

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Южно-Уральский государственный университет»  
(Национальный исследовательский университет)  
Институт естественных и точных наук  
Кафедра «Математическое и компьютерное моделирование»

РАБОТА ПРОВЕРЕНА  
РЕЦЕНЗЕНТ,

\_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 2017 Г.

ДОПУСТИТЬ К ЗАЩИТЕ  
ЗАВЕДУЮЩИЙ КАФЕДРОЙ,  
Д.Ф.-М.Н., ДОЦЕНТ

\_\_\_\_\_ С.А.ЗАГРЕБИНА

« \_\_\_\_ » \_\_\_\_\_ 2017 Г.

Автоматизация сервисов учета клиентов танцевальной студии с помощью  
платформы «1С:Предприятие 8.3.»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ– 01.03.02.2017.072.10.000 ВКР

Нормоконтролер,  
Доцент каф. МиКМ, к.ф.-м.н.,  
\_\_\_\_\_ Т.А. Макаровских  
\_\_\_\_\_ 2017 г.

Руководитель проекта,  
к.ф.-м.н., доцент каф. МиКМ,  
\_\_\_\_\_ Т.А. Макаровских  
\_\_\_\_\_ 2017 г.

Автор работы  
Студент группы ЕТ-485  
\_\_\_\_\_ В.А. Маткин  
\_\_\_\_\_ 2017 г.

Дипломная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Южно-Уральский государственный университет»  
(Национальный исследовательский университет)  
Институт естественных и точных наук  
Кафедра «Математическое и компьютерное моделирование»  
Направление «Прикладная математика и информатика»

УТВЕРЖДАЮ  
Заведующий кафедрой МиКМ,  
\_\_\_\_\_ С.А. Загребина  
\_\_\_\_\_ 2017 г.

## ЗАДАНИЕ

на выпускную квалификационную работу студента

Маткина Виктора Алексеевича

Группа ЕТ-485

### 1 Тема работы

Автоматизация сервисов для клиентов танцевальной студии с помощью платформы «1С:Предприятие 8.3» утверждена приказом по университету от «\_\_» \_\_\_\_\_ 2017 г. № \_\_\_\_\_.

### 2 Срок сдачи студентом законченной работы

«\_\_» \_\_\_\_\_ 2017 г.

### 3 Исходные данные к работе

База данных о клиентах, сотрудниках и перечень услуг танцевальной студии «TROUBLE MAKERS»

4 Содержание пояснительной записки (перечень подлежащих разработке вопросов)

4.1 Анализ автоматизированных систем.

4.2 Анализ данных требуемых при разработке программного обеспечения.

4.3 Реализация программного комплекса.

4.4 Тестирование программного комплекса.

5 Перечень графического материала

5.1

5.1.1

5.1.2

5.1.3

5.1.4

5.1.5

5.1.6

5.1.7

5.1.8

5.1.9

5.1.10

5.1.11

5.1.12

5.1.13

6 Дата выдачи задания « \_\_\_ » \_\_\_\_\_ 2017 г.

Руководитель работы \_\_\_\_\_ /Т.А. Макаровских/

Задание принял к исполнению \_\_\_\_\_ /В.А. Маткин/

## КАЛЕНДАРНЫЙ ПЛАН

Наименование разделов выпускной квалификационной работы	Срок выполнения разделов работы	Отметка о выполнении руководителя
1. Сбор материала и литературы по теме дипломной работы	30.01.2017 – 17.02.2017	
2. Изучение теоретического материала	17.02.2017 – 04.03.2017	
3. Постановка задач	04.03.2017 – 16.03.2017	
4. Разработка программного обеспечения	16.03.2017 – 29.03.2017	
5. Тестирование программного обеспечения	29.03.2017 – 14.04.2017	
6. Проверка работы руководителем, исправление замечаний, получение отзыва	14.04.2017 – 13.05.2017	
7. Подготовка презентации и доклада	02.06.2017	
8. Нормоконтроль	10.06.2017	
9. Рецензирование, представление зав. кафедрой	17.07.2017	

Руководитель работы \_\_\_\_\_ /Т.А. Макаровских/

Задание принял к исполнению \_\_\_\_\_ /В.А. Маткин/

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Южно-Уральский государственный университет  
(Национальный исследовательский университет)»  
Институт естественных и точных наук  
Кафедра «Математическое и компьютерное моделирование»

## АННОТАЦИЯ

Маткин В.А. Автоматизация сервисов для клиентов танцевальной студии / В.А.Маткин. – Челябинск: ЮУрГУ, Институт естественных и точных наук, 2017. – 41 с., 8 ил., 6 табл., библиогр. список – 9 названий

В дипломной работе была разработана система автоматизации. Были рассмотрены теоретические основы разработки автоматизированных систем. Был проведен анализ данных для разработки программного обеспечения. Была построена конфигурация на платформе «1С:Предприятие 8.3». Произведено тестирование и оценка эффективности разработанной программы.

<b>ОГЛАВЛЕНИЕ</b>	
<b>ВВЕДЕНИЕ</b> .....	9
<b>1 Анализ предметной области «автоматизированные системы для школ танцев»</b> .....	12
<b>1.1 Структура управления танцевальной студии «TROUBLE MAKERS»</b>	12
<b>1.2 Особенности автоматизации работы школы танцев</b> .....	13
<b>1.3 Анализ автоматизированных систем на рынке</b> .....	15
<b>Выводы по главе один</b> .....	17
<b>2 Проектирование конфигурации «1С:Школа танцев»</b> .....	19
<b>2.1 Обоснование выбора среды для реализации программного обеспечения</b> .....	19
<b>2.2 Сущности модели базы данных конфигурации</b> .....	20
<b>2.3 справочники</b> .....	22
<b>2.4 Документы</b> .....	25
<b>2.5 Регистрация информации</b> .....	31
<b>2.5 Интерфейс</b> .....	36
<b>Выводы по главе два</b> .....	38
<b>3 Возможности конфигурации «1С:Школа танцев»</b> .....	40
<b>3.1 Подсистема «клиенты и сотрудники»</b> .....	40
<b>3.2 Подсистема «Финансы»</b> .....	43
<b>3.3 Подсистема «Графики»</b> .....	44
<b>3.4 Рабочий стол</b> .....	45
<b>Выводы по главе три</b> .....	46
<b>Заключение</b> .....	47
<b>БИБЛИОГРАФИЧЕСКИЙ СПИСОК</b> .....	48
<b>ПРИЛОЖЕНИЕ А</b> .....	50
<b>ТЕХНИЧЕСКОЕ ЗАДАНИЕ</b> .....	50
Министерство образования и науки Российской Федерации.....	50
Федеральное государственное автономное образовательное учреждение высшего образования.....	50
«Южно-Уральский государственный университет (национальный исследовательский университет)».....	50
Институт естественных и точных наук.....	50
Кафедра «Математическое и компьютерное моделирование» .....	50

1	ВВЕДЕНИЕ.....	51
1.1	Наименование программного изделия.....	51
2	Назначение и цели создания (развития) системы .....	51
2.1	Назначение системы.....	51
2.2	Цели создания системы.....	51
3	Характеристика объектов автоматизации .....	51
4	Требования к системе.....	52
4.1	Требования к системе в целом .....	52
4.2	Требования к функциям и задачам, выполняемым системой .....	52
4.3	Требования к видам обеспечения .....	52
4.4	Требования к квалификации.....	53
4.5	Требования к защите информации от несанкционированного доступа .....	53
4.6	Требования к надежности .....	54
4.6.1	Требования к надежному функционированию .....	54
4.6.2	Контроль входной и выходной информации .....	55
4.6.3	Время восстановления после отказа .....	55
4.7	Требования к составу и параметрам технических средств .....	55
4.8	Требования к стандартизации.....	55
5	Состав и содержание работ по созданию системы .....	57
	<b>ПРИЛОЖЕНИЕ Б</b> .....	59
	<b>МОДУЛИ ОБЪЕКТОВ</b> .....	59
	Министерство образования и науки Российской Федерации.....	59
	Федеральное государственное автономное образовательное учреждение высшего образования.....	59
	«Южно-Уральский государственный университет (национальный исследовательский университет)».....	59
	Институт естественных и точных наук.....	59
	Кафедра «Математическое и компьютерное моделирование» .....	59
	<b>Б1. Модуль объекта документа «Установка цен» .....</b>	<b>60</b>
	<b>Б2. Модуль объекта документа «Установка цен» .....</b>	<b>60</b>
	<b>Б3. Модуль объекта документа «Возврат Долга Клиентом» .....</b>	<b>61</b>
	<b>Б3. Модуль объекта документа «Уведомление» .....</b>	<b>61</b>
	<b>Б4. Модуль объекта документа «Предварительная Запись» .....</b>	<b>61</b>



<b>ПРИЛОЖЕНИЕ В</b> .....	64
<b>МОДУЛИ ФОРМ</b> .....	64
Министерство образования и науки Российской Федерации.....	64
Федеральное государственное автономное образовательное учреждение высшего образования.....	64
«Южно-Уральский государственный университет (национальный исследовательский университет)».....	64
Институт естественных и точных наук.....	64
Кафедра «Математическое и компьютерное моделирование» .....	64
<b>В1. Модуль формы элемента «Клиенты»</b> .....	65
<b>В2. Модуль формы списка «УслугиРаботы»</b> .....	65
<b>В3. Модуль формы выбора «УслугиРаботы»</b> .....	65
<b>В4. Модуль формы документа «ПриемНаряд»</b> .....	66
<b>В5. Модуль формы отчета «ДневнойЛистокУчета»</b> .....	71
<b>В6. Модуль формы отчета «ДниРождения»</b> .....	74
<b>В7. Модуль формы обработки «ПриемКлиентов»</b> .....	78
<b>В8 Модуль формы обработки «Уведомления»</b> .....	88
<b>В10. Модуль формы обработки «ЖурналЗаписиНаПрием»</b> .....	90

## ВВЕДЕНИЕ

Рассмотрим создание автоматизированной системы учета клиентов для школы танцев. Целевой аудиторией таких организаций являются люди разных возрастов, поэтому необходима четкая структура автоматизации подобных заведений, по возможности, исходя из занятости, иметь ряд ограничений, которые будут учитываться в системе автоматизации. Рост подобных школ с каждым годом растет, растет и конкуренция между ними. Ни одна студия танца в настоящее время не может обойтись без информационной поддержки, осуществляемой посредством соответствующих программных продуктов, чтобы быть конкурентоспособным в сфере данных услуг. Чаще всего для этого используются базы данных.

Легкий и быстрый доступ к нужной информации является одним из главных критериев эффективного управления. Важной категорией являются системы обработки информации, от которых во многом зависит эффективность работы предприятия или учреждения. Такая система должна:

- обеспечивать получение общих и/или детализированных отчетов по итогам работы;
- позволять легко определять тенденции изменения важнейших показателей;
- обеспечивать получение информации, критической по времени, без существенных задержек;
- выполнять точный и полный анализ данных.

Так как работы по учету ведутся вручную, создание автоматизированной системы будет **актуально**. Упрощение работы с клиентами позволит при минимальных затратах усилий получить наилучший результат, минимизировать ошибки, связанные с человеческим фактором и расширить потенциал компании, а также позволит конкурировать с более крупными организациями в этой сфере.

**Целью работы** является создание автоматизированной системы для студии танцев, чтобы вести учет работы школы танцев, просматривать перечень клиентов, тренеров, услуг предоставляемых студией. Также вести учет посещаемости клиентов, упростить внесение изменений, добавление записей, свести к минимуму трудности исправления допущенных ошибок, упростить работу персонала.

Исходя из цели работы были поставлены следующие **задачи**:

- формирование базы данных для сбора и хранения информации о тренерах, клиентах и расписания занятий школы танцев;
- разработка ПО для ведения и анализа учета посещаемости клиентов и работы персонала;

**Объект исследования** – танцевальная студия.

**Предмет исследования** – автоматизация деятельности танцевальной студии «TROUBLE MAKERS»

**Информационной** базой дипломной работы послужили данные о клиентах, сотрудниках и месте размещения танцевальной студии «TROUBLE MAKERS».

В качестве источника информации о работе с платформой «1С: Предприятие 8.3» послужили работы Макаровских Т.А., Радченко М.Г., Хрустальной Е.Ю, Диго С.М., был тщательно изучен видеокурс Камкова А.Г.

Работа состоит из введения, трех глав, заключения, библиографического списка и приложений.

В первой главе содержится анализ автоматизированных систем на рынке, обзор исследуемой студии, ставятся задачи необходимые при создании программного обеспечения.

Во второй главе описывается анализ данных, требуемых для дальнейшей разработки и внедрения этих данных в систему.

В третьей главе представлены возможности разработанного программного обеспечения и внедрение автоматизированной системы на примере студии танцев «TROUBLE MAKERS».

# 1 Анализ предметной области «автоматизированные системы для школ танцев»

## 1.1 Структура управления танцевальной студии «TROUBLE MAKERS»

Школа танцев «TROUBLE MAKERS» – это школа, предоставляющая услуги по изучению современных стилей танца.

Данная школа танцев насчитывает порядка трехсот постоянных клиентов, а штат сотрудников растет с каждым годом. Структура управления студии танцев «TROUBLE MAKERS» представлена на рисунке 1.



Рисунок 1 – Структура управления школы танцев «TROUBLE MAKERS»

### 1.1.1 обязанности директора

Центральное место в структуре школы танцев занимает директор школы, который ведет деятельность по двум основным направлениям: внешняя деятельность и внутренняя.

Внешняя деятельность включает:

- основное направление развития школы танцев;
- введение дополнительных услуг;
- поиск сотрудников и проведение собеседований;
- определение маркетинговой стратегии;
- поиск возможностей развития школы, открытие дополнительных филиалов;

Внутренняя деятельность состоит из:

- составления свода правил функционирования школы;
- контроль качества работы;
- распределение бюджета;
- выдача заработной платы сотрудникам;

### **1.1.2 Обязанности администратора**

В обязанности администратора входит:

- предварительная запись клиентов на занятия;
- ведение клиентской базы;
- продажа абонементов и принятие оплаты;
- учет посещаемости занятий;
- принятие звонков, консультация;
- поиск новых клиентов;

### **1.1.3 Обязанности тренера-преподавателя**

В обязанности преподавателя входит:

- ознакомление с техникой безопасности учеников на каждом занятии;
- поддержание порядка в зале;
- повышение квалификации;

### **1.1.4 Обязанности обслуживающего персонала**

Обслуживающий персонал обязан поддерживать порядок рабочего места тренеров, администратора и директора.

