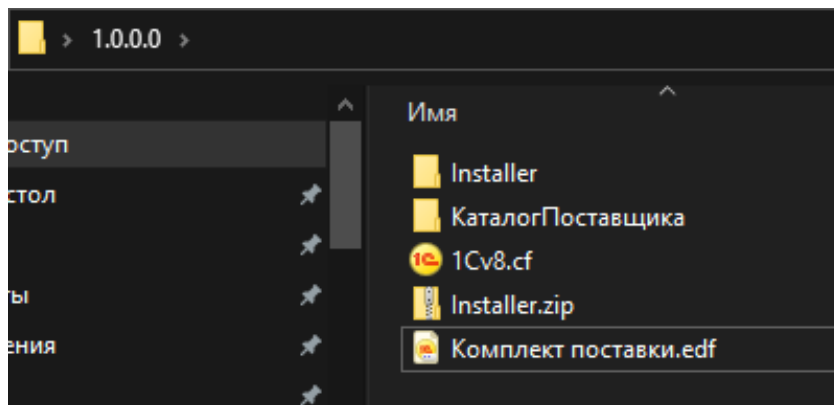


Рисунок 51 - Комплект поставки

Каталог поставщика (Рис. 52) содержит такие файлы как:

1. 1Cv8.cf - шаблон конфигурации (Пустая конфигурация);
2. 1Cv8.dt - выгрузка конфигурации;
3. 1Cv8.mft – манифест (в нём указывается минимальная редакция платформы, с которой совместимо данная система);
4. SQL Server;
5. SSMS.

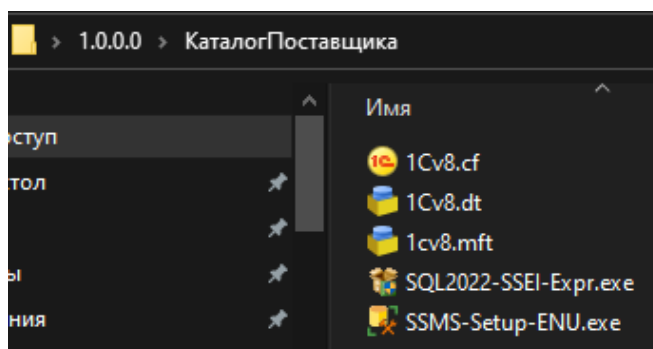


Рисунок 52 - Файлы поставки

В организации для внедрения будет использоваться архив установки, который установит на компьютер пользователя поставляемые файлы (Рис. 53).

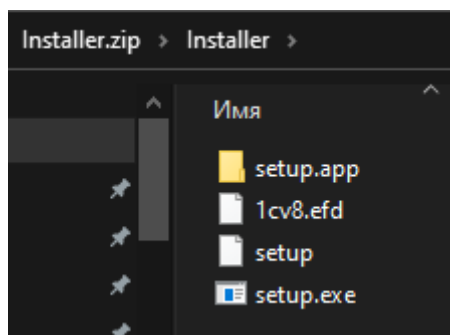


Рисунок 53 - Архив установки

Использование такого способа внедрения (с помощью файла поставки), позволит с легкостью выпускать файл обновления конфигурации.

4 ОБЗОР РАБОТЫ РАЗРАБОТАННОЙ ИС

4.1 Инсталляция

4.1.1 Файловая система

Для установки информационной системы 1С: Предприятие «Учет посещаемости» как «Файловая система» необходимо:

1. Установить 1С: Предприятие скаченное с сайта 1С (Рис. 54). На всех этапах установки есть подсказки;

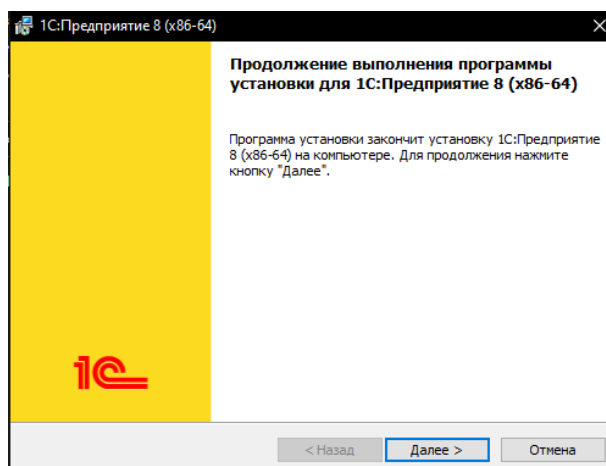


Рисунок 54 - Установка 1С: Предприятие

2. Установить файл поставки (Рис. 55, 56);

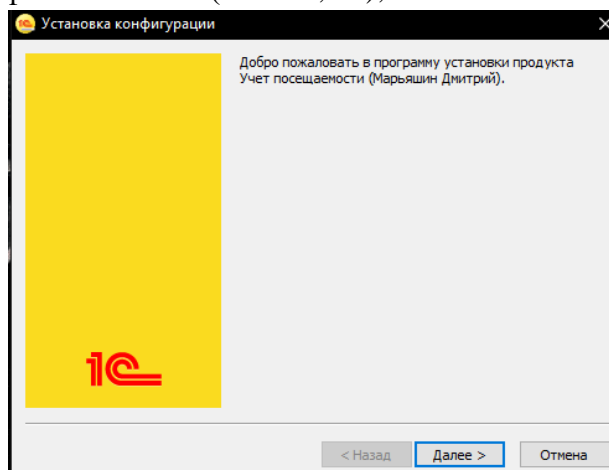


Рисунок 55 - Установка конфигурации

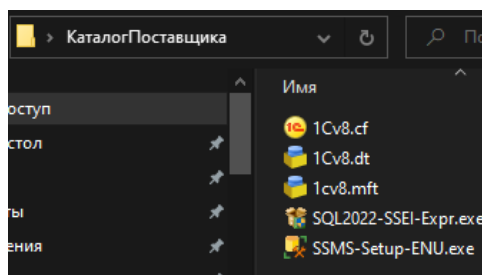


Рисунок 56 - Результат установки

3. Запустить 1С: Предприятие и создать новую информационную базу. Указать имя и каталог где будет храниться конфигурация (Можно указать каталог в локальной сети, что позволит подключаться с нескольких компьютеров) (Рис. 57).

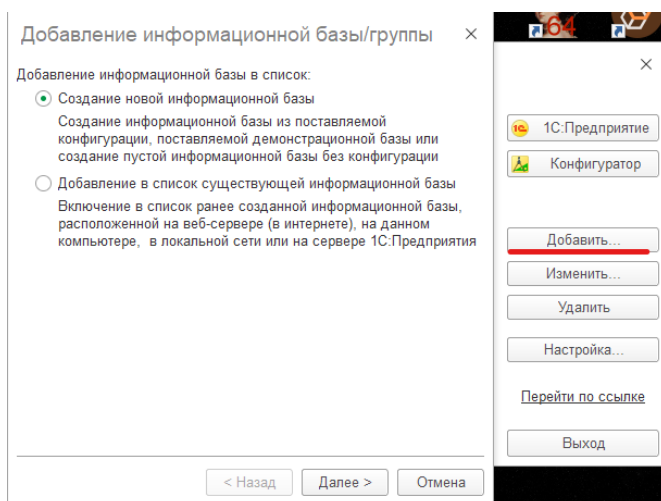


Рисунок 57 - Создание информационной базы

4. Зайти в Конфигуратор (Рис. 58), Конфигурация => Открыть конфигурацию, загрузить конфигурацию из файла (Рис. 59);

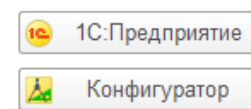


Рисунок 58 - Конфигуратор

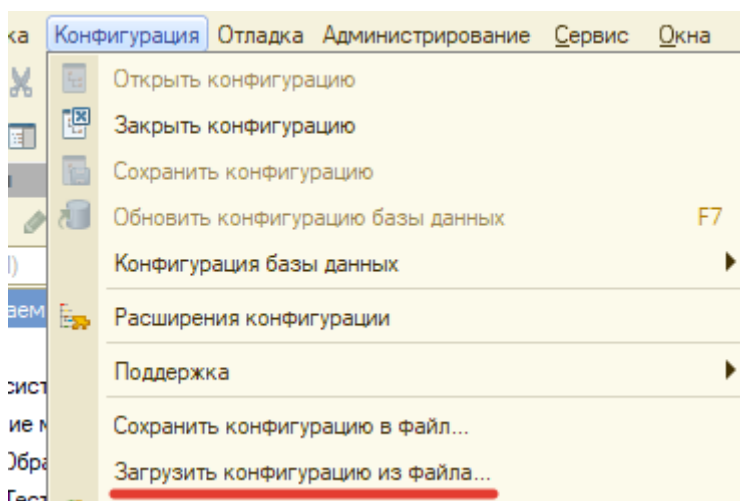


Рисунок 59 - Загрузить информационную базу

5. В открывшемся окне перейти в \Каталог поставщика, выбрать 1Cv8.cf и нажать «Открыть» (Рис. 60), а затем принять (Рис. 61);

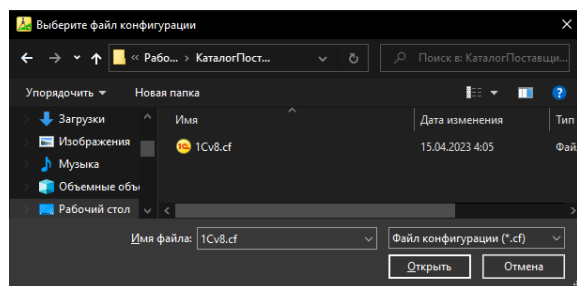


Рисунок 60 - Загрузка системы

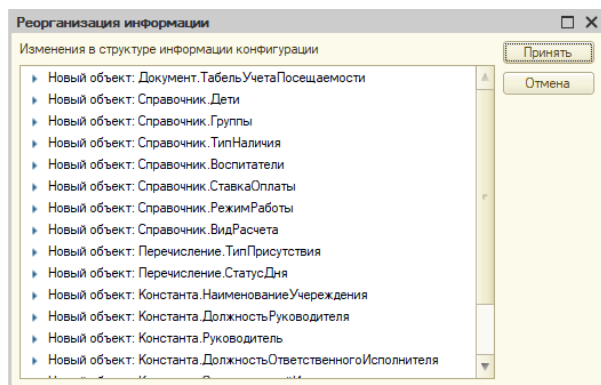


Рисунок 61 - Загрузка информации

6. После перезапуска конфигуратора ИС «Учет посещаемости» установиться на ваш компьютер или в локальную сеть (Рис. 62).

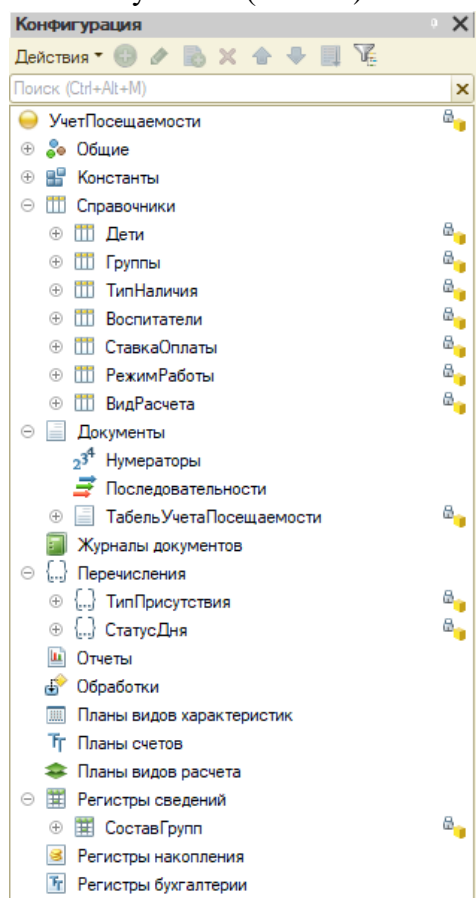


Рисунок 62 - Успешный результат установки

4.1.2 Клиент - Серверная система

Для установки информационной системы 1С: Предприятие «Учет посещаемости» как «Клиент - Серверная система» необходимо:

1. Первые 2 пункта из «Установка Файловой системы»;
2. В «\Каталог поставщика» запустить SQL2022-SSEI-Expr.exe;
3. В мастере установки выбрать Пользовательский тип установки (Рис. 63);

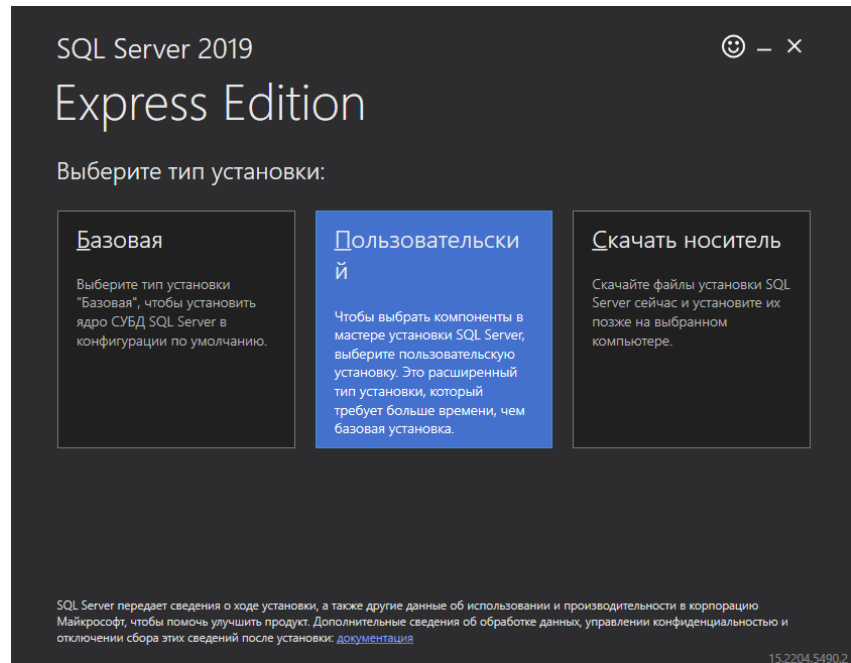


Рисунок 63 - Мастер установки SQL Server

4. Указать расположение для скачивания носителя SQL Server и нажать «Установить» (Рис. 64);