## **1.2 Особенности автоматизации работы школы танцев**

По мере развития предприятия его сотрудникам все сложнее видеть глобальную картину происходящего в студии, вести контроль клиентов, статистику, наблюдать за современными тенденциями в отрасли, оперативно принимать решения и совершать сделки, прогнозировать спрос и планировать предложения. Комплексная автоматизация процессов работы школы танцев поможет справиться с этими задачами и обеспечить постоянное развитие.

Задачи автоматизации того или иного заведения сферы услуг определяются его бизнес-процессами. В случае автоматизации школы танцев имеется несколько особенностей, отличающих их даже от других спортивных комплексов.

Прежде всего, это необходимость индивидуального подхода к клиенту. Если, к примеру, на горнолыжной трассе система может просто регулировать проход к подъёмнику, не требуя персонализации, то в школу танцев клиент идет за определенной процедурой, и даже к определенному тренеру. Посетитель школы танцев оплачивает услуги и рассчитывает получить за эти деньги индивидуальное обслуживание, рассчитанное именно на его потребности. Ему важно время оказания услуги, место, специалист, оказывающий услугу. Во-вторых, здесь важна возможность забронировать услугу. В третьих, школы танцев постоянно расширяют спектр предлагаемых услуг: групповые танцевальные занятия, различные направления, персональные тренировки и прочее. Это требует от программы автоматизации большой гибкости и возможности настройки в процессе эксплуатации.

Возможности автоматизации школы танцев включают в себя:

- учет клиентов;
- учет посещений;
- учет времени посещения занятий;
- возможность записи по тренерам;
- учет оплаты услуг, организация различных форм оплаты, учет бесплатных услуг
- формирование и печать оперативных отчетов, автоматизация финансового анализа деятельности клуба.

Система автоматизации – комплексное решение, обладающее несколькими интересными, именно для школы танцев, качествами. Прежде всего, это многофункциональность.

Каждый клиент, воспользовавшийся услугами школы танцев, учитывается системой с сохранением истории всех услуг в системе отчетов. На основании этих данных администратор школы танцев анализируют статистику посещаемости студии, динамику востребованности тех или иных услуг.

### 1.3 Анализ автоматизированных систем на рынке

На современном рынке имеется множество программных средств для автоматизации работы школы танцев. Так как каждая школа танцев имеет ряд особенностей, в зависимости от структуры организации, есть возможность выбора и настройки автоматизированной системы, удовлетворяющей потребностям студии, независимо, относится школа к малому бизнесу либо это крупное предприятие.

Рассмотрим функционал некоторых программ автоматизации школ танцев и проведем анализ. Особенности исследуемого программного обеспечения представлены в таблице 1.

Таблица 1 – особенности систем автоматизации

ПО особенности	«ЛистОк»	«Dance Studio»	«Хеликс»	«Альфа CRM»	«holihop»
Быстрый запуск	✓	✗	✓	✗	✓
Разграничение прав доступа персонала	✗	✓	✓	✗	✓
СМС и Email рассылка	✓	✓	✓	✓	✓
Учет посещений	✓	✓	✓	✓	✓



Продолжение таблицы 1

Автоматический расчёт зарплаты	✓	✓	✓	✗	✓
Мобильное приложение	✓	✓	✓	✗	✓
Возможность ведения учетов о продаже товаров	✗	✓	✓	✗	✓
Детальные отчеты	✓	✓	✓	✓	✓
Интеграция с сторонним сайтом	✓	✗	✗	✗	✗
Личный кабинет клиента	✓	✓	✓	✗	✓
Работа без интернета	✗	✓	✓	✓	✓
Импорт – экспорт данных exel	✓	✓	✓	✗	✓
Напоминание о днях рождения	✓	✓	✓	✗	✓
Печать документов и отчетов	✓	✓	✓	✗	✓
Интерфейс	✓	✗	✓	✗	✗

Создание информативных графиков	✓	✓	✓	✗	✗
Поддержка магнитных карт	✓	✓	✓	✗	✓
Цена и тарифы	5100-24990 рублей в месяц	От 7000 рублей в месяц	От 7000 до 100900 рублей	3000-5200 рублей в месяц	От 9100 рублей в месяц

Функциональность анализируемых систем однотипна, но каждая из программ имеет ряд достоинств и недостатков при использовании определенной организацией. Достоинством всех систем является то, что они качественно выполняют основные требования для ведения и анализа учета клиентов и персонала, имеется мобильное приложение, присутствует СМС-рассылка. Интерфейс вышеперечисленных программ требует доработки и простоты в использовании. Лидирующую строчку среди этих программ занимает «хеликс», имея максимальный спектр оказания услуг, но цена данного программного обеспечения превосходит остальные. Система автоматизации «Альфа-CRM» уступает в функционале остальным программам и рассчитана на малый объём данных.

#### **Выводы по главе один**

Таким образом, невозможно найти универсальное программное обеспечение для ведения учета танцевальной студии. Каждая школа танцев имеет свою специфику, методы ведения учета и самое главное – финансовые средства, которые могут не позволить приобрести дорогостоящий продукт. В силу этого, важным требованием к системам поддержки является возможность быстрой адаптации под задачи конкретной танцевальной студии. Такой гибкостью конфигурирования примечательна платформа 1С:Предприятие.

Ввиду этого, для реализации программного продукта в выпускной квалифицированной работе была выбрана среда «1С: Предприятие 8.3».

Таким образом, поставлены следующие задачи:

- 1) хранение в базе данных информации о клиентах, сотрудниках, различных видов абонементов и расписании;
- 2) разграничение прав доступа пользователей;
- 3) предоставление статистики по дням и периодам;
- 4) удаленный поиск и фильтрация данных;
- 5) Создание журнала записей
- 6) возможность предварительной записи клиентов на занятия;
- 7) учет посещаемости клиентов;

## **2 Проектирование конфигурации «1С:Школа танцев»**

### **2.1 Обоснование выбора среды для реализации программного обеспечения**

Конфигурация «1с:Школы танцев» разработана на платформе «1С:Предприятие 8.3» с использованием базовых и прикладных объектов предлагаемых данной платформой.

Система программ «1С: Предприятие 8» включает в себя платформу и прикладные решения, разработанные на ее основе, для автоматизации деятельности организаций и частных лиц. Сама платформа не является программным продуктом для использования конечными пользователями, которые обычно работают с одним из многих прикладных решений (конфигураций), разработанных на данной платформе. Такой подход позволяет автоматизировать различные виды деятельности, используя единую технологическую платформу.

Основу концепции системы «1С: Предприятие» составляет понятие конфигурация.

Конфигурацией в системе 1С: Предприятие называется совокупность взаимосвязанных составных частей:

- структуры учетных данных, форм их ввода, выбора, печати;
- состав механизмов учета итоговых данных и движений учетных данных;
- состав различных отчетов и обработок;
- набора пользовательских интерфейсов;
- набора ролей (прав доступа);
- набора общих процедур и функций (Глобальный модуль и общие модули), макетов табличных документов и др.;
- вспомогательных объектов: наборы пользовательских интерфейсов, наборы прав, а также различная вспомогательная информация (картинки, шаблоны, стили и т.д.).

## 2.2 Сущности модели базы данных конфигурации

Для отображения структуры будущих данных воспользуемся информационной моделью для отображения этих данных.

При создании объектных моделей данных используются такие понятия, как сущности, атрибуты и связи. Сущность – это отдельный элемент деятельности организации (сотрудник или клиент, место или вещь, понятие или событие), который должен быть представлен в базе данных. Атрибут – это свойство, которое описывает некоторый аспект объекта и значение которого следует зафиксировать, а связь является ассоциативным отношением между сущностями. Существует модель типа «сущность-связь», или ER-модель (Entity-Relationship model).

В настоящее время ER-модель стала одним из основных методов концептуального проектирования баз данных. Объектно-ориентированная модель расширяет определение сущности с целью включения в него не только атрибутов, которые описывают состояние объекта, но и действий, которые с ним связаны, т.е. его поведение. В таком случае говорят, что объект инкапсулирует состояние и поведение.

Основной концепцией ER-моделей является тип сущности, который представляет группу объектов реального мира, обладающих одинаковыми свойствами. Тип сущности характеризуется независимым существованием и может быть объектом с физическим (или реальным) существованием или объектом с концептуальным (или абстрактным) существованием. Каждая сущность может обладать любым количеством связей с другими сущностями модели.[13]

**Связь** (Relationship) – поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Связь – это ассоциация между сущностями, при которой каждый экземпляр одной сущности ассоциирован с произвольным (в том числе нулевым) количеством экземпляров другой сущности. Она показывает, как в предметной области информационные объекты взаимодействуют друг с другом. Между двумя

сущностями может быть определено несколько связей, каждая из которых обладает своей семантикой. Наличие множества связей и определяет сложность инфологических моделей. При построении ER-модели используются бинарные связи, т.е. связи между двумя сущностями. В этом случае можно выделить родительскую сущность (откуда выходит связь) и дочернюю сущность. Важной характеристикой является мощность связи (кардинальность или степень связи), которая показывает, сколько экземпляров дочерней сущности связано с одним экземпляром родительской сущности. Различают следующие типы связей:

- связь один к одному(1:1) – один объект А может принадлежать или же соответствовать один объект Б.
- связь один ко многим (1:M) – когда объекту А может принадлежать или же соответствовать несколько объектов Б, но объекту Б может соответствовать только один объект
- связь многие-ко-многим(M:M) – реализуется в том случае, когда нескольким объектам из таблицы А может соответствовать несколько объектов из таблицы Б, и в то же время нескольким объектам из таблицы Б соответствует несколько объектов из таблицы А.[13]

Для задач, поставленных в работе, задействованы следующие сущности:

- Клиенты (**код клиента**, № карты клиента, ФИО, дата рождения, скидка, адрес, телефон, пол физического лица, email, комментарий);
- Сотрудники (**код сотрудника**, ФИО, дата рождения, пол физического лица, адрес, телефон, email, комментарий);
- Организации (**код организации**, наименование, график работы, фактический адрес, юридический адрес, телефон, директор, главный бухгалтер);
- Услуги и работы (**код услуги**, наименование, цена);
- Скидки (**код скидки**, наименование, процент);
- Склады (**код склада**, наименование);

На рисунке 2 приведена схема сущностей и связей между ними, необходимые для обеспечения будущей конфигурации.

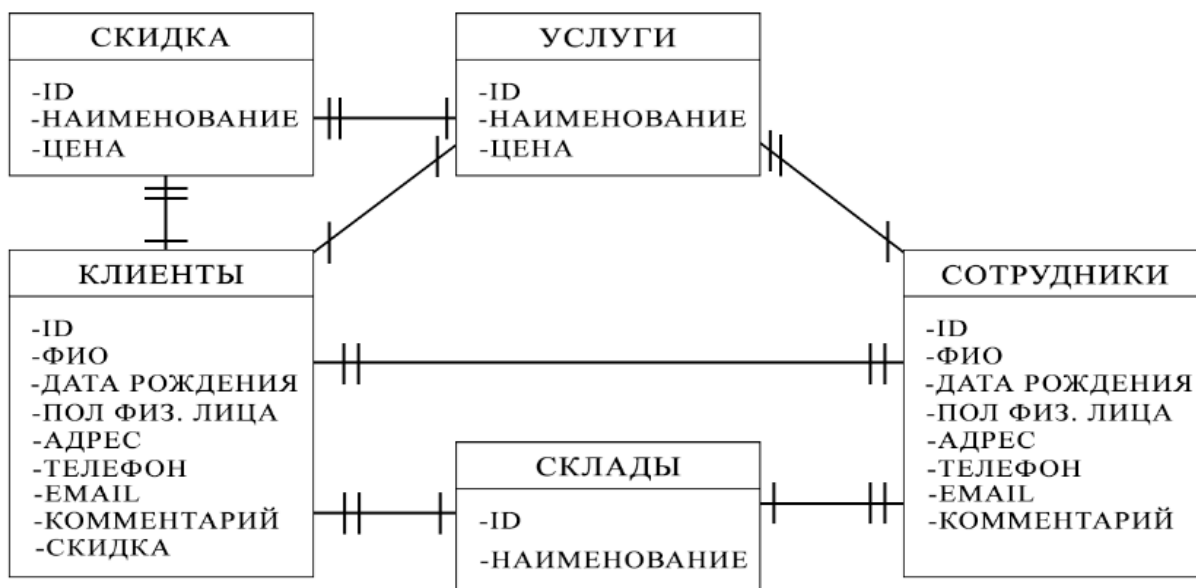


Рисунок 2 – ER диаграмма

### 2.3 справочники

В модели разработки «1С:Предприятие» используется подход, в котором все прикладное решение описывается метаданными в виде совокупности прикладных объектов, выбираемых из жестко определенного набора прототипов (классов). Можно было бы назвать создаваемые объекты бизнес-компонентами, а их прототипы шаблонами. Каждый такой прототип отражает определенные совокупности объектов или процессов предметной области, имеющих схожие поведенческие характеристики и сходную роль в общей картине решения. Одним из таких прототипов является «справочник», с его помощью удобно описать сущности схемы метаданных. [16]

При разработке конфигурации «1С: Школа танцев» были разработаны следующие справочники, которые соответствуют сущностям.

#### 2.3.1 Справочник «Клиенты»

Данный справочник хранит в себе личную информацию и данные о клиентах. Структура справочника приведена в таблице 2.

Таблица 2 – Структура справочника «Клиенты»

Реквизит	Синоним	Тип значения	Обязательное поле
Наименование	ФИО	Строка	ДА
Скидка	Скидка	СправочникСсылка. Скидки	Нет
Адрес	Адрес	Строка	Нет
Телефон	Телефон	Строка	Нет
Email	Email	СправочникСсылка.Залы	Нет
ДатаРождения	Дата рождения	Дата	Нет
Пол	Пол физ. лица	ПеречислениеСсылка.Пол	Да
Комментарий	Комментарий	Строка	Нет

В таблице 2 видно, что некоторые реквизиты имеет тип Справочник.Ссылка\* это значит, что данный тип является сложным и реализуется для тех объектов конфигурации, в которых хранятся данные объектного типа. Такими объектами являются: справочники, перечисления, документы и т.д. По сравнению с примитивными типами данных, сложные типы имеют больше функциональных возможностей прикладного решения. Именно благодаря им платформа позволяет связывать между собой разные объекты, хранить изображения в базе и другое. Их использование в других программных продуктах не представляется возможным в явном виде, поскольку они были созданы непосредственно для платформы 1:СПредприятие 8.X. [17]

### 2.3.2 Справочник «Сотрудники»

Данный справочник хранит в себе личную информацию и данные о клиентах. Структура справочника приведена в таблице 3.