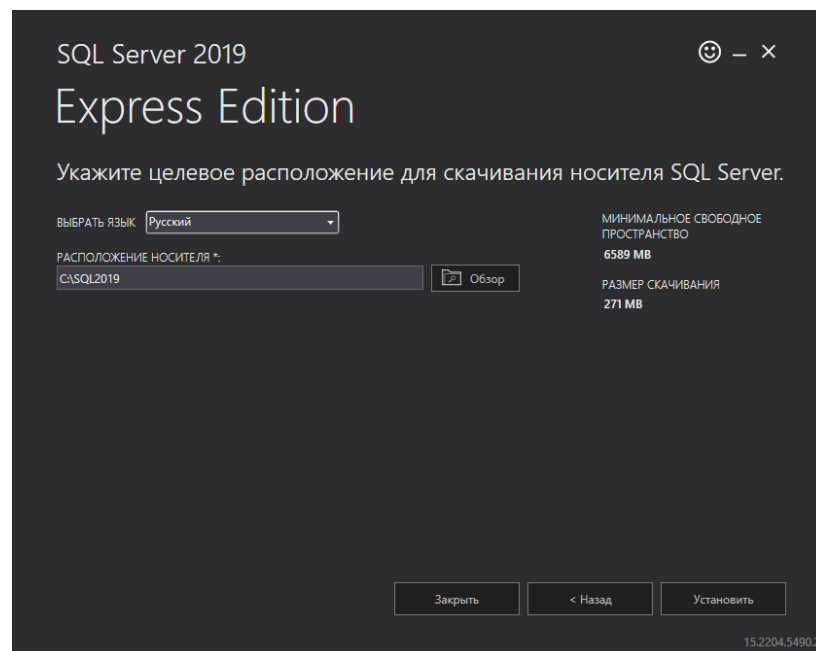


Рисунок 64 - Мастер установки SQL Server

5. После установки носителя SQL Server откроется окно центра установки SQL Server. Необходимо запустить мастер установки SQL Server 2019 (Рис. 65).

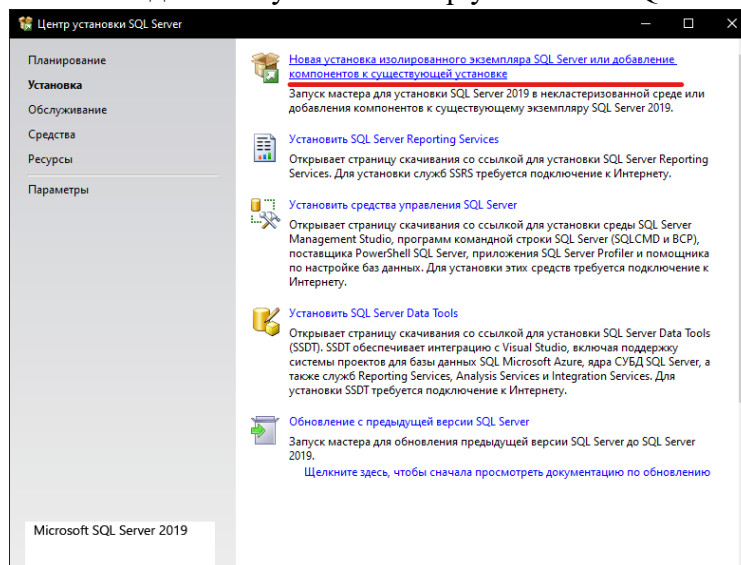


Рисунок 65 - Центр установки SQL Server

6. Следовать рекомендациям до пункта «Выбор компонентов», в данном пункте выбрать как на рисунке 66.

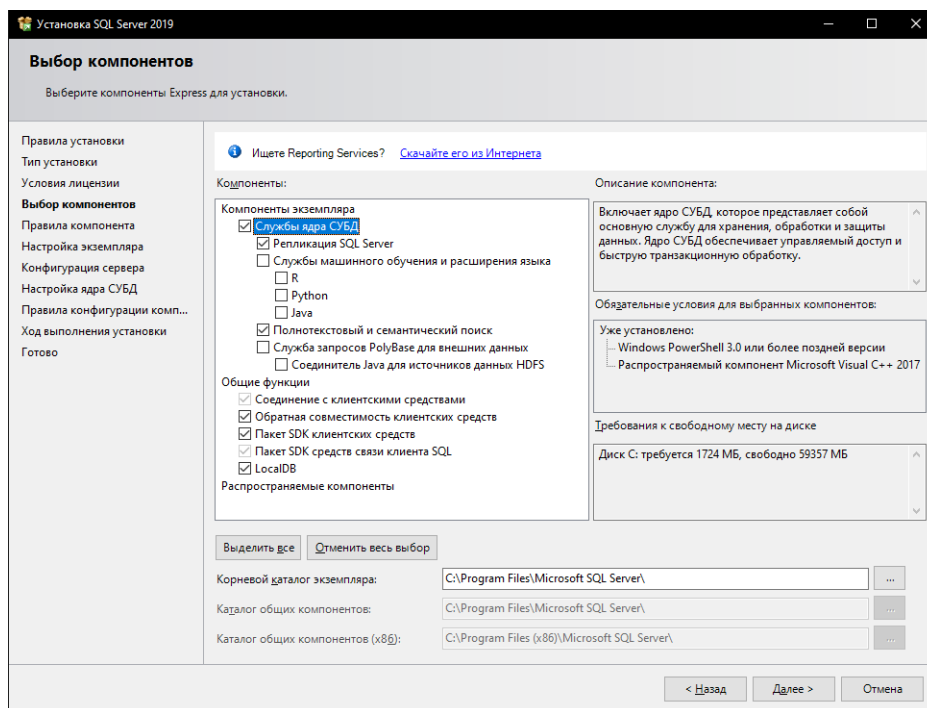


Рисунок 66 - Выбор компонентов

7. Следовать рекомендациям до пункта «Настройка экземпляра», на этом этапе необходимо придумать имя экземпляра (Рис. 67);

☐ Экземпляр по умолчанию
☒ Именованный экземпляр:

Идентификатор экземпляра:

Каталог SQL Server: C:\Program Files\Microsoft SQL Server\MSSQL15.SQL_MDA

Рисунок 67 - Настройка экземпляра

8. Следовать рекомендациям до пункта «Настройка ядра СУБД», на этом этапе необходимо выбрать смешанный режим унификации и установить пароль (Рис. 68). В результате на ваш компьютер установится сервер SQL (Ядро СУБД);

Укажите режим проверки подлинности, администраторов, каталоги данных, TempDB, максимальную степень параллелизма, ограничения памяти и параметры FILESTREAM для ядра СУБД.

Режим аутентификации

☐ Режим аутентификации Windows
☒ Смешанный режим (аутентификация SQL Server и Windows)

Укажите пароль для учетной записи системного администратора SQL Server (sa).

Введите пароль:

Подтвердите пароль:

Назначить администраторов SQL Server

DIMAM.Dima (Dima)	Администраторы SQL Server имеют неограниченный доступ к ядру СУБД.

Рисунок 68 - Настройка ядра СУБД

9. После успешной установки необходимо запустить из «\Каталог поставщика» файл SSMS-Setup-ENU.exe (Рис. 69);

RELEASE 19.0.2

Microsoft SQL Server Management Studio with Azure Data Studio

Welcome. Click "Install" to begin.

Location:

By clicking the "Install" button, I acknowledge that I accept the [Privacy Statement](#) and the License Terms for [SQL Server Management Studio](#) and [Azure Data Studio](#)

SQL Server Management Studio transmits information about your installation experience, as well as other usage and performance data, to Microsoft to help improve the product. To learn more about data processing and privacy controls, and to turn off the collection of this information after installation, see the [documentation](#)

Рисунок 69 - Мастер установки SSMS

10. Нажать «Install» и следовать рекомендациям установщика;

11. После успешной установки SSMS необходимо настроить доступ к экземпляру SQL Server. Для этого необходимо выполнить (Win+ R) `wf.msc` и перейти во вкладку «Правила для входящих подключений» (Рис. 70);

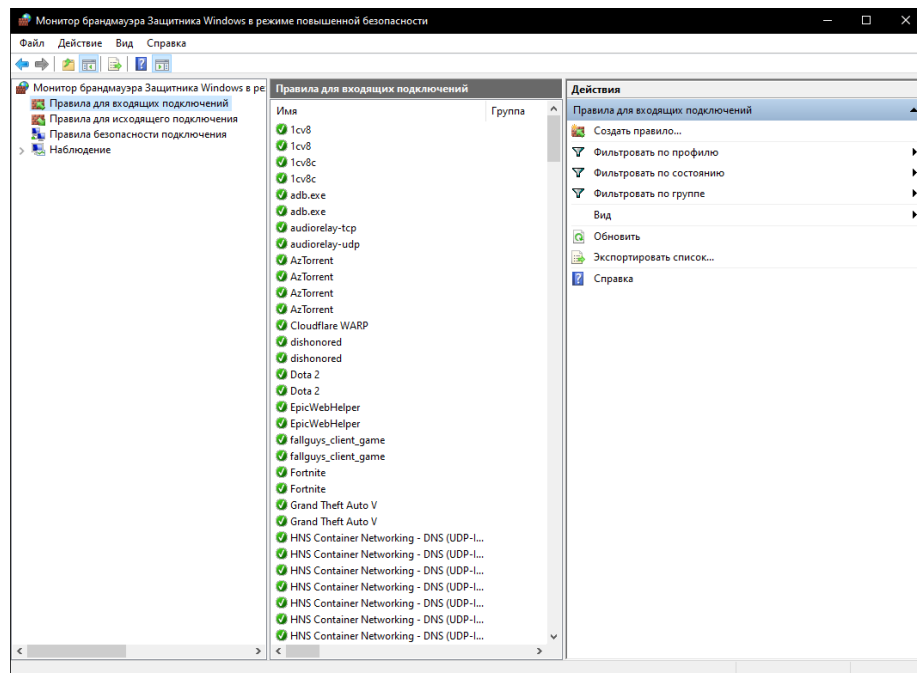


Рисунок 70 - Монитор Брандмауэра

12. Создать правило для порта TCP (Рис. 71);

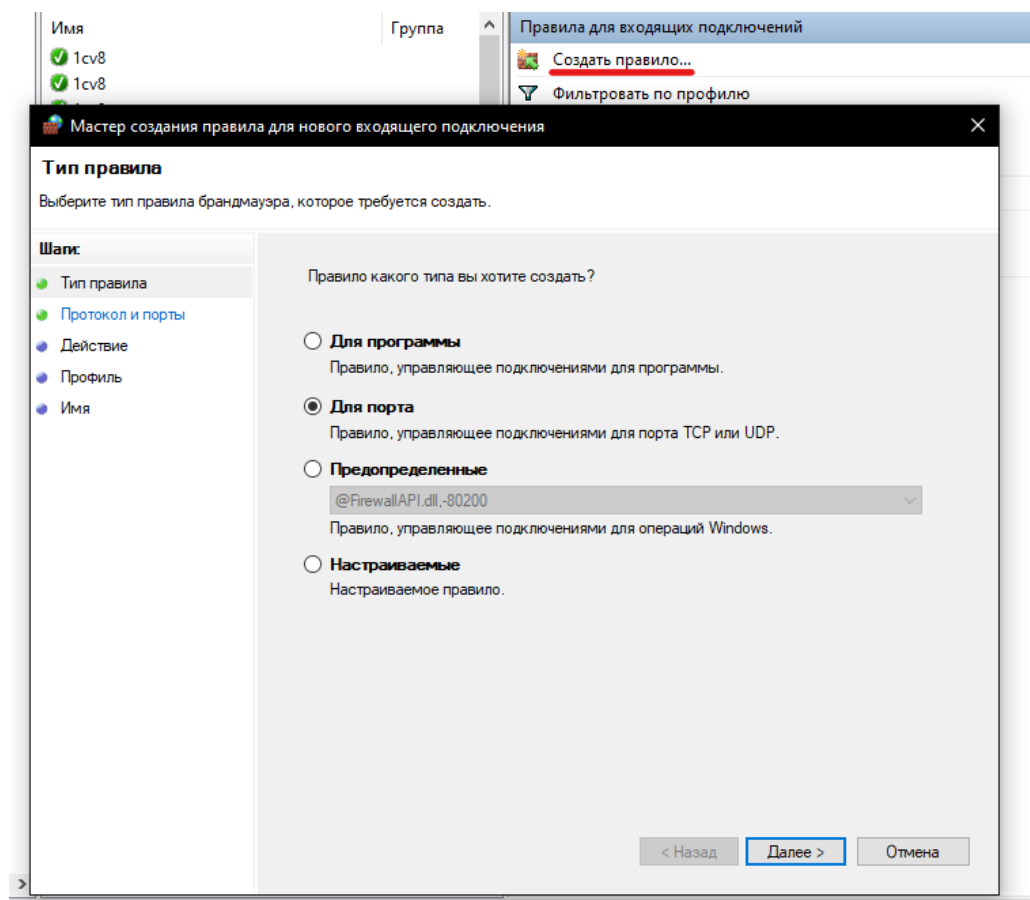


Рисунок 71 - Мастер создания правила

13. Правило должно применяться на определенный локальный порт «1433» (1433 – порт, используемый SQL Server Browser) (Рис. 72). Необходимо разрешить подключение и указать профили, к которым применяется это правило.

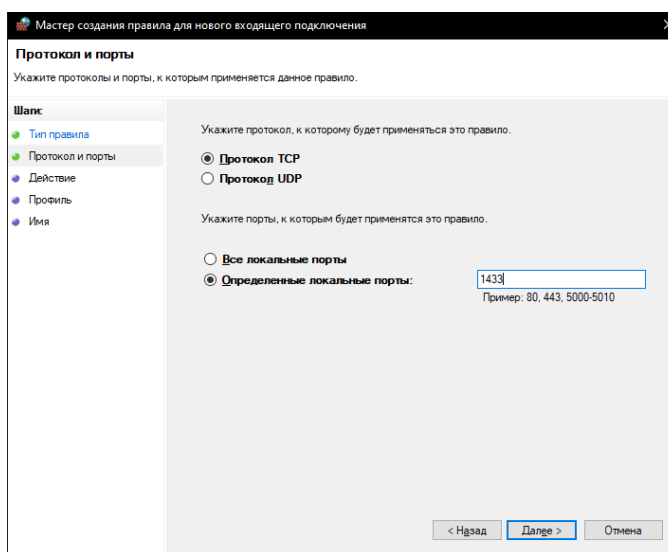


Рисунок 72 - Выбор протокола и локального порта

14. Указать имя данному правилу и нажать «Готово»;
15. После успешного создания правила, необходимо открыть программу SQL Server Configuration Manager (Это интерактивное приложение для управления всеми службами на основе SQL Server, сетевыми протоколами, портами средства прослушивания и создания псевдонимов серверов);
16. Открыть вкладку «Сетевая конфигурация SQL Server» и выбрать протоколы для необходимого ядра SQL Server (Рис. 73);