Таблица 3 – Структура справочника «Сотрудники»

Реквизит	Синоним	Тип значения	Обязательное поле
Наименование	ФИО	Строка	ДА
Адрес	Адрес	Строка	Нет
Телефон	Телефон	Строка	Нет
Email	Email	СправочникСсылка.Залы	Нет
ДатаРождения	Дата рождения	Дата	Нет



Окончание таблицы 3			
Пол	Пол физ. лица	ПеречислениеСсылка.Пол	Да
Комментарий	Комментарий	Строка	Нет

### 2.3.3 Справочник «УслугиРаботы»

Данный справочник предназначен для хранения информации об услугах, предоставляемых школой танцев. (Табл.4)

Таблица 4 – Структура справочника «Сотрудники»

Реквизит	Синоним	Тип значения	Обязательное поле
Наименование	Наименование	Строка	ДА
Норма времени	Норма времени.мин.	Число	Да
Включать в прайс	Включать в прайс	Булево	Нет

### 2.3.3 Справочник «Скидки»

Справочник предназначен для учета скидки клиента на услугу (Табл. 5).

Таблица 5 – Структура справочника «Скидки»

Реквизит	Синоним	Тип значения	Обязательное поле
Наименование	Наименование	Строка	ДА
Процент	Процент, %	Число	Да

### 2.3.3 Справочник «Организации»

Данный справочник предназначен для хранения информации об организациях с которыми сотрудничает школа танцев. (Табл.6)

Таблица 6 – Структура справочника «Организации»

Реквизит	Синоним	Тип значения	Обязательное поле
Наименование	Наименование	Строка	ДА

ПолноеНаименован ие	Полное наименовани е	Строка	Да
Директор	Директор	СправочникСсылка.Сотрудн ики	Да
ГлавныйБухгалтер	Главный бухгалтер	СправочникСсылка.Сотрудн ики	Да
ЮридическийАдрес	Юридически й адрес	Строка	Да
ФактическийАдрес	Фактический адрес	Строка	Да
Телефон	Телефон	Строка	Нет
ГрафикРаботы	График работы	СправочникСсылка.Графики	Нет

### 2.3.3 Справочник «Склады»

С помощью справочника «Склады» осуществляется выбор зала для посещения клиентов (табл.7).

Таблица 6 – Структура справочника «Организации»

Реквизит	Синоним	Тип значения	Обязательное поле
Наименование	Наименование	Строка	ДА

### 2.4 Документы

Документы предназначены для хранения основной информации о всех событиях, происходящих на предприятии, и, разумеется, имеющих смысл с точки зрения экономики. При помощи документов отражаются и платежи с расчетного счета, и операции по кассе, и кадровые перемещения, и движения по складу, и прочие подобные события.

В процессе конфигурирования настраивается произвольное количество видов документов. Типичными примерами видов документов являются такие документы, как "Платежное поручение", "Счет", "Приходная накладная", "Расходная накладная", "Накладная на внутреннее перемещение",

"Приходный кассовый ордер" и так далее. Каждый вид документа предназначен для отражения своего типа событий. Это определяет его структуру и свойства, которые описываются в конфигурации.

Документы играют центральную роль для основных механизмов, реализуемых компонентами системы. В системе 1С:Предприятие документ является основной учетной единицей. Каждый документ содержит информацию о конкретной хозяйственной операции и характеризуется своим номером, датой и временем. Дата и время – наиболее важные характеристики документов, так как позволяют устанавливать строгую временную последовательность совершения операций. Все документы (вне зависимости от вида) образуют единую последовательность. Фактически, эта последовательность отражает последовательность событий – так, как они происходили реально. Внутри даты последовательность документов определяется их временем, при этом время документа является не столько средством отражения реального (астрономического) времени ввода документа, сколько средством, позволяющим четко упорядочить документы внутри одной даты.

Данные, вводимые в документ (реквизиты документа), обычно содержат информацию о произошедшем событии: например, в накладной – информацию о том, с какого склада, каких товаров и сколько отгружено; в приказе о приеме на работу – информацию о сотруднике, оклад, другие сведения. Кроме собственно записи, для документа весьма важным свойством является его проведение. При проведении документ может отразить зафиксированное им событие в механизмах, реализуемых компонентами. Например, если установлена компонента "Бухгалтерский учет", документ может записать бухгалтерскую операцию, отразив в виде проводок в бухгалтерском учете информацию, содержащуюся в документе.

С точки зрения программиста, у документа есть экранная форма – Диалог. Именно его видит пользователь на экране. Также у документа обычно есть одна или несколько бумажных форм – Таблиц. С помощью Таблиц

документ печатается на бумаге. Поведение документа определяется с помощью встроенного языка 1С. На языке записывается, что документ делает в системе, как он формирует бумажный бланк и как он себя ведет на экране.

Документ обычно имеет печатную форму. Печатная форма полностью формируется средствами встроенного языка 1С, что придает большую гибкость программе. Это одна из причин ошеломляющего успеха системы 1С:Предприятие. Печатная форма документа – это фактически отчет, формируемый обычно из формы открытого документа и содержащий сведения из этого документа. Чтобы научиться формировать печатные формы, вам нужно изучить главу Отчеты (объект "Таблица").

У документа может быть несколько шаблонов печатных форм. Средствами встроенного языка (метод ИсходнаяТаблица) задается тот шаблон печатной формы, который будет использоваться.

Шаблоны печатных форм (таблицы) могут быть общие. Это дает возможность использовать один шаблон для печати нескольких видов документов. Т.е. общая таблица является глобальной для конфигурации и может быть использована в любом модуле. Примером из типовой конфигурации может являться шаблон печатной формы "Отчет о движении документа", который используется разными модулями конфигурации.[19]

Разработанная конфигурация включает в себя документы, которые будут представлены далее.

#### **2.4.1 Документ «Прием»**

Данный документ служит для учета приема клиентов, пришедших на занятия. Структура документа приведена в таблице 7.

Таблица 7 – структура документа «прием»

<b>Реквизиты</b>			
<b>Реквизит</b>	<b>Синоним</b>	<b>Тип значения</b>	<b>Обязательное поле</b>

Скидка	Скидка	СправочникСсылка.ст_Скидки	Нет
ВремяНачала	Время начала	Дата, состав даты – время	Да
ВремяОкончания	Время окончания	Дата, состав даты – время	Да
флПрием	Прием	СправочникСсылка.Сотрудники	Да
Сотрудник	Сотрудник	СправочникСсылка.Сотрудники	Да
Клиент	Клиент	СправочникСсылка.Клиенты	Да
НомерКвитанцииОбОплате	Номер квитанции об оплате	Строка	Да
ОплаченнаяСумма	Оплаченная сумма	Число	Да
Сумма	Сумма	Число	Да
ВидыДенежногоРасчета	Виды денежного расчета	ПеречислениеСсылка.ВидыДенежногоРасчета	
Табличные части			
<b>Реквизит табличной части</b>	<b>Синоним</b>	<b>Тип значения</b>	<b>Обязательное поле</b>
УслугаРабота	Услуга (работа)	СправочникСсылка.УслугиРаботы	Да
Количество	Количество	Число	Да
Цена	Цена	Число	Да
ЦенаСоСкидкой	Цена со скидкой	Число	Нет
Сумма	Сумма	Число	Да

### 2.4.2 Документ «УстановкаЦен»

Документ позволяет установить цену для определенной услуги. Структура документа представлена в таблице 8.

Таблица 8 – структура документа «прием»

Табличные части			
Реквизит табличной части	Синоним	Тип значения	Обязательное поле
Услуга	Услуга	СправочникСсылка.УслугиРаботы	Да
Цена	Цена	Число	Да

### 2.4.3 Документ «ВозвратДолгаКлиентом»

Документ позволяет вести отчетность о задолженностях клиентов. Структура документа представлена в таблице 9.

Таблица 9 – структура документа «ВозвратДолгаКлиентом»

Реквизиты	Синоним	Тип значения	Обязательное поле
Клиент	Клиент	СправочникСсылка.Клиенты	Да
Сумма	Сумма	Число	Да

### 2.4.4 Документ «Уведомление»

Данный документ служит для оповещение клиентов об обязательном посещении мастер классов раз в полгода. Структура документа представлена в таблице 10.

Таблица 10 – структура документа «Уведомление»

<b>Реквизиты</b>			
<b>Реквизит</b>	<b>Синоним</b>	<b>Тип значения</b>	<b>Обязательное поле</b>
НеУведомлять	Не уведомлять	Булево	Нет
Текст	Текст	Строка	Да
<b>Табличные части</b>			
<b>Реквизит табличной части</b>	<b>Синоним</b>	<b>Тип значения</b>	<b>Обязательное поле</b>
Клиент	Клиент	СправочникСсылка.Клиента	Да

#### 2.4.5 Документ «Предварительная запись»

Документ позволяет вести учет клиентов по записи. Структура документа представлена в таблице 11.

Таблица 11 – структура документа «Предварительная запись»

<b>Реквизит</b>	<b>Синоним</b>	<b>Тип значения</b>	<b>Обязательное поле</b>
ТипЗаписи	Тип записи	ПеречислениеСсылка.ТипыПредварительнойЗаписи	Да
ВремяОкончания	Время окончания	Дата	Да
Клиент	Клиент	СправочникСсылка.Клиенты	Да
Телефон	Телефон	Строка	Да
Сотрудник	Сотрудник	СправочникСсылка.Сотрудники	Да
Зал	Зал	СправочникСсылка.Склады	Да
Неявка	Неявка	Булево	Нет
Первичная	Первичная	Булево	Нет
ВремяНачала	Время начала	дата	Да

Подтвержде ние	Подтвержд ение	Булево	Нет
Комментари й	Комментар ий	Строка	Нет

## 2.5 Регистрация информации

Существуют такие виды объектов, как регистры, их основное назначение – оптимизирование получения данных для дальнейших отчетов. Конфигуратор включает в себя четыре вида регистров:

- регистры накопления – показатели накопления;
- регистры сведений – показатели состояния;
- регистры бухгалтерии;
- регистры расчета.

Регистры служат для более быстрого считывания информации, их можно сравнить с каталогом библиотеки. То есть они не только хранят информацию, но систематизируют ее. В общем случае основное использование регистров в 1с можно изобразить следующей схемой: «Документ – регистр – отчет», но существуют исключения.[9]

В данной работе используются регистры сведений и регистры накопления.

Ресурс регистра сведений, в отличие от регистров другого вида, может иметь не только числовое значение, но и другой тип данных. Данный регистр имеет особое свойство, которое не используется в других видах регистров – периодичность, может быть независимым, в таком случае записи производятся непосредственно в регистр, минуя регистрирующий документ. Кроме того, данный вид регистра имеет автоматический контроль уникальности записей по периоду и измерениям.



Регистр накоплений предназначен для накопления числовых показателей и делится на два подвида – остатки и обороты. Данные регистра накопления хранятся в БД в виде двух таблиц – таблица движений и таблица итогов. Обращение напрямую возможно только к таблице движений.[10]

Далее представлены регистры, используемые при выполнении данной работы.

### 2.5.1 Регистр сведений «ПаспортныеДанныеФизЛиц»

Структура регистра представлена в таблице 12.

Таблица 12 – Структура регистра сведений  
«ПаспортныеДанныеФизЛиц»

<b>Измерения</b>	<b>Тип</b>
Владелец	СправочникСсылка.Сотрудники, СправочникСсылка.Клиенты
<b>Ресурсы</b>	<b>Тип</b>
ВидДокумента	СправочникСсылка.ДокументыУдостоверяющиеЛичность
Серия	Строка
Номер	Строка
КемВыдан	Строка
КодПодразделения	Строка

### 2.5.2 Регистр сведений «ГрафикиРаботыСотрудников»

Регистр хранит информацию о расписании рабочих дней сотрудников предприятия. Структура регистра «ГрафикиРаботыСотрудников» представлена в таблице 13.

Таблица 13 – Структура регистра сведений  
«ГрафикиРаботыСотрудников»

<b>Измерения</b>	<b>Тип</b>
Сотрудник	СправочникСсылка.Сотрудники
<b>Ресурсы</b>	<b>Тип</b>
График	СправочникСсылка.Графики

Окончание таблицы 13

### 2.5.3 Регистр сведений «ЦеныНаУслуги»

Хранит информацию о ценах на различные услуги, предоставляемые предприятием. Структура регистра «ЦеныНаУслуги» представлена в таблице 14.

Таблица 14 – Структура регистра сведений  
«ГрафикиРаботыСотрудников»

<b>Измерения</b>	<b>Тип</b>
Услуга	СправочникСсылка.УслугиРаботы
<b>Ресурсы</b>	<b>Тип</b>
Цена	Число

### 2.5.4 Регистр сведений «ИсторияПосещений»

Содержит информацию о посещениях занятий клиентами. Структура регистра «ИсторияПосещений» представлена в таблице 15.

Таблица 15 – Структура регистра сведений  
«ГрафикиРаботыСотрудников»

<b>Измерения</b>	<b>Тип</b>
Клиент	СправочникСсылка.Клиенты

### **2.5.5 Регистр сведений «УведомлениеОПолугодномПосещении МастерКласса»**

Хранит информацию о посещении мастер классов, проходящих раз в полгода. Структура регистра «Уведомление Ополугодном Посещении Мастер Класса» представлена в таблице 16.

Таблица 16 – Структура регистра сведений «УведомлениеОполугодномПосещенииМастерКласса»

<b>Измерения</b>	<b>Тип</b>
Клиент	СправочникСсылка.Клиенты
<b>Ресурсы</b>	<b>Тип</b>
Уведомлять	Булево

### **2.5.6 Регистр накопления «ДолгиКлиентов»**

Регистр хранит информацию о сумме задолженности пациента предприятию. Структура регистра «ДолгиПациентов» представлена в таблице 17.

Таблица 17 – Структура регистра сведений «ДолгиКлиентов»

<b>Измерения</b>	<b>Тип</b>
Клиент	СправочникСсылка.Клиенты

Ресурсы	Тип
Сумма	Число

### 2.5.7 Регистр сведений «ЗаписьНаПрием»

Данный регистр хранит в себе информацию о предварительной записи.

Структура оегистра «ЗаписьНаПрием» представлена в таблице 18.