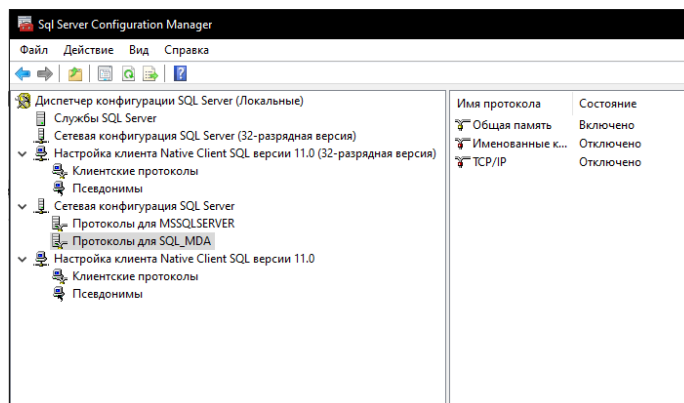


Рисунок 73 - Выбор протокола для необходимого ядра SQL Server

17. Включить протокол «Именованные каналы» (Рис. 74);

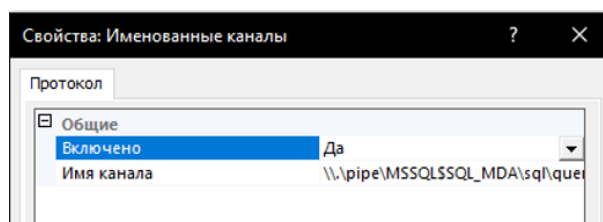


Рисунок 74 - Свойства протокола «Именованные каналы»

18. Включить протокол «TCP/IP» (Рис. 75), перейти в его свойства, во вкладку IP-адреса. Опуститься к пункту IPAll. и в TCP Порт записать 1433 (Рис. 76);

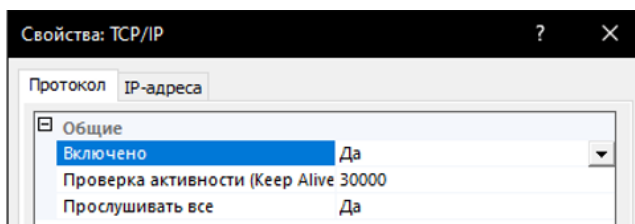


Рисунок 75 - Свойства протокола TCP/IP

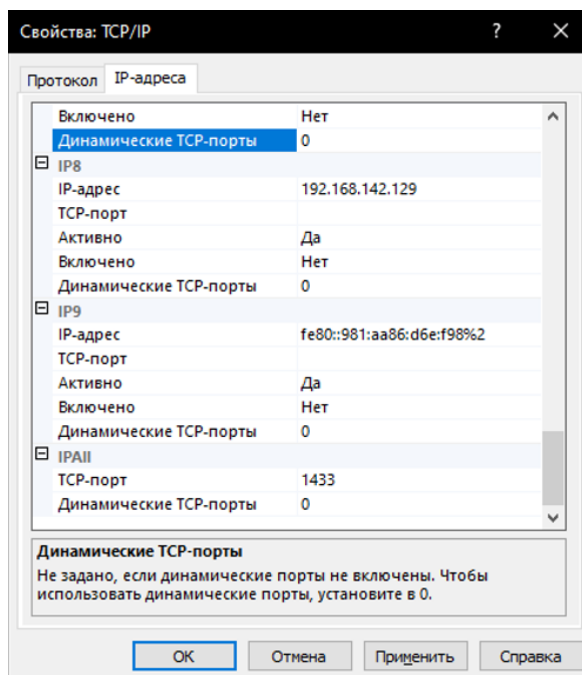


Рисунок 76 - IP-адреса протокола TCP/IP

19. Включить необходимый IP-адрес для подключения (Рис. 77) и перезапустить службу Ядра СУБД (Рис. 78);

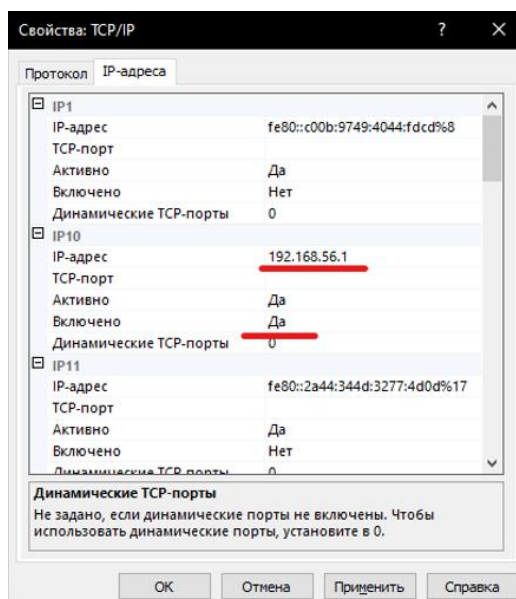


Рисунок 77 - IP-адрес для подключения

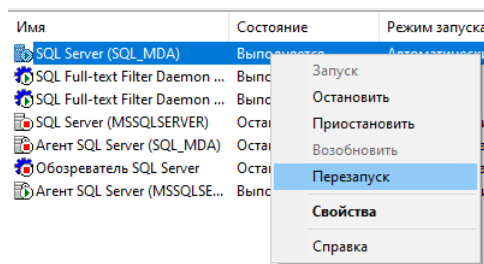


Рисунок 78 - Перезапуск службы

20. Запустить SSMS и подключиться к серверу (Рис. 79)

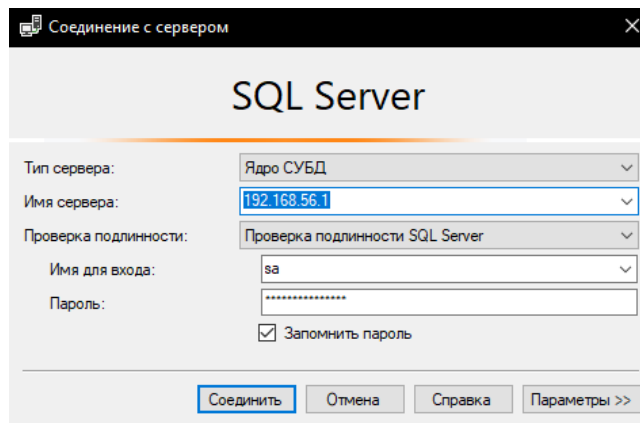


Рисунок 79 - Соединение с сервером

21. Создать пользователя для работы с базой данных (Рис. 80) Установить ему пароль и настроить права;

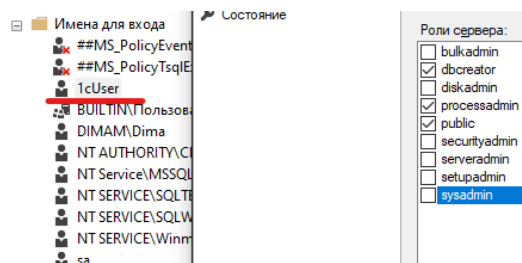


Рисунок 80 - Пользователь для работы с сервером

22. Создать пустую базу данных, указав имя и каталог хранения (Рис. 81).

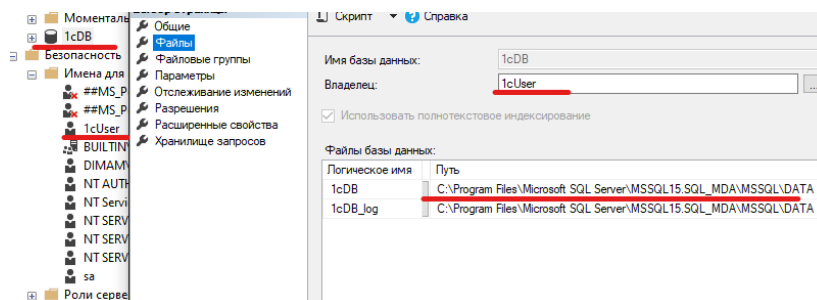


Рисунок 81 - Создание базы SQL

23. Запустить консоль управления сервером 1С и создать новую информационную базу указав ранее созданный сервер и базу SQL (Рис. 82, 83, 84);

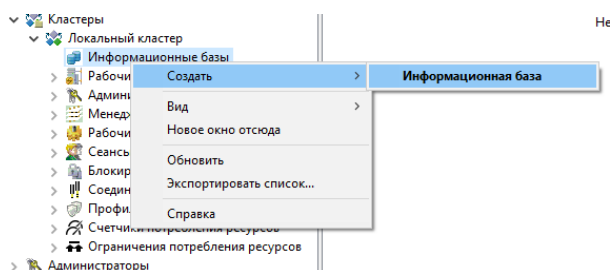


Рисунок 82 - Запуск консоли 1С

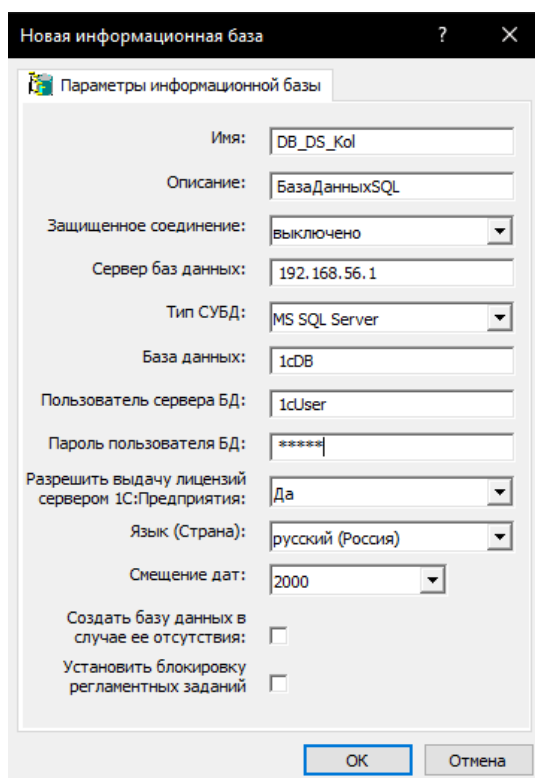


Рисунок 83 - Создание информационной базы

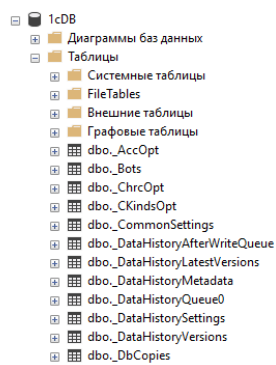


Рисунок 84 - Результат создания DB

24. Открыть 1С: Предприятие => Добавить уже существующую... Заполнить данные сервера и нажать «Далее», «Готово» (Рис. 85)

Добавление информационной базы/группы

Укажите наименование информационной базы:

SQL_УчетПосещаемости

Выберите тип расположения информационной базы:

☐ На данном компьютере или на компьютере в локальной сети

Каталог информационной базы:

C:\Users\Dima\Documents\InfoBase

☐ На веб-сервере

Адрес информационной базы:

https://example.com/infobase

[Дополнительно...](#)

☒ На сервере 1С:Предприятия

Кластер серверов:

DimaM

Имя информационной базы:

DB_DS_Kol

< Назад Далее > Отмена

Рисунок 85 - Подключение к серверу

25. Запустить Конфигуратор и загрузить выгрузку (Рис. 86).

О программе

1С®

1С:Предприятие 8.3 (8.3.19.1264)

Copyright © ООО "1С-Софт", 1996-2021. Все права защищены

Конфигурация:

Учет посещаемости

Использование конфигурации:

Наименование: SQL_УчетПосещаемости

Режим: Серверный

Сервер: DimaM

Имя базы: DB_DS_Kol

Приложение: Конфигуратор

Пользователь: Администратор

Локализация: Информационная база: русский (Россия), Сеанс: русский (Россия)

Лицензия: Текущая: Локальный HASP4 ORGL8 20, получило клиентское приложение Локальный HASP4 EN8SA 1 Информационная база...

[Лицензии используемых компонентов](#)

OK

Рисунок 86 - Ожидаемый результат

4.2 Запуск

Для запуска информационной системы 1С: Предприятие «Учет посещаемости» необходимо открыть 1С: Предприятие, выбрать информационную базу и нажать «1С: предприятие» (Рис. 87).

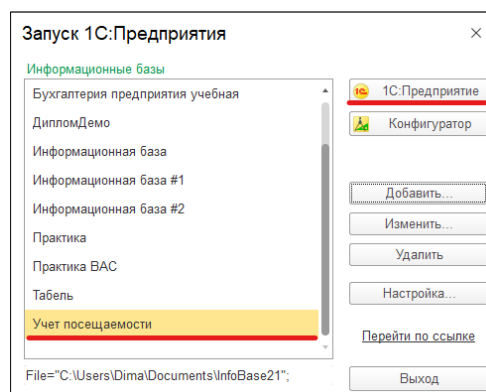


Рисунок 87 - Запуск информационной системы

При первом запуске приложения откроется форма «Общие сведения», в которой необходимо заполнить все данные (Рис. 88).

Рисунок 88 - Форма «Общие сведения»

4.3 Функции информационной системы

4.3.1 Работа со справочниками

Для заполнения, редактирования или удаления записей любого справочника необходимо перейти в раздел «Справочники» (Рис. 89) и выбрать нужный справочник.

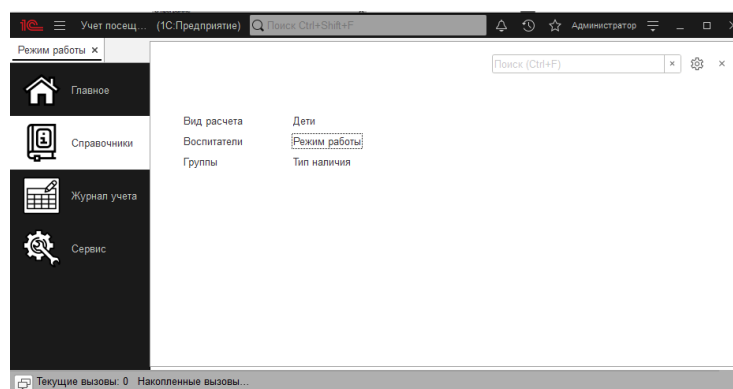


Рисунок 89 - Раздел "Справочники"

Откроется форма списка справочника (Рис. 90). На данной форме по кнопке «Еще» можно получить весь список доступных команд (Рис. 91). Каждая команда имеет название той функции, которую реализует.