Измерения			
Измерение	Синоним	Тип значения	Обязательное поле
Сотрудник	Сотрудник	СправочникСсылка.Сотрудники	Да
Зал	Зал	СправочникСсылка.Склады	Да
Ресурсы			
Ресурс	Синоним	Тип значения	Обязательное поле
ТипЗаписи	Тип записи	ПеречислениеСсылка.ТипыПредварительнойЗаписи	ДА
ВремяНачала	Время начала	Дата	Да
ВремяОкончания	Время окончания	Дата	Да
Клиент	Клиент	Справочник.Ссылка.Клиенты	Нет
Телефон	Телефон	Строка	Да
Подтверждение	Подтверждение	Булево	Нет
Неявка	Неявка	Булево	Нет
Первичная	Первичная	Булево	Нет
Реквизиты			

## 2.5 Интерфейс

**Подсистемы** – это общие объекты конфигурации. На их основе платформа формирует командный интерфейс прикладного решения и визуально разделяет всю функциональность программы на крупные и мелкие блоки. Каждый объект конфигурации можно включить в состав одной или нескольких подсистем. Таким образом, в терминах подсистем можно описать всю структуру прикладного решения.[11]

Объекты, которые были описаны выше, распределены на три подсистемы разработанной конфигурации: «графики», «клиенты и сотрудники», «финансы» (Рис. 2).



Рисунок 2 – Подсистемы конфигурации

Объекты конфигурации возможно разместить в одной или нескольких подсистемах, в которых они будут отображаться.

После того как созданы все основные объекты конфигурации, необходимо определить роли пользователей.

С помощью объекта *Роль* разработчик получает возможность описать набор прав на выполнение тех или иных действий над каждым объектом базы данных и над всей конфигурацией в целом.

Как правило, роли создаются отдельно для каждого вида деятельности, и каждому пользователю системы ставится в соответствие одна или несколько ролей.

Если пользователю поставлено в соответствие несколько ролей, предоставление доступа будет осуществляться по следующему алгоритму:

- если хотя бы в одной роли есть разрешение, то доступ будет открыт;
- если во всех ролях разрешение отсутствует, то доступ будет закрыт

[10].

При создании ролей исходят, как правило, из того, какие полномочия требуются различным группам пользователей на доступ к информации.

В конфигурации «1С:Школа танцев» созданы следующие роли.

### *1. Программист*

Программист обладает правами на все объекты и все виды объектов. Однако у него снято разрешение на интерактивное удаление для всех объектов. Это необходимо для того, чтобы программист случайно не мог удалить какой-либо объект базы данных.

Для того, чтобы программист мог работать с объектами, которые будут созданы после расстановки прав, для него задан параметр *Устанавливать права для новых объектов*.

### *2. Директор*

У роли Директор нет прав на доступ ко всем объектам, за исключением тех видов объектов конфигурации, для которых не создано ни одного объекта. Для таких видов объектов конфигурации остаются установленными полные права.

У роли Директор установлены права *Вывод* и *Просмотр* для всей конфигурации (права *Чтение* и *Использование* при этом устанавливаются автоматически).

Директор может видеть все справочники, документы и отчеты, а также обладает правами на *Добавление*, *Изменение* и *Удаление* объектов конфигурации. Однако он не может изменять или удалять такие документы, как «Запись на занятия», «Установить расписание тренеру», «Посещение занятий» и «Продажа абонементов», поскольку данный набор функций не входит в его обязанности.

### *3. Администратор*

У роли Администратор установлены все права на объекты конфигурации, относящиеся к подсистемам: *Клиенты*, *сотрудники*, *Расписание*. Однако с помощью фильтра объектов у подсистем внесены некоторые уточнения в установленные права.

В частности, для всех справочников запрещено право *Удаления* и снята видимость у документа *«Зарплата тренеров»*.

#### 4. Тренер

Аналогичным образом у роли Тренер установлены все права на объекты конфигурации к таким подсистемам, как Тренера и Расписание. При этом внесены некоторые исключения.

Тренер не может изменять, удалять и добавлять элементы справочников Тренера и Стили. Для того, чтобы тренер не мог изменить свою зарплату или начислить ее сверх нормы, для документа *«Зарплата тренерам»* установлено только право *Чтение*. Также снята видимость у документа *«Запись на занятия»*, поскольку данная функция не входит в его обязанности.

#### 5. Клиент

Роль Клиент обладает правом просмотра информации о тренерах, стилях и стоимости абонементов школы танцев. Также он может видеть свою историю посещений занятий и актуальное расписание на неделю, для этого установлено право Просмотр для документа *«Посещение занятий»* и отчета *«Расписание занятий по залам»*. При этом клиенту предоставлена возможность самостоятельной записи на занятия посредством документа *«Запись на занятия»*.

#### 6. Расчетчик

Для роли Расчетчик установлены все права на объекты конфигурации подсистемы Склад. Помимо этого, он имеет права на добавление, изменение и удаление документа *«Зарплата тренерам»*.

### **Выводы по главе два**

Выявлено, что важной особенностью платформы 1С: Предприятие является гибкость и открытость решений платформы.

Построена ER-диаграмма предметной области. Выделены сущности схемы метаданных: *«Стили»*, *«Сотрудники»*, *«Клиенты»*, *«Залы»* *«Организации»*. В *«1С:Школа танцев»* данные сущности представлены

объектом конфигурирования *Справочники*. Каждый справочник представляет собой список однородных объектов: клиенты, сотрудники и т.д.

Решение задач, поставленных в выпускной квалифицированной работе, реализовано с помощью предлагаемых платформой объектов *Документы*, *Регистры накопления*, *Регистры сведений*, *Отчеты*.



### 3 Возможности конфигурации «1С:Школа танцев»

#### 3.1 Подсистема «клиенты и сотрудники»

В данной подсистеме осуществляется работа связанная с клиентами и сотрудниками, а также учитывается прием клиентов (рис.3).

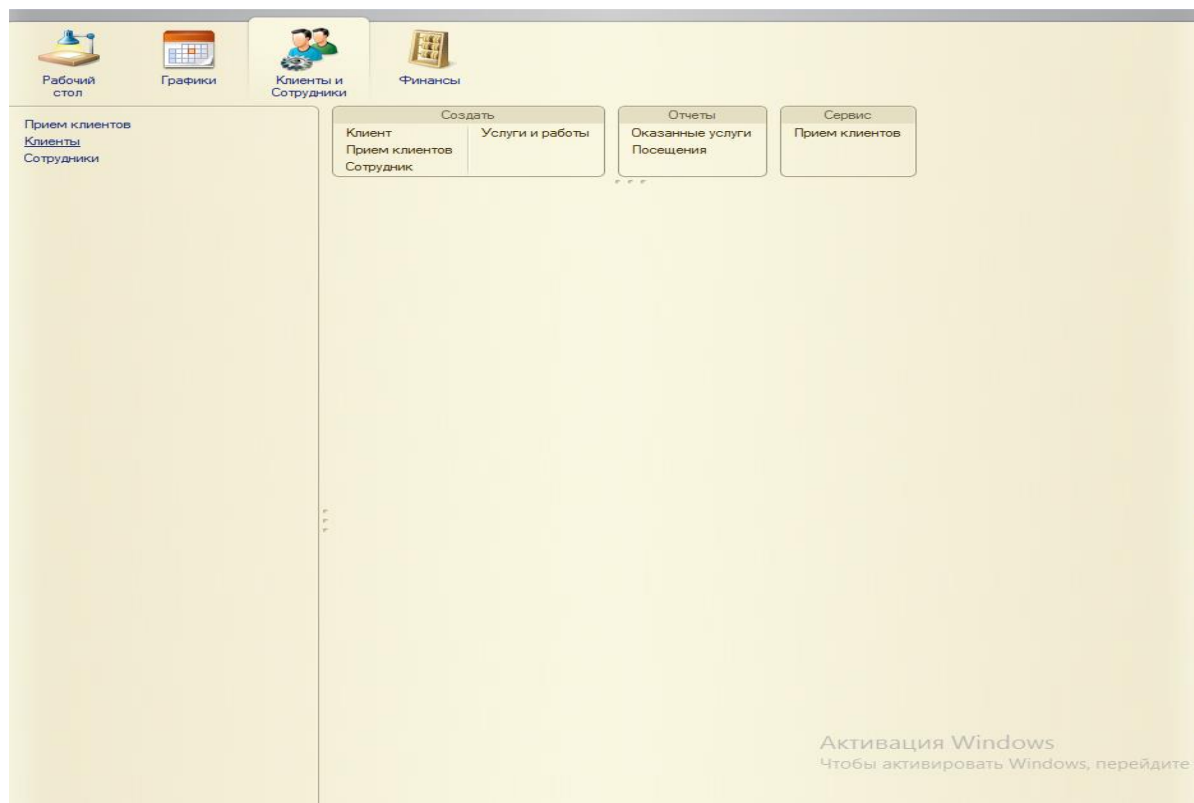


Рисунок 3 – Обзор подсистемы клиенты и сотрудники

В ходе многолетней практики и работы с клиентами в школе танцев, было выявлено что для клиента данной организации важен пол сотрудника, поэтому было предпринято разделение по полу признаку и учет данного фактора. Информация о личных данных сотрудника хранится в справочнике «сотрудники» (рис. 4).

Дундин Олег Геннадьевич (Сотрудник) (1С:Предприятие)

**Дундин Олег Геннадьевич (Сотрудник)**

Записать и закрыть

Код: 00001

ФИО: Дундин Олег Геннадьевич

Дата рождения: 01.03.1996 Пол физ. лица: Мужской

Контактная информация

Адрес:

Телефон:

email:

Комментарий:

Рисунок 4 – Форма элемента справочника «сотрудники»

В школе танцев «TROUBLE MAKERS» реализована система скидок. В форме элемента «клиенты» за выбор скидки отвечает поле «скидка»(рис. 5).

Казанцева Софья Андреевна (Клиент) (1С:Предприятие)

**Казанцева Софья Андреевна (Клиент)**

Записать и закрыть Печать

№ карты: 00000001

ФИО: Казанцева Софья Андреевна

Дата рождения: 01.03.2004 Пол физ. лица: Женский

Скидка: Постоянный клиент

Контактная информация

Адрес:

Телефон: +79514836767

email: cjyz.2004@mail.ru

Комментарий:

Рисунок 5 – Форма элемента справочника «клиенты»

С помощью документа «прием» мы можем отслеживать была ли оказана услуга клиенту, следить за оплатой данной услуги, а также выбрать сотрудника который будет исполнять ее (рис.6).

**Прием клиентов 0000000001 от 01.06.2017**

Провести и закрыть | Провести | Печать

Наряд  
 Номер: 0000000001    Дата: 01.06.2017

Сотрудник: Дундин Олег Генадьевич

Пациент  
 Клиент: Казанцева Софья Андреевна

Скидка: Постоянный клиент

Запись на прием  
 Время начала: 16:00    Время окончания: 17:00    1 ч.

Принят:

Оказанные услуги (работы)

N	Услуга (работа)	Количество	Цена	Цена со скидкой	Сумма
1	Хип-Хоп	1	1 700,00	1 615,00	1 615,00
					1 615,00

Оплата  
 Виды денежного расчета: Наличные

Номер квитанции об оплате: 1    Оплаченная сумма: 1 615,00    полностью оплачено

Рисунок 6 – Форма документа «Прием»

За фиксирование услуг, которые были предоставлены студией, отвечает отчет «Оказанные услуги» (рис.7), запрос отчета представлен в листинге 1. Пользователю предоставляется три выбора варианта отчета:

- по услугам
- услуги по клиенту
- услуги по сотрудникам

**Оказанные услуги**

Вариант отчета: Услуги по клиенту

Сформировать | Настройки...

Период: Прешлый месяц, до такой же даты  
 Клиент: Равно    Хохлова Яна Николаевна

**Оказанные услуги за период**

Параметры: Период: 01.05.2017 - 12.05.2017  
 Отбор: Клиент Равно "Хохлова Яна Николаевна"

Клиент	Сотрудник	Итого
Номер наряда		Сумма
Хохлова Яна Николаевна		1 700,00
000000002	Дундин Олег Генадьевич	1 700,00
Итого		1 700,00

**Выбор варианта отчета**

По услугам  
 Услуги по клиенту  
 Услуги по сотруднику

Выбрать    Отмена

Рисунок 7 – Форма отчета «оказанные услуги»

Листинг 1 – Запрос отчета «оказанные услуги»

**ВЫБРАТЬ**

ПриемНарядУслуги.Ссылка КАК Регистратор,  
 ПриемНарядУслуги.УслугаРабота,

СУММА (ПриемНарядУслуги.Количество) КАК Количество,  
СУММА (ПриемНарядУслуги.ЦенаСоСкидкой) КАК Цена,  
СУММА (ПриемНарядУслуги.Сумма) КАК Сумма,  
ПриемНарядУслуги.Ссылка.Номер,  
ПриемНарядУслуги.Ссылка.Клиент,  
ПриемНарядУслуги.Ссылка.Сотрудник

ИЗ

Документ.ПриемНаряд.Услуги КАК ПриемНарядУслуги

ГДЕ

ПриемНарядУслуги.Ссылка.Дата МЕЖДУ &НачалоПериода И &КонецПериода  
И ПриемНарядУслуги.Ссылка.Проведен = ИСТИНА

СТРУППИРОВАТЬ ПО

ПриемНарядУслуги.Ссылка,  
ПриемНарядУслуги.УслугаРабота,  
ПриемНарядУслуги.Ссылка.Номер,  
ПриемНарядУслуги.Ссылка.Клиент,  
ПриемНарядУслуги.Ссылка.Сотрудник

### 3.2 Подсистема «Финансы»

В данной подсистеме устанавливаются цены на услуги, учитывается возврат долгов клиентами, ведется вся финансовая составляющая компании(рис.8).

Подсистема разделена на панель навигации и панель действий. Тем временем панель навигации включает в себя:

- услуги
- возврат долга клиентом
- установка цен
- цены на услуги

Панель действий включает в себя функции «создать» и «отчеты», где мы можем просмотреть прайс лист а также видим все долги клиентов.

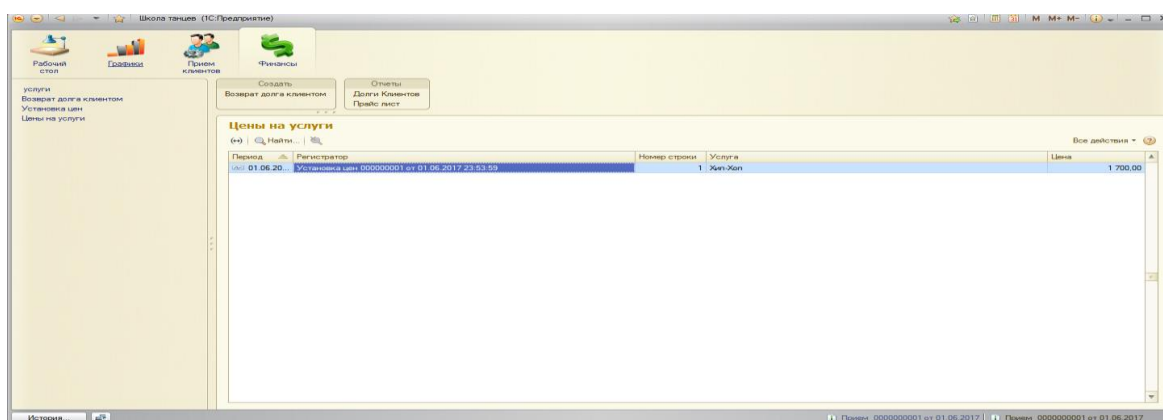


Рисунок 8 – Внешний вид подсистемы «Финансы»

Отчет, формирующий прайс лист, представлен на рисунке 9. Запрос отчета представлен в листинге 2.

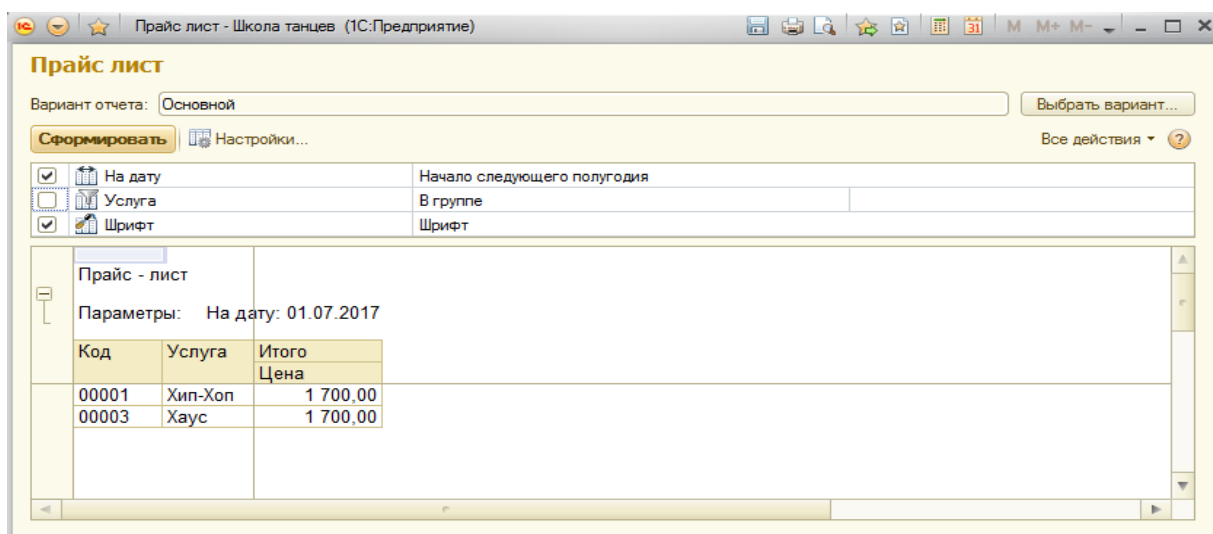


Рисунок 9 – Прайс лист на начало следующего полугодия

Листинг 2 – Запрос отчета «прайс лист»

**ВЫБРАТЬ**

ЦеныНаУслугиСрезПоследних . Услуга ,  
ЦеныНаУслугиСрезПоследних . Цена

**ИЗ**

РегистрСведений . ЦеныНаУслуги . СрезПоследних ( &Период , ) КАК

ЦеныНаУслугиСрезПоследних

**ГДЕ**

ЦеныНаУслугиСрезПоследних . Услуга . ВключатьВПрайс = ИСТИНА

### 3.3 Подсистема «Графики»

В данной подсистеме реализуются графики работ организации, клиентов, сотсавляются отчеты о днях рождения, присутствует сервис «уведомления».(рис 10).

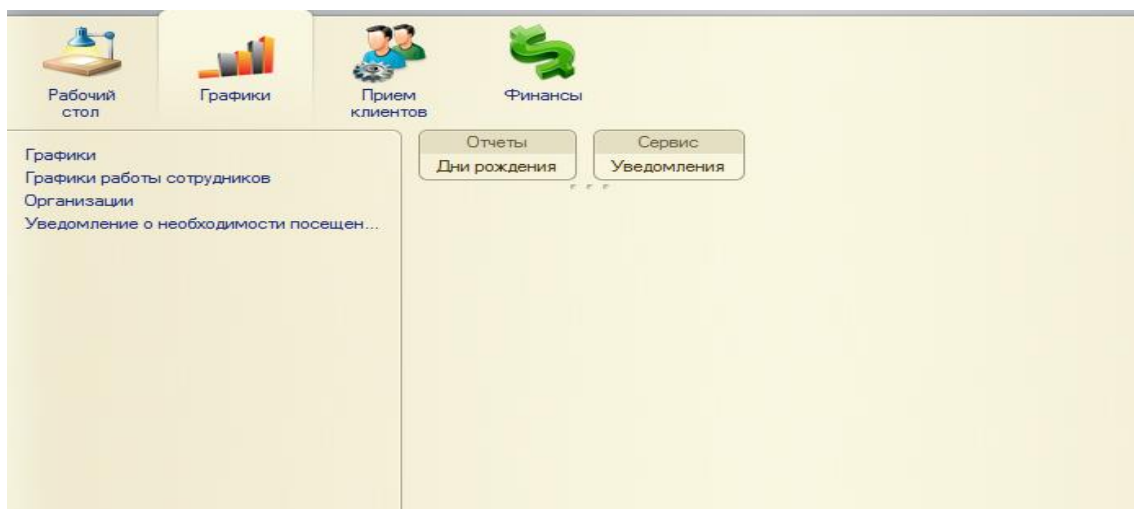


Рисунок 10 –Итерфейс подсистемы «графики»

Справочник организации содержит в себе информацию о предприятиях, с которыми ведется сотрудничество. Пример создания организации представлен на рисунке 11.

Организации (создание) (1С:Предприятие)

**Организации (создание)**

Записать и закрыть Все действия

Код:

Наименование:

Полное наименование:

График работы:  ...

Контактная информация

Фактический адрес:

Юридический адрес:

Телефон:

Должности

Директор:  ...

Главный бухгалтер:  ...

Рисунок 11 – создание организации

Пример графика работы представлен на рисунке 12.

нечетные, 4 часа (Графики) (1С:Предприятие)

**нечетные, 4 часа (Графики)**

Записать и закрыть Все действия

Код:

Наименование:

Добавить Все действия

N	Дни	Время начала	Время окончания
1	Понедельник	12:00	16:00
2	Среда	12:00	16:00
3	Пятница	12:00	16:00

Рисунок 12 – график работы

### 3.4 Рабочий стол

Рабочий стол представляет собой журнал записи на прием. Осуществлен выбор отображения по сотрудникам и залам. Журнал может отображать записи на:

- день
- неделю
- месяц

С помощью журнала реализуется запись клиента на прием, а также ведется учет посещений, журнал записи легко настраивается. Для облечения работы с журналом, табличные части окрашиваются в различные цвета (рис. 13).

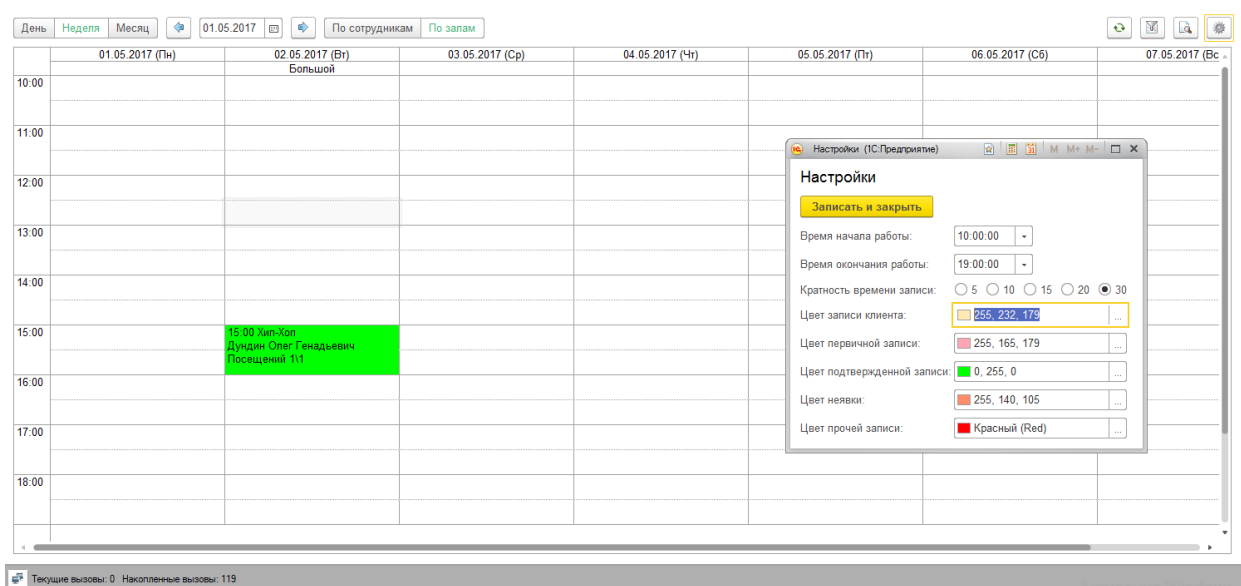


Рисунок 13 – Журнал записи на прием

### Выводы по главе три

Разработана конфигурация «1С:Школа танцев» для автоматизации деятельности танцевальной студии, в которой реализованы следующие задачи:

- ведение учета клиентской базы, посещаемости клиентов и их платежей;
- модуль «Склад» для ведения учета спортивного оборудования и инвентаря;
- предварительная запись клиентов на занятия;
- Журнал записи клиентов;

## **Заключение**

В дипломном проекте рассмотрена процедура проектирования конфигурации для автоматизации деятельности школы танцев. На примерах продемонстрирована актуальность разработки, проведен анализ устоявшихся на рынке систем, были выявлены достоинства и ограничения данных продуктов. В результате исследования рынка, обоснована потребность в создании нового программного обеспечения, удовлетворяющего деятельности танцевальной студии.

С помощью программного обеспечения, спроектированного в ходе работы, возможно вести учет клиентов, их посещаемость, присутствует функция предварительной записи на занятия. Программа позволяет следить за финансовой составляющей предприятия. В связи с гибкостью платформы, конфигурацию возможно настроить по требованию заказчика. Возможно добавление мобильных сервисов, что позволит повысить интерес к танцевальной студии.



## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Программа автоматизации танцевальной школы и студии. ЛистОк [Оф. сайт]// URL: <https://listokcrm.ru/> (дата обращения 06.02.2017).
- 2 Хеликс школа танцев система автоматизации [Оф. сайт]// URL: <http://www.dance1c.ru/> (дата обращения 06.02.2017).
- 3 Hollihop [Оф. сайт]// URL: <http://h-hope.ru/> (дата обращения 06.02.2017).
- 4 CRM-система для учебного центра. «Альфа-CRM» [Оф. сайт]// URL: <https://alfacrm.pro/> (дата обращения 06.02.2017).
- 5 Учет клиентов танцевальной студии. DanceStudio [Оф. сайт]// URL: <http://www.dance-soft.ru/> (дата обращения 06.02.2017).
- 6 Диго С.М. Базы данных: проектирование и использование: Учебник / С.М. Диго – М.: Финансы и статистика, 2005. – 592 с.: ил.
- 7 Радченко, М.Г. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы / М.Г. Радченко, Е.Ю. Хрусталева. – М.: ООО «1С-Публишинг», 2013. – 965 с.: ил.
- 8 Хрусталева, Е.Ю. Знакомство с разработкой мобильных приложений на платформе «1С:Предприятие 8» / Е.Ю. Хрусталева. – М.: ООО «1С-Публишинг», 2014. – 292 с.: ил. – (Библиотека разработчика).
- 9 Хрусталева, Е.Ю. Разработка сложных отчетов в «1С:Предприятии 8». Система компоновки данных / Е.Ю. Хрусталева. – 2-е изд. – М.: 1С-Публишинг, 2012. – 484 с.: ил. – (Библиотека разработчика).
- 10 Хрусталева, Е.Ю. Язык запросов «1С:Предприятия 8» / Е.Ю. Хрусталева. – М.: ООО «1С-Публишинг», 2013. – 369 с.: ил. – (Библиотека разработчика).
- 11 Виды связей между таблицами в базе данных. [Оф. сайт]// URL: <http://zametkinapolyah.ru/zametki-o-mysql/chast-3-2-vidy-svyazej-mezhdu-tablicami-v-baze-dannyx-svyazi-v-relyacionnyx-bazax-dannyx-otnosheniya-kortezhi-atributy.html> (дата обращения 22.03.2017).

12 Языки программирования. 1С:Предприятие [Оф. сайт]// URL: [http://life-prog.ru/1\\_3088\\_ER-modeli.html](http://life-prog.ru/1_3088_ER-modeli.html) (дата обращения: 24.03.2017).

13 Разработка и управление базами данных в СУБД MS Access 2007. Теоретические основы построения баз данных. Виды связей между таблицами [Оф. сайт]// URL: <http://online-academy.ru/demo/access/urok1/teor/teor4.htm> (дата обращения 20.02.2017).

14 Учебник по 1С:Предприятие [Оф. сайт]// URL: [http://www.mista.ru/tutor\\_1c/index.htm](http://www.mista.ru/tutor_1c/index.htm) (дата обращения 15.04.2016).

15 1С:Предприятие 8.х. Часть 3: Типы данных [Оф. сайт]// URL: <http://www.vr-online.ru/content/1spredpriyatie-8x-chast-3-tipy-dannyh-2370> (дата обращения 20.05.2016).

16 Введение в конфигурирование в системе «1С:Предприятие 8». Основные объекты. М.: ООО «1С-Учебный центр №3». – 2010. – 110 с.

17 1С:Автоматизированное составление расписания. Университет [Оф. сайт]// URL: [http://solutions.1c.ru/asp\\_univer/features](http://solutions.1c.ru/asp_univer/features) (дата обращения: 20.05.2016).