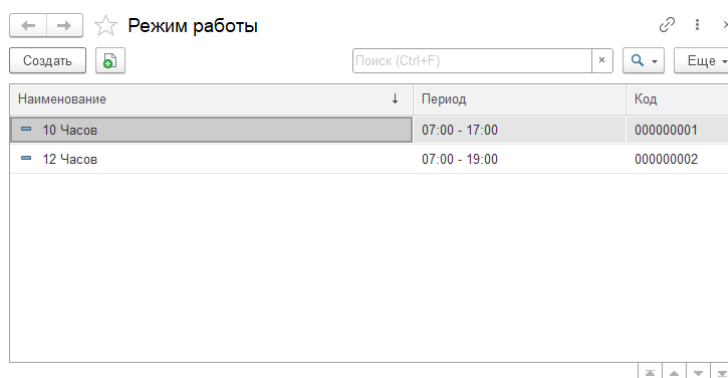


Рисунок 90 - Форма списка справочника

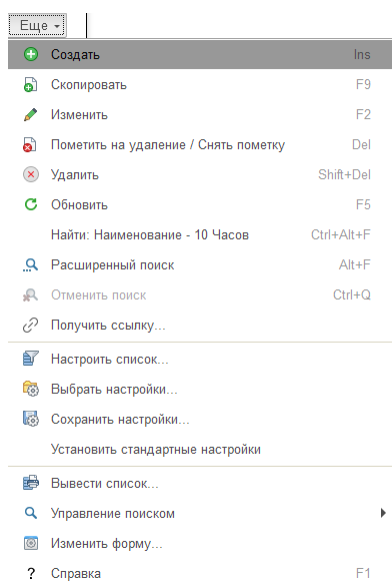


Рисунок 91 - Доступный список команд

Для добавления или изменения записи необходимо перейти на форму элемента справочника (Рис. 92). Для создания элемента необходимо нажать кнопку «Создать», для редактирования «Изменить» (или двойной клик по элементу в списке), для удаления «Удалить».

Рисунок 92 - Форму элемента справочника

По кнопке «Записать» данные сохраняются.

4.3.2 Регистрация и учет детей

Для регистрации детей в системе необходимо перейти в справочник «Дети» и нажать кнопку «Создать». После чего откроется форма элемента справочника (Рис. 93).

Рисунок 93 - Форма элемента справочника "Дети"

Пунктирной красной линией выделены поля обязательные для заполнения.

Поле ФИО имеет ряд ограничений:

- 1) ввод только кириллицы;
- 2) только 3 слова в строке, разбитой пробелами. Пример: «Иванов Иван Иванович».

В случае неправильного заполнения будет выведено сообщение «Что-то не так с ФИО».

Поля «Дата зачисления» и «Дата отчисления» определяют временной период, в течение которого ребенок будет зарегистрирован в системе.

Для того, чтобы зачислить ребенка в определенную группу, необходимо заполнить поля «Группа» и «Действует С». После того, как запись в справочнике будет создана, можно нажать кнопку «Обновить», в результате чего, в нижней таблице будет отображена история зачислений и переводов.

На форме списка детей есть поле отбора (Рис. 94), которое позволяет отобразить актуальные списки детей на введенную дату.

ФИО	Группа	Номер
Белов Александр Иванович	Зайки	X000
Григорьева Ирина Анатольевна	Зайки	X000
Иванова Марина Андреевна	Зайки	X000
Калинина Любовь Васильевна	Осминожки	X000

Рисунок 94 - Поле отбора

Справочник «Группы» позволяет просматривать актуальные списки (Рис. 95) на текущий день.

Зачисление	Ребенок
23.11.2022	Смирнов Виктор Владимирович
01.09.2022	Белов Александр Иванович
01.09.2022	Тихонов Андрей Геннадьевич
08.04.2023	Романова Анастасия Петровна
12.04.2023	Маркова Анна Сергеевна
10.04.2023	Григорьева Ирина Анатольевна
10.04.2023	Иванова Марина Андреевна

Рисунок 95 - Справочник "Группы"

4.3.3 Работа с табелем

Для создания табеля необходимо перейти в раздел «Журнал учета» (Рис. 96) и нажать «Табель учета посещаемости».

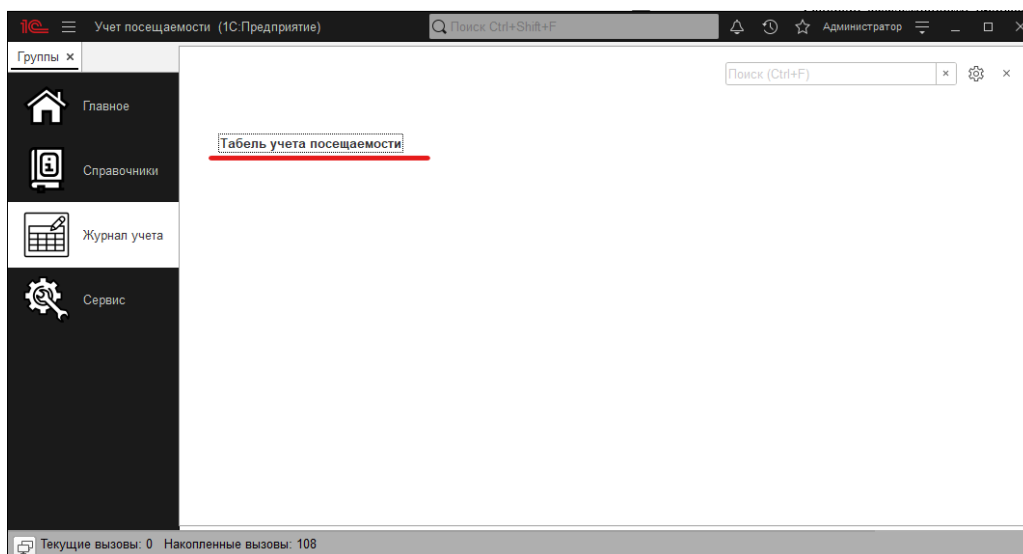


Рисунок 96 - Раздел «Журнал учета»

В форме списка документа нажать кнопку «Создать». Откроется форма документа табеля (Рис. 97).

Табель учета посещаемости (создание)

Провести и закрыть Записать Провести Печать

Номер: Дата: 15.04.2023 0:00:00 Группа: Вид расчета:

Период: Апрель 2023 г. Всего дней: 30 Рабочих: 20

Заполнить Настроить выходные

Ребенок	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Причины непосещения

НУ - Неявка по уважительной причине
 НЯ - Неявка без уважительной причины
 НБ - Неявка по болезни
 В - Выходной
 X - Отчислен

Рисунок 97 - Форма документа "Табель учета посещаемости"

В верхней части формы находится поле период, которое позволяет выбрать месяц и год табеля (Рис. 98).

Изменение периода

Год: 2023 Месяц: Апрель

Вы

- Январь
- Февраль
- Март
- Апрель
- Май
- Июнь
- Июль
- Август
- Сентябрь
- Октябрь
- Ноябрь
- Декабрь

Рисунок 98 - Выбор периода табеля

Настроить выходные дни месяца можно при помощи кнопки «Настроить выходные». После нажатия откроется форма (Рис. 99), в которой при помощи календаря можно изменить статус дня.