ПРИЛОЖЕНИЕ А  
ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Институт естественных и точных наук  
Кафедра «Математическое и компьютерное моделирование»

Разработка программного комплекса для прогнозирования  
потребления электроэнергии  
ТЕХНИЧЕСКОЕ ЗАДАНИЕ  
НА РАЗРАБОТКУ ПРОГРАММНОГО КОМПЛЕКСА «ПРОГНОЗ»  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ– 01.03.02.2017.072.10.000. ТЗ ВКР

Нормоконтролер,  
к.ф.-м.н., доцент каф. МиКМ,  
\_\_\_\_\_ Макаровских Т.А.  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Руководитель проекта,  
к.ф.-м.н., доцент каф. МиКМ,  
\_\_\_\_\_ Макаровских Т.А.  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Автор работы,  
Студент группы ЕТ-485  
\_\_\_\_\_ Маткин В.А.  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Челябинск, 2017

# 1 ВВЕДЕНИЕ

## 1.1 Наименование программного изделия

Полное наименование программы – Автоматизированная система для ведения учета танцевальной студии «1С:Школа танцев». Краткое наименование – «1С:Школа танцев».

## 2 Назначение и цели создания (развития) системы

### 2.1 Назначение системы

Система предназначена для автоматизации деятельности танцевальной студии «TROUBLE MAKERS» в соответствии с требованиями, зафиксированными в данном Техническом задании.

### 2.2 Цели создания системы

1. Предоставить возможность хранения в базе данных информации о клиентах, сотрудниках, стилях, видах предоставляемых услуг и расписании;
2. Система должна обеспечивать ведение учета посещений клиентов, проданных абонементов;
3. Обеспечить разграничение прав доступа пользователей к функциям программы и информации, хранимой в базе данных;
4. Возможность составления различных видов отчетов.

## 3 Характеристика объектов автоматизации

Объект автоматизации – танцевальная студия – организация среднего размера, предоставляющая спортивные и танцевальные услуги населению. Условия эксплуатации автоматизированной системы – доступ к СУБД с клиентского компьютера (рабочее место администратора).

Программное обеспечение должно быть установлено на персональный компьютер с операционной системой Windows.

## **4 Требования к системе**

### **4.1 Требования к системе в целом**

### **4.2 Требования к функциям и задачам, выполняемым системой**

Необходимо, чтобы ИС «1С:Школа танцев» позволяла решать следующие задачи:

- 1) возможность просмотра информации о клиентах, тренерах и стилях школы танцев, а также предоставляемых ими услугах;
- 2) установление расписания занятий для конкретного тренера;
- 3) возможность записи клиента на занятия;
- 4) учет поступления денежных средств от клиентов при покупке абонеента;
- 5) возможность ведения посещаемости клиентов;
- 6) возможность начисления заработной платы преподавателям танцевальной студии;
- 7) учет спортивного инвентаря и оборудования;
- 8) контроль качества работы тренеров;

### **4.3 Требования к видам обеспечения**

В информационной системе должен быть реализован понятный и простой интерфейс.

Также должны присутствовать следующие функции, выполняемые данной системой:

- Добавление записей в базу данных;
- Редактирование базы данных;
- Сортировка тренеров и клиентов по ФИО или дате рождения;
- Поиск клиента по ФИО;
- Формирование отчетов (выручка тренеров; рейтинг по количеству клиентов в разрезе тренеров и стилей; платежи клиентов; инвентарь, находящийся на складе; расписание занятий по залам).

#### **4.4 Требования квалификации**

Разработанной системой может пользоваться любое количество человек, включая администратора, руководителя организации, преподавателя и даже самого клиента. Дополнительного персонала, такого как IT-специалисты, не потребуется, поскольку организации будет отдан готовый программный продукт. Для работы с программой необходим базовый уровень владения ПК перечисленных выше пользователей.

#### **4.5 Требования к защите информации от несанкционированного доступа**

Для защиты информации от несанкционированного доступа в ИС «1С:Школа танцев» созданы следующие роли и назначения для каждой из них:

##### *1. Руководитель*

У руководителя организации установлены права *Вывод* и *Просмотр* для всей конфигурации.

Руководитель может видеть все справочники, документы и отчеты, а также обладает правами на *Добавление*, *Изменение* и *Удаление* объектов конфигурации. Однако он не может изменять или удалять такие документы, как «Запись на занятия», «Установить расписание тренеру», «Посещение занятий» и «Продажа абонементов», поскольку данный набор функций не входит в его обязанности.

##### *2. Администратор*

У администратора установлены все права на объекты конфигурации, относящиеся к подсистемам: *Клиенты*, *Тренера*, *Расписание*. Однако с помощью фильтра объектов у подсистем внесены некоторые уточнения в установленные права.

В частности, для всех справочников запрещено право *Удаления* и снята видимость у документа «*Зарплата тренеров*».

##### *3. Тренер*

У тренера установлены все права на объекты конфигурации к таким подсистемам, как Тренера и Расписание. При этом внесены некоторые исключения.

Тренер не может изменять, удалять и добавлять элементы справочников Тренера и Стили. Для того, чтобы тренер не мог изменить свою зарплату или начислить ее сверх нормы, для документа «*Зарплата тренерам*» установлено только право *Чтение*. Также снята видимость у документа «*Запись на занятия*», поскольку данная функция не входит в его обязанности.

#### *4. Клиент*

Клиент обладает правом просмотра информации о тренерах, стилях и стоимости абонементов школы танцев. Также он может видеть свою историю посещений занятий и актуальное расписание на неделю, для этого установлено право Просмотр для документа «*Посещение занятий*» и отчета «*Расписание занятий по залам*». При этом клиенту предоставлена возможность самостоятельной записи на занятия посредством документа «*Запись на занятия*».

#### *5. Расчетчик*

Для роли Расчетчик установлены все права на объекты конфигурации подсистемы Склад. Помимо этого, он имеет права на добавление, изменение и удаление документа «*Зарплата тренерам*».

Для всех сотрудников существует запрет возможности редактирования самого приложения.

### **4.6 Требования к надежности**

#### 4.6.1 Требования к надежному функционированию

Программа должна нормально функционировать при бесперебойной работе ЭВМ. При возникновении сбоя в работе аппаратуры восстановление нормальной работы программы должно производиться после:

- перезагрузки операционной системы;

- запуска исполняемого файла программы; повторного выполнения действий, потерянных до последнего сохранения информации в файл на диске.

#### 4.6.2 Контроль входной и выходной информации

Программа должна контролировать выбор пользователем пункта меню «Выход» и предупреждать о возможной потере несохраненных изменений.

#### 4.6.3 Время восстановления после отказа

Время восстановления после отказа должно состоять из:

- времени запуска пользователем исполняемого файла программы;
- времени повторного ввода потерянных данных.

#### 4.7 Требования к составу и параметрам технических средств

Программа должна корректно работать на следующем или совместимом с ним оборудовании:

- персональный компьютер с операционной системой Windows;
- принтер.

#### 4.8 Требования к стандартизации

Программный код 1С должен быть оформлен с учётом определённых норм, только в этом случае возможно получение сертификата совместимости продуктов 1С.

*Основные требования к конфигурации:*

1. Нежелательно использовать тип ЛюбаяСсылка для типизированных объектов метаданных;
2. Рекомендуется делать сортировки Справочников и Документов по имени;
3. Объекты с префиксом Удалить логично определять в конец списка;
4. Для метаданных со строковым типом лучше применять переменную длину строки;
5. Фиксированная длина оправдана если это определяется спецификой процедуры;



6. Реквизиты, формы и измерения должны быть расположены в дереве по проекту;

7. Индексирование допустимо в целях оптимизации отчётов и отбора;

8. При существенном снижении быстродействия количество индексаций нужно снизить.

#### *Общие правила для кода IC*

Для подчиненных объектов метаданных, таких как реквизиты, измерения, ресурсы рекомендуется не использовать имена, совпадающие с именами объектов-владельцев. Если нужно сформировать базу однотипных данных, следует делать это функционалом единого запроса. Проверку на наличие пустых строк надо делать по методу Пустой(). Ключевые слова должны быть написаны в заглавном исполнении. Рекомендуется чётко обозначать псевдоним полей.

#### *Оформление текстов запросов*

1. Все ключевые слова языка запросов пишутся заглавными буквами.

2. Рекомендуется указывать и необязательные конструкции запроса, прежде всего — явно назначать псевдонимы полям, в целях повышения наглядности текста запроса и «устойчивости» использующего его кода.

3. Следует обязательно указывать ключевое слово «КАК» перед псевдонимом поля источника.

4. Текст запроса должен быть структурирован, не следует писать запрос в одну строку, даже короткий. Текст запроса должен быть нагляден, поскольку это существенно улучшает его понимание другими разработчиками.

5. В запросы, сложные для понимания, в которых используются вложенные запросы, объединения или соединения рекомендуется вставлять комментарии.

6. При создании объекта Запрос рекомендуется указывать комментарии, для получения какой информации или каких иных целей будет использован данный запрос.

7. При программной «сборке» текста запроса рекомендуется комментировать все этапы его сборки.

8. В общем случае, при объединении в запросе результатов нескольких запросов следует использовать конструкцию «ОБЪЕДИНИТЬ ВСЕ», а не «ОБЪЕДИНИТЬ».

#### *Дополнительные правила для кода 1С*

Когда максимально возможная длина строки неизвестна, для хранения используются строковые реквизиты неограниченной длины. Нельзя заводить формы внутри раздела инициализации модуля. При разработке кода общего модуля и модулей объектов, которые должны быть доступны на сервере и во внешнем соединении, следует соблюдать следующие правила.

1. Запрещено использование объектов, имеющих тип данных, не доступный на сервере и во внешнем соединении:

2. Запрещено использование средств, отвечающих за диалог с пользователем:

3. Запрещается вызов экспортных процедур других общих модулей, у которых не установлен признак компиляции на сервере и/или во внешнем соединении.

4. Участки кода, в которых используются конструкции, не доступные на сервере или во внешнем соединении, должны выделяться соответствующими инструкциями препроцессору;

5. При написании кода модулей объектов, которые исполняются на сервере или доступны во внешнем соединении, недопустимо использовать переменные, процедуры и функции, которые определены в модуле обычного приложения и в модуле управляемого приложения.

### **5 Состав и содержание работ по созданию системы**

Процесс создания автоматизированной системы «1С:Школа танцев» можно разбить на несколько этапов.

1. Создать все необходимые объекты конфигурации (справочники, документы, регистры и т.д.);

2. Сформировать базу клиентов, тренеров, залов, абонементов и спортивного инвентаря;

3. Организовать полноценный учет для ведения посещаемости клиентов, их платежей и начисления заработной платы преподавателям.

4. Создать отчеты, отражающие качество работы как тренеров, так и всей танцевальной студии в целом.

5. Обеспечить разграничение прав пользователей;

6. Разработать интерфейс программы;

Тестирование.

## ПРИЛОЖЕНИЕ Б МОДУЛИ ОБЪЕКТОВ

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Институт естественных и точных наук  
Кафедра «Математическое и компьютерное моделирование»

Разработка программного комплекса для прогнозирования  
потребления электроэнергии  
ТЕХНИЧЕСКОЕ ЗАДАНИЕ  
НА РАЗРАБОТКУ ПРОГРАММНОГО КОМПЛЕКСА «ПРОГНОЗ»  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ– 01.03.02.2017.072.10.000. ТЗ ВКР

Нормоконтролер,  
к.ф.-м.н., доцент каф. МиКМ,  
\_\_\_\_\_ Макаровских Т.А.  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Руководитель проекта,  
к.ф.-м.н., доцент каф. МиКМ,  
\_\_\_\_\_ Макаровских Т.А.  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Автор работы,  
Студент группы ЕТ-485  
\_\_\_\_\_ Маткин В.А.  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Челябинск, 2017

## Б1. Модуль объекта документа «Установка цен»

```
Процедура ОбработкаПроведения(Отказ, Режим)
  //{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные
вручную изменения будут утеряны!!!
  Движения.ЦеныНаУслуги.Записывать = Истина;
  Для Каждого ТекСтрокаЦены Из Цены Цикл
    // регистр ЦеныНаУслуги
    Движение = Движения.ЦеныНаУслуги.Добавить ();
    Движение.Период = Дата;
    Движение.Услуга = ТекСтрокаЦены.Услуга;
    Движение.Цена = ТекСтрокаЦены.Цена;
  КонецЦикла;
  //}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

## Б2. Модуль объекта документа «Установка цен»

```
Процедура ОбработкаПроведения(Отказ, Режим)
  //{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные
вручную изменения будут утеряны!!!
  Движения.ИсторияПосещений.Записывать = Истина;
  // регистр ИсторияПосещений
  Движение = Движения.ИсторияПосещений.Добавить ();
  Движение.Период = Дата;
  Движение.Клиент = Клиент;

  Движения.ДолгиКлиентов.Записывать = Истина;
  // регистр ДолгиКлиентов Приход
  Движение = Движения.ДолгиКлиентов.Добавить ();
  Движение.Период = Дата;
  Движение.Клиент = Клиент;
  Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
  Движение.Сумма = Услуги.Итог("Сумма");

  Движения.ДолгиКлиентов.Записывать = Истина;
  // регистр ДолгиКлиентов Расход
  Движение = Движения.ДолгиКлиентов.Добавить ();
  Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
  Движение.Период = Дата;
  Движение.Клиент = Клиент;
  Движение.Сумма = ОплаченнаяСумма;

  Движения.УведомлениеОПолугодномПосещенииСтоматолога.Зап
исывать = Истина;
  Движение =
Движения.УведомлениеОПолугодномПосещенииСтоматолога.Добавить ();
  Движение.Период = Дата;
  Движение.Клиент = Клиент;
  Движение.Уведомлять = Истина;
```

```
    //}} __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ  
КонецПроцедуры
```

```
Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)  
    ЭтотОбъект.Сумма = Услуги.Итог("Сумма");  
КонецПроцедуры
```

### **Б3. Модуль объекта документа «Возврат Долга Клиентом»**

```
Процедура ОбработкаПроведения(Отказ, Режим)  
    //{{ __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ  
    // Данный фрагмент построен конструктором.  
    // При повторном использовании конструктора, внесенные  
вручную изменения будут утеряны!!!  
    Движения.ДолгиКлиентов.Записывать = Истина;  
    // регистр ДолгиКлиентов Расход  
    Движение = Движения.ДолгиКлиентов.Добавить();  
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
    Движение.Период = Дата;  
    Движение.Клиент = Клиент;  
    Движение.Сумма = Сумма;  
  
    //}} __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ  
КонецПроцедуры
```

### **Б3. Модуль объекта документа «Уведомление»**

```
Процедура ОбработкаПроведения(Отказ, Режим)  
    //{{ __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ  
    // Данный фрагмент построен конструктором.  
    // При повторном использовании конструктора, внесенные  
вручную изменения будут утеряны!!!  
  