Изменение выходных

2023		Апрель			
Пн	Вт	Ср	Чт	Пт	Сб
					1
					2
3	4	5	6	7	8
10	11	12	13	14	15
17	18	19	20	21	22
24	25	26	27	28	29

Выходной
Рабочий

Сохранить

Рисунок 99 - Форма изменения выходных

В центре формы расположен табель посещаемости. Для его заполнения необходимо нажать кнопку «Заполнить» (Рис. 100), при этом поля группа и период не должны быть пустыми.

Номер: Дата: 15.04.2023 0:00:00 Группа: Осминокки Вид расчета:

Период: Апрель 2023 г. Всего дней: 30 Рабочих: 20

Ребенок	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Причины непосещения
Никитина Нина	В	В						В	В						В	В						В	В						В	В	
Шестаков Даниил	В	В						В	В						В	В						В	В						В	В	
Лебедева Ксения	В	В						В	В						В	В						В	В						В	В	
Федотова Алина	В	В	х	х	х	х	х	В	В						В	В						В	В						В	В	
Капинина Любовь	В	В	х	х	х	х	х	В	В						В	В						В	В						В	В	

Рисунок 100 - Ожидаемый результат заполнения.

Для заполнения пропусков необходимо нажать на пересечение ребенка и дня пропуска. В выбранной ячейке откроется список выбора пропуска (Данные загружаются из справочника «Тип наличия») (Рис. 101). Ячейки «В» и «х» редактировать нельзя. При повторном нажатии кнопки «Заполнить» табель будет очищен для предотвращения ошибок.

В	-				В	В
В	НУ (Неявка по уважительной причине) НЯ (Неявка без уважительной причины) НБ (Неявка по болезни)					

Рисунок 101 - Выбор причины пропуска

В нижней части формы расположена область подсказки (Рис. 102). Она отображает данные из справочника «Тип наличия».

НУ - Неявка по уважительной причине
 НЯ - Неявка без уважительной причины
 НБ - Неявка по болезни
 В - Выходной
 х - Отчислен

Рисунок 102 - Область подсказки

Печать табеля производится по кнопке «Печать» (Рис. 103).

Утверждена приказом Минздрава России от 30.03.2015 N 52н

ТАБЕЛЬ
УЧЕТА ПОСЕЩАЕМОСТИ ДЕТЕЙ

за Апрель 2023 г.

Учреждение: Муниципальное дошкольное образовательное учреждение Новоржевский детский сад "Болотки" Форма по ОК/Д 0204808

Структурное подразделение: группа "Осминокки" (от 3 до 5 лет) Дата: 30.06.2021

Вид расчета: по ОКПО 24279350

Режим работы: по ОКПО

№ п/п	Фамилия, имя ребенка	Номер счета	Плата по ставке	Дни посещения																														Пропущено дней всего	Дни посещения, в том числе зачисляемые	Дни посещения, подлежащие оплате	Причины непосещения (основание)
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30				
1	Никитина Нина	X000008336	1 672	В	В									В	В							В	В					В	В	0	0	20					
2	Шестаков Даниил	X000008347	1 672	В	В									В	В							В	В					В	В	0	0	20					
3	Лебедева Ксения	X000009841	1 672	В	В									В	В	НУ						В	В					В	В	1	0	19					
4	Федотова Алина	X000008345	1 672	В	В	х	х	х	х	х	х	х	х	В	В	НБ						В	В					В	В	6	1	15					
5	Капинина Любовь	X000005961	1 672	В	В	х	х	х	х	х	х	х	х	В	В							В	В					В	В	5	0	15					
Всего присутствует детей				В	В	3	3	3	3	3	В	В	4	5	4	5	5	В	В	5	5	5	5	В	В	5	5	5	5	В	В	1	89				
Всего отсутствует детей				В	В	2	2	2	2	2	В	В	1	0	1	0	0	В	В	0	0	0	0	В	В	0	0	0	0	В	В	12					

Руководитель учреждения: Митосова Н.А. (подпись) Заведующий (подпись) Митосова Н.А. (расшифровка подписи) Ответственный исполнитель (подпись) Митосова Н.А. (расшифровка подписи) Печенова В.В. (расшифровка подписи)
 Воспитатель: Печенова В.В. (подпись)

" 18 " Апрель 2023 г.

Рисунок 103 - Результат функции "Печать"

ЗАКЛЮЧЕНИЕ

Основой эффективности производственной практики является самостоятельная и индивидуальная работа студентов в производственных условиях.

Во время прохождения практики происходит закрепление и конкретизация результатов теоретического учебно-практического обучения, приобретение умения и навыков практической работы по присваиваемой квалификации и избранной специальности или профессии.

Совместно с руководителем практики был найден процесс, который нуждался в автоматизации. Разработанная информационная система полностью решила поставленную задачу и позволит в будущем совершать меньше ошибок и тратить меньше времени на заполнение и обработку табеля посещаемости детей.

Во время прохождения практики я выполнил следующее:

- своевременно приступил к прохождению практики;
- ответственно выполнял задания программы практики, индивидуальные поручения и указания руководителей практики, как со стороны учебного заведения, так и со стороны принимающей организации;
- в полном объеме и в установленные сроки выполнил программу практики и индивидуальное задание;
- в период практики знакомился с соответствующими учебно-методическими материалами;
- подбирал и обобщал материалы практики для подготовки отчета;
- соблюдал правила трудового и внутреннего распорядка в организации по месту прохождения практики.

В ходе прохождения практики в МДОУ Новодеревенский детский сад «Колокольчик» и подготовки отчета мною была рассмотрена краткая характеристика и организационная структура рассматриваемого объекта, была разработана конфигурация на платформе «1С: Предприятие», которая автоматизировала процесс заполнения и обработки табеля посещаемости детей.

В результате прохождения практики были решены поставленные цели и задачи, а также закреплены теоретические и практические знания, приобретены профессиональные умения и навыки в данной области.

СПИСОК ИСТОЧНИКОВ

1. Бартеньев О. В. 1С: Предприятие: программирование для всех; Диалог-МИФИ - , 2005. - 464 с.
2. Ивлев В. А., Попова Т. В. ABIS. Информационные системы на основе действий; 1С-Паблишинг - Москва, 2005. - 248 с.
3. Крылов Е. В., Острейковский В. А., Типикин Н. Г. Техника разработки программ. В 2 книгах. Книга 2. Технология, надежность и качество программного обеспечения; Высшая школа - Москва, 2008. - 472 с.
4. Кашаев С. 1С:Предприятие 8. Учимся программировать на примерах (+ CD-ROM); БХВ-Петербург - Москва, 2010. - 384 с.
5. Котляров В.П. Основы тестирования программного обеспечения: Учебное пособие / В.П.Котляров, Т.В.Коликова — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. — 285 с.
6. Гладкий А. А. 1С 8 с нуля. Комплексное руководство для начинающих / Гладкий А. А. - М.: Феникс, 2011. — 478 с.
7. Рыбалка, В. В. Hello, 1С. Пример быстрой разработки приложений на платформе 1С:Предприятие 8.3. Мастер-класс (+ CD-ROM) / В.В. Рыбалка. - М.: 1С-Паблишинг, 2012. - 233 с.
8. Информационная система 1С:ИТС: сайт. — 2018. — URL: <https://its.1c.ru/> (дата обращения: 07.04.2023). — Режим доступа: для зарегистрированных пользователей.