    // регистр УведомлениеОПолугодномПосещенииСтоматолога  
    Движения.УведомлениеОПолугодномПосещенииСтоматолога.Зап  
исывать = Истина;  
    Для Каждого ТекСтрокаКлиенты Из Клиенты Цикл  
        Движение =  
Движения.УведомлениеОПолугодномПосещенииСтоматолога.Добавить();  
        Движение.Период = Дата;  
        Движение.Клиент = ТекСтрокаКлиенты.Клиент;  
        Движение.Уведомлять = Не НеУведомлять;  
    КонецЦикла;  
  
    //}} __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ  
КонецПроцедуры
```

### **Б4. Модуль объекта документа «Предварительная Запись»**

```
#Если Сервер Или ТолстыйКлиентОбычноеПриложение Или  
ВнешнееСоединение Тогда
```

#Область ОбработчикиСобытий

Процедура ОбработкаЗаполнения (ДанныеЗаполнения,  
СтандартнаяОбработка)

ТипДанныхЗаполнения = ТипЗнч (ДанныеЗаполнения) ;

Если ТипДанныхЗаполнения = Тип ("Структура") Тогда

ЗаполнитьЗначенияСвойств (ЭтотОбъект,  
ДанныеЗаполнения) ;

КонецЕсли;

ТипЗаписи =  
Перечисления.ТипыПредварительнойЗаписи.ЗаписьКлиента;

КонецПроцедуры

Процедура ОбработкаПроверкиЗаполнения (Отказ,  
ПроверяемыеРеквизиты)

Если ТипЗаписи =  
Перечисления.ТипыПредварительнойЗаписи.ЗаписьКлиента Тогда

ПроверяемыеРеквизиты.Удалить (ПроверяемыеРеквизиты.Найти ("Вид  
ПрочейЗаписи")) ;

Иначе

ПроверяемыеРеквизиты.Удалить (ПроверяемыеРеквизиты.Найти ("Кли  
ент")) ;

КонецЕсли;

Если ЗначениеЗаполнено (Сотрудник) Тогда

ПроверяемыеРеквизиты.Удалить (ПроверяемыеРеквизиты.Найти ("Каб  
инет")) ;

ИначеЕсли ЗначениеЗаполнено (Кабинет) Тогда

ПроверяемыеРеквизиты.Удалить (ПроверяемыеРеквизиты.Найти ("Сот  
рудник")) ;

КонецЕсли;

КонецПроцедуры

Процедура ПередЗаписью (Отказ, РежимЗаписи, РежимПроведения)

Если ОбменДанными.Загрузка Тогда

Возврат;

КонецЕсли;

Если Не РежимЗаписи =  
РежимЗаписиДокумента.ОтменаПроведения Тогда

```
РежимЗаписи = РежимЗаписиДокумента.Проведение;  
КонецЕсли;
```

КонецПроцедуры

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

```
// регистр ЗаписьНаПрием  
Движения.ЗаписьНаПрием.Записывать = Истина;  
Движение = Движения.ЗаписьНаПрием.Добавить();  
Движение.Период = Дата;  
Движение.Сотрудник = Сотрудник;  
Движение.Кабинет = Кабинет;  
Движение.ТипЗаписи = ТипЗаписи;  
Движение.ВидПрочейЗаписи = ВидПрочейЗаписи;  
Движение.ВремяНачала = ВремяНачала;  
Движение.ВремяОкончания = ВремяОкончания;  
Движение.Клиент = Клиент;  
Движение.Телефон = Телефон;  
Движение.Подтверждение = Подтверждение;  
Движение.Неявка = Неявка;  
Движение.Первичная = Первичная;  
Движение.Комментарий = Комментарий;
```

КонецПроцедуры

Процедура ПриКопировании(ОбъектКопирования)

```
ТипЗаписи =  
Перечисления.ТипыПредварительнойЗаписи.ЗаписьКлиента;
```

КонецПроцедуры

#КонецОбласти

#КонецЕсли



## ПРИЛОЖЕНИЕ В МОДУЛИ ФОРМ

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Институт естественных и точных наук  
Кафедра «Математическое и компьютерное моделирование»

Разработка программного комплекса для прогнозирования  
потребления электроэнергии  
ТЕХНИЧЕСКОЕ ЗАДАНИЕ  
НА РАЗРАБОТКУ ПРОГРАММНОГО КОМПЛЕКСА «ПРОГНОЗ»  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ– 01.03.02.2017.072.10.000. ТЗ ВКР

Нормоконтролер,  
к.ф.-м.н., доцент каф. МиКМ,  
\_\_\_\_\_ Макаровских Т.А.  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Руководитель проекта,  
к.ф.-м.н., доцент каф. МиКМ,  
\_\_\_\_\_ Макаровских Т.А.  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Автор работы,  
Студент группы ЕТ-485  
\_\_\_\_\_ Маткин В.А.  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Челябинск, 2017

## **V1. Модуль формы элемента «Клиенты»**

&НаКлиенте  
Процедура АК043У (Команда)

ТабДок = Печать\_АК043У (Новый Структура ("Клиент",  
Объект.Ссылка) );  
ПодготовкаВПечатиТабДок (ТабДок, " карта клиента: " +  
Объект.Ссылка) ;

КонецПроцедуры

&НаСервере  
Функция Печать\_АК043У (Структура)

Возврат  
ПодготовитьТабДокДляПечати\_АК\_043\_3У (Структура) ;

КонецФункции

&НаКлиенте  
Процедура Договор (Команда)

ТабДок = ПечатьДоговора (Новый Структура ("Клиент",  
Объект.Ссылка) );  
ПодготовкаВПечатиТабДок (ТабДок, "Договор на оказание  
услуг с: " + Объект.Ссылка) ;

КонецПроцедуры

&НаСервере  
Функция ПечатьДоговора (Структура)  
Возврат ПодготовитьДляПечатиТабДок\_Договор (Структура) ;  
КонецФункции

## **V2. Модуль формы списка «УслугиРаботы»**

&НаСервере  
Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)  
Список.Параметры.УстановитьЗначениеПараметра ("Период",  
ТекущаяДата ( ) ) ;  
КонецПроцедуры

## **V3. Модуль формы выбора «УслугиРаботы»**

&НаСервере  
Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)  
Список.Параметры.УстановитьЗначениеПараметра ("Период", ТекущаяДата ());  
КонецПроцедуры

#### **В4. Модуль формы документа «ПриемНаряд»**

&НаКлиенте  
Процедура флПриемПриИзменении (Элемент)  
    УправлениеВидимостью ();  
КонецПроцедуры

&НаСервере  
Процедура УправлениеВидимостью ()

    Элементы.ОказанныеУслугиРаботы.Видимость =  
Объект.флПрием;  
    Элементы.Оплата.Видимость = Объект.флПрием;  
    Элементы.УслугиЦенаСоСкидкой.Видимость = Не  
Объект.Скидка.Пустая (); //Цену со скидкой видно только если  
указана скидка

    РассчитатьИтоговуюСуммуПоДокументу ();

КонецПроцедуры

&НаСервере  
Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)

    УправлениеВидимостью ();  
    ЗаполнитьРазницуВременНачалаИКонца ();  
    РассчитатьИтоговуюСуммуПоДокументу ();  
    ИзменениеВидаДенежногоРасчета ();  
    ПолучениеПоясненияДолгКлиента ();

КонецПроцедуры

&НаКлиенте  
Процедура ИзмененоВремяПриема (Элемент)

    ЗаполнитьРазницуВременНачалаИКонца ();

КонецПроцедуры

&НаСервере  
Процедура ЗаполнитьРазницуВременНачалаИКонца ()

    ВремяНаПрием = РазницаВременСтрокой (Объект.ВремяНачала,  
Объект.ВремяОкончания);

КонецПроцедуры

&НаКлиенте  
Процедура УслугиКоличествоПриИзменении (Элемент)  
    ПересчитатьСтроку (ТекущийЭлемент.ТекущиеДанные);  
КонецПроцедуры

```
&НаКлиенте
Процедура УслугиЦенаПриИзменении (Элемент)
    ПересчитатьСтроку (ТекущийЭлемент.ТекущиеДанные);
КонецПроцедуры
```

```
&НаКлиенте
Процедура СкидкаПриИзменении (Элемент)
```

```
    ПересчетВсейТаблицы ();
    УправлениеВидимостью ();
```

```
КонецПроцедуры
```

```
&НаКлиенте
Процедура ПересчетВсейТаблицы ()
```

```
    Для Каждого СтрокаТЧ Из Объект.Услуги Цикл
```

```
        ПересчитатьСтроку (СтрокаТЧ);
```

```
    КонецЦикла;
```

```
КонецПроцедуры
```

```
&НаКлиенте
Процедура ЗаполнитьЦены (Строка)
```

```
    Строка.Цена = ПолучитьЦену (Объект.Дата,
Строка.УслугаРабота);
```

```
    Строка.ЦенаСоСкидкой = ПолучитьЦену (Объект.Дата,
Строка.УслугаРабота, Объект.Скидка);
```

```
КонецПроцедуры
```

```
&НаСервере
Функция ПолучитьЦену (Дата, Услуга, Скидка = Неопределено)
```

```
    Возврат ПолучитьЦенуНаУслугу (Дата, Услуга, Скидка);
```

```
КонецФункции
```

```
&НаКлиенте
Процедура ПересчитатьСтроку (Строка)
```

```
    Скидка = Объект.Скидка;
```

```
    Если Скидка.Пустая() Тогда //не указана
        Строка.ЦенаСоСкидкой = Строка.Цена;
```

```
    Иначе
```

```
        Строка.ЦенаСоСкидкой = Строка.Цена - Строка.Цена *
ПолучитьПроцентСкидки (Объект.Скидка) / 100;
```

```

        КонечЕсли;

        Строка.Сумма = Строка.Количество *
Строка.ЦенаСоСкидкой;

    КонечПроцедуры

&НаСервере
Функция ПолучитьПроцентСкидки (Скидка)

        Возврат Скидка.Процент;

    КонечФункции

&НаКлиенте
Процедура УслугиУслугаРаботаПриИзменении (Элемент)

        ТекДанные = ТекущийЭлемент.ТекущиеДанные;

        ЗаполнитьЦены (ТекДанные);

        Если ТекДанные.Количество = 0 Тогда
            ТекДанные.Количество = 1; //По умолчанию 1 т.к.
большинство услуг в количестве 1
        КонечЕсли;

        ПересчитатьСтроку (ТекДанные);

    КонечПроцедуры

&НаКлиенте
Процедура КлиентПриИзменении (Элемент)

        ПодставитьСкидкуКлиента ();
        СкидкаПриИзменении (Неопределено);
        ПолучениеПоясненияДолгКлиента ();

    КонечПроцедуры

&НаСервере
Процедура ПодставитьСкидкуКлиента ()

        //Подставим скидку присвоенную этому Клиенту
        Клиент = Объект.Клиент;
        Если Объект.Скидка.Пустая () Тогда
            Объект.Скидка = Клиент.Скидка;
        КонечЕсли;

    КонечПроцедуры

&НаКлиенте
Процедура УслугиПриИзменении (Элемент)
        РассчитатьИтоговуюСуммуПоДокументу ();

```

```

КонецПроцедуры

&НаСервере
Процедура РассчитатьИтоговуюСуммуПоДокументу ( )

    СуммаПоДокументу = Объект.Услуги.Итог ("Сумма");
    Элементы.УслугиСумма.ТекстПодвала = Формат
(СуммаПоДокументу, "ЧЦ=15; ЧДЦ=2");

    ПолучениеПояснениеКОплате ( );

КонецПроцедуры

&НаСервере
Процедура ПолучениеПояснениеКОплате ( )

    СуммаПоДокументу = Объект.Услуги.Итог ("Сумма");

    Разница = СуммаПоДокументу - Объект.ОплаченнаяСумма;

    Если Разница = 0 Тогда
        ПояснениеПоОплате = "полностью оплачено";
    ИначеЕсли Разница > 0 Тогда
        ПояснениеПоОплате = "недоплачено (долг) : " +
Формат (Разница, "ЧЦ=15; ЧДЦ=2");
    Иначе //Если Разница < 0 Тогда
        ПояснениеПоОплате = "переплачено : " + Формат ( -
Разница, "ЧЦ=15; ЧДЦ=2");
    КонецЕсли;

КонецПроцедуры

&НаСервере
Процедура ПолучениеПоясненияДолгКлиента ( )

    //ПояснениеДолгКлиента =
    МоментВремени = Новый МоментВремени(Объект.Дата,
Объект.Ссылка);
    Остатки =
РегистрыНакопления.ДолгиКлиентов.Остатки(МоментВремени, Новый
Структура ("Клиент", Объект.Клиент), , "Сумма");

    Если Остатки.Количество() Тогда
        ПояснениеДолгКлиента = "долг: " +
Формат(Остатки[0].Сумма, "ЧЦ=15; ЧДЦ=2");
    Иначе
        ПояснениеДолгКлиента = "";
    КонецЕсли;

КонецПроцедуры

&НаКлиенте
Процедура ОплаченнаяСуммаПриИзменении (Элемент)

```

```

        ПолучениеПояснениеКОплате ( ) ;
    КонечПроцедуры

    &НаКлиенте
    Процедура ПослеЗаписи(ПараметрыЗаписи)
        Оповестить ("ИзмененыДанныеПриема") ;
    КонечПроцедуры

    &НаКлиенте
    Процедура АК043У(Команда)

        ТабДок = Печать_АК043У(Новый Структура ("Клиент",
    Объект.Клиент) ) ;
        ПодготовкаВПечатиТабДок(ТабДок, "Амбулаторная карта
    043/у клиента: " + Объект.Клиент) ;

    КонечПроцедуры

    &НаСервере
    Функция Печать_АК043У(Структура)

        Возврат
    ПодготовитьТабДокДляПечати_АК_043_3У(Структура) ;

    КонечФункции

    &НаКлиенте
    Процедура Договор(Команда)

        ТабДок = ПечатьДоговора(Новый Структура ("Клиент,
    Сотрудник", Объект.Клиент, Объект.Сотрудник) ) ;
        ПодготовкаВПечатиТабДок(ТабДок, "Договор на оказание
    услуг с: " + Объект.Клиент) ;

    КонечПроцедуры

    &НаСервере
    Функция ПечатьДоговора(Структура)
        Возврат ПодготовитьДляПечатиТабДок_Договор(Структура) ;
    КонечФункции

    &НаКлиенте
    Процедура ВидыДенежногоРасчетаПриИзменении(Элемент)
        ИзменениеВидаДенежногоРасчета ( ) ;
    КонечПроцедуры

    &НаСервере
    Процедура ИзменениеВидаДенежногоРасчета ( )

        Элементы.НомерКвитанцииОбОплате.Видимость =
    Объект.ВидыДенежногоРасчета =
    Перечисления.ВидыДенежногоРасчета.Наличные ;

```

КонецПроцедуры

&НаКлиенте

Процедура ОбработкаВыбора (ВыбранноеЗначение, ИсточникВыбора)  
ДобавитьУслугу (ВыбранноеЗначение);

КонецПроцедуры

&НаКлиенте

Процедура ДобавитьУслугу (Услуга)

Элементы.Услуги.ДобавитьСтроку ();  
ТекДанные = ТекущийЭлемент.ТекущиеДанные;

ТекДанные.УслугаРабота = Услуга;  
ТекДанные.Количество = 1;

ЗаполнитьЦены (ТекДанные);  
ПересчитатьСтроку (ТекДанные);

КонецПроцедуры

&НаКлиенте

Процедура Подбор (Команда)

ФормаПодбора =  
ПолучитьФорму ("ОбщаяФорма.ФормаПодбораУслуг", , ЭтаФорма);  
ФормаПодбора.ЗакрыватьПриВыборе = Ложь;  
АдресЗапасовВХранилище =  
ФормаПодбора.ОткрытьМодально ();

КонецПроцедуры

## **В5. Модуль формы отчета «ДневнойЛистокУчета»**

&НаСервере

Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)

Дата = ТекущаяДата ();  
СформироватьОтчет ();

КонецПроцедуры

&НаКлиенте

Процедура Сформировать (Команда)

СформироватьОтчет ();

КонецПроцедуры

&НаСервере

Процедура СформироватьОтчет ()

ТабДок.Очистить ();  
ТабДок.НачатьАвтогруппировкуСтрок ();



```

Макет =
Отчеты.ДневнойЛистокУчета.ПолучитьМакет ("ДневнойЛистокУчета");

//Получение данных для вывода в отчет
ПостроительЗапроса = Новый ПостроительЗапроса;
ПостроительЗапроса.Текст = "ВЫБРАТЬ
|
ПриемНарядУслуги.Ссылка.Номер КАК НомерНаряда,
|
ПриемНарядУслуги.Ссылка.Сотрудник КАК Сотрудник,
|
ПриемНарядУслуги.Ссылка.Клиент КАК Клиент,
|
ПриемНарядУслуги.УслугаРабота КАК УслугаРабота,
|
СУММА (ПриемНарядУслуги.Сумма) КАК Сумма,
| ПриемНарядУслуги.Ссылка
КАК Регистратор
| ИЗ
|
Документ.ПриемНаряд.Услуги КАК ПриемНарядУслуги
| ГДЕ
|
ПриемНарядУслуги.Ссылка.Дата МЕЖДУ НАЧАЛОПЕРИОДА (&НаДату,
ДЕНЬ) И КОНЕЦПЕРИОДА (&НаДату, ДЕНЬ)
| И
ПриемНарядУслуги.Ссылка.Проведен = ИСТИНА
|
| СГРУППИРОВАТЬ ПО
| ПриемНарядУслуги.Ссылка,
|
ПриемНарядУслуги.Ссылка.Номер,
|
ПриемНарядУслуги.Ссылка.Сотрудник,
|
ПриемНарядУслуги.Ссылка.Клиент,
|
ПриемНарядУслуги.УслугаРабота
|
| УПОРЯДОЧИТЬ ПО
| Сотрудник,
| НомерНаряда
| ИТОГИ
| МИНИМУМ (НомерНаряда) ,
| СУММА (Сумма)
| ПО
| ОБЩИЕ,
| Сотрудник,
| Регистратор,
| УслугаРабота
| АВТОУПОРЯДОЧИВАНИЕ";
ПостроительЗапроса.Параметры.Вставить ("НаДату", Дата);
ПостроительЗапроса.ЗаполнитьНастройки ();

```

```

ПостроительЗапроса.Выполнить ( ) ;
Результат = ПостроительЗапроса.Результат;

Область = Макет.ПолучитьОбласть ("Шапка") ;
Область.Параметры.Дата = Дата;
ТабДок.Вывести (Область) ;

ИтогиОбщие =
Результат.Выбрать (ОбходРезультатаЗапроса.ПоГруппировкам) ;

//Получим и заполним общий итог
ИтогиОбщие.Следующий ( ) ;
ОбластьПодвал = Макет.ПолучитьОбласть ("Подвал") ;
ЗаполнитьЗначенияСвойств (ОбластьПодвал.Параметры,
ИтогиОбщие) ;

ВыборкаВрачи =
ИтогиОбщие.Выбрать (ОбходРезультатаЗапроса.ПоГруппировкам) ;
Пока ВыборкаВрачи.Следующий ( ) Цикл

Область = Макет.ПолучитьОбласть ("СтрокаВрач") ;
ЗаполнитьЗначенияСвойств (Область.Параметры,
ВыборкаВрачи) ;
ТабДок.Вывести (Область, ВыборкаВрачи.Уровень ( ) ) ;

ТабДок.Вывести (Макет.ПолучитьОбласть ("ШапкаТЧ"),
ВыборкаВрачи.Уровень ( ) ) ;

ОбластьПодвалТЧ =
Макет.ПолучитьОбласть ("ПодвалТЧ") ;

ЗаполнитьЗначенияСвойств (ОбластьПодвалТЧ.Параметры,
ВыборкаВрачи) ;

ВыборкаДокумент =
ВыборкаВрачи.Выбрать (ОбходРезультатаЗапроса.ПоГруппировкам) ;
Пока ВыборкаДокумент.Следующий ( ) Цикл
Область =
Макет.ПолучитьОбласть ("СтрокаТЧКлиент") ;
ЗаполнитьЗначенияСвойств (Область.Параметры,
ВыборкаДокумент) ;
ТабДок.Вывести (Область,
ВыборкаДокумент.Уровень ( ) ) ;

ВыборкаУслуги =
ВыборкаДокумент.Выбрать (ОбходРезультатаЗапроса.ПоГруппировкам) ;
Пока ВыборкаУслуги.Следующий ( ) Цикл
Область =
Макет.ПолучитьОбласть ("СтрокаТЧУслуга") ;

```

```
        ЗаполнитьЗначенияСвойств (Область.Параметры, ВыборкаУслуги);
        ТабДок.Вывести (Область,
ВыборкаУслуги.Уровень ());
        КонецЦикла;
```

```
        КонецЦикла;
```

```
        ТабДок.Вывести (ОбластьПодвалТЧ,
ВыборкаВрачи.Уровень ());
```

```
        КонецЦикла;
```

```
        ТабДок.Вывести (ОбластьПодвал);
```

```
        ТабДок.ЗакончитьАвтогруппировкуСтрок ();
```

```
        ТабДок.ПоказатьУровеньГруппировокСтрок (1);
```

```
КонецПроцедуры
```

## **В6. Модуль формы отчета «ДниРождения»**

```
&НаСервере
```

```
Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)
```

```
        Отчет.НаДату = ТекущаяДата ();
```

```
        СформироватьОтчет ();
```

```
КонецПроцедуры
```

```
&НаКлиенте
```

```
Процедура Сформировать (Команда)
```

```
        СформироватьОтчет ();
```

```
КонецПроцедуры
```

```
&НаСервере
```

```
Процедура СформироватьОтчет ()
```

```
        ТабДок.Очистить ();
```

```
        ТабДок.НачатьАвтогруппировкуСтрок ();
```

```
        Макет =
```

```
Отчеты.ДниРождения.ПолучитьМакет ("ДниРождения");
```

```
        НаДату = Отчет.НаДату;
```

```
        //Получение данных для вывода в отчет
```

```
        ПостроительЗапроса = Новый ПостроительЗапроса;
```

```
        ПостроительЗапроса.Текст = "ВЫБРАТЬ
```

```
        |   ""Сотрудники"" КАК Тип,
```

```

Сотрудники
    | Сотрудники.Ссылка,
    | Сотрудники.ДатаРождения
    | ИЗ
    | Справочник.Сотрудники КАК
    | ГДЕ
    | Сотрудники.ДатаРождения
    <> ДАТАВРЕМЯ(1, 1, 1)
    | И Сотрудники.ЭтоГруппа =
    | И ВЫБОР
    |
    | КОГДА
    ДЕНЬГОДА (Сотрудники.ДатаРождения) - ДЕНЬГОДА (&ДатаСегодня) >= 0
    |
    | ТОГДА
    ДЕНЬГОДА (Сотрудники.ДатаРождения) - ДЕНЬГОДА (&ДатаСегодня)
    |
    | ИНАЧЕ 366 +
    ДЕНЬГОДА (Сотрудники.ДатаРождения) - ДЕНЬГОДА (&ДатаСегодня)
    |
    | КОНЕЦ < 10
    |
    | ОБЪЕДИНИТЬ ВСЕ
    |
    | ВЫБРАТЬ
    | "Клиенты",
    | Клиенты.Ссылка,
    | Клиенты.ДатаРождения
    | ИЗ
    | Справочник.Клиенты КАК
    | ГДЕ
    | Клиенты.ДатаРождения <>
    ДАТАВРЕМЯ(1, 1, 1)
    | И Клиенты.ЭтоГруппа =
    | И ВЫБОР
    |
    | КОГДА
    ДЕНЬГОДА (Клиенты.ДатаРождения) - ДЕНЬГОДА (&ДатаСегодня) >= 0
    |
    | ТОГДА
    ДЕНЬГОДА (Клиенты.ДатаРождения) - ДЕНЬГОДА (&ДатаСегодня)
    |
    | ИНАЧЕ 366 +
    ДЕНЬГОДА (Клиенты.ДатаРождения) - ДЕНЬГОДА (&ДатаСегодня)
    |
    | КОНЕЦ < 10";
    ПостроительЗапроса.Параметры.Вставить ("ДатаСегодня",
НаДату);
    ПостроительЗапроса.ЗаполнитьНастройки();
    ПостроительЗапроса.Выполнить();
    Результат = ПостроительЗапроса.Результат;

    ТаблицаДнейРождений = Результат.Выгрузить();

    МассивТипов = Новый Массив;
    МассивТипов.Добавить(Тип("Дата"));
    ПараметрыДаты = Новый
КвалификаторыДаты(ЧастиДаты.Дата);

```

```

ДопустимыеТипыДата = Новый ОписаниеТипов (МассивТипов, ,
ПараметрыДаты);

ТаблицаДнейРождений.Колонки.Добавить ("ДатаСледующегоДняР
ождения", ДопустимыеТипыДата);

Для Каждого СтрокаТЗ Из ТаблицаДнейРождений Цикл
    //НаДату
    День = День (СтрокаТЗ.ДатаРождения);
    Месяц = Месяц (СтрокаТЗ.ДатаРождения);
    Если Месяц = 2 И День = 29 Тогда //29 февраля
        День = 28; //празднуем 28
    КонецЕсли;
    ДатаВЭтомГоду = Дата (Год (НаДату), Месяц, День);
    Если ДатаВЭтомГоду < НаДату Тогда
        СтрокаТЗ.ДатаСледующегоДняРождения =
ДобавитьМесяц (ДатаВЭтомГоду, 12);
    Иначе
        СтрокаТЗ.ДатаСледующегоДняРождения =
ДатаВЭтомГоду;
    КонецЕсли;
КонецЦикла;

МенеджерВременныхТаблиц = Новый
МенеджерВременныхТаблиц;
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ * ПОМЕСТИТЬ ДниРождения ИЗ
&ВнешнийИсточник КАК ВнешнийИсточник";
Запрос.МенеджерВременныхТаблиц =
МенеджерВременныхТаблиц;
Запрос.УстановитьПараметр ("ВнешнийИсточник",
ТаблицаДнейРождений);
Запрос.Выполнить ();

Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
| ДниРождения.Ссылка КАК Объект,
| ДниРождения.Ссылка.Адрес КАК Адрес,
| ДниРождения.Ссылка.Телефон КАК
Телефон,
| ДниРождения.Ссылка.Пол КАК Пол,
| ДниРождения.ДатаРождения,
|
ДниРождения.ДатаСледующегоДняРождения,
|
РАЗНОСТЬДАТ (ДниРождения.ДатаРождения,
ДниРождения.ДатаСледующегоДняРождения, ГОД) КАК ИсполняетсяЛет,
| РАЗНОСТЬДАТ (&ДатаСегодня,
ДниРождения.ДатаСледующегоДняРождения, ДЕНЬ) КАК ДнейОсталось,
| ДниРождения.Тип КАК Тип
| ИЗ
| ДниРождения КАК ДниРождения
| ГДЕ

```

```

|         РАЗНОСТЬДАТ (&ДатаСегодня,
ДниРождения.ДатаСледующегоДняРождения, ДЕНЬ) < 6
|
| УПОРЯДОЧИТЬ ПО
|     Тип УБЫВ,
|     ДнейОсталось,
|     Объект
| ИТОГИ ПО
|     Тип,
|     ДнейОсталось
| АВТОУПОРЯДОЧИВАНИЕ";
    Запрос.МенеджерВременныхТаблиц =
МенеджерВременныхТаблиц;
    Запрос.УстановитьПараметр ("ДатаСегодня", НаДату);
    Результат = Запрос.Выполнить ();

    Область = Макет.ПолучитьОбласть ("Шапка");
    Область.Параметры.ДатаСегодня = Формат
(НаДату, "ДФ=DD");
    ТабДок.Вывести (Область);

    ВыборкаТип =
Результат.Выбрать (ОбходРезультатаЗапроса.ПоГруппировкам);
    Пока ВыборкаТип.Следующий () Цикл

        Область = Макет.ПолучитьОбласть ("СтрокаТип");
        ЗаполнитьЗначенияСвойств (Область.Параметры,
ВыборкаТип);
        ТабДок.Вывести (Область, ВыборкаТип.Уровень ());

        ВыборкаДнейОсталось =
ВыборкаТип.Выбрать (ОбходРезультатаЗапроса.ПоГруппировкам);
        Пока ВыборкаДнейОсталось.Следующий () Цикл

            Область =
Макет.ПолучитьОбласть ("СтрокаДнейОсталось");
            Если ВыборкаДнейОсталось.ДнейОсталось = 0
Тогда
                Область.Параметры.Текст = "Сегодня";
            ИначеЕсли ВыборкаДнейОсталось.ДнейОсталось =
1 Тогда
                Область.Параметры.Текст = "Завтра";
            Иначе
                Область.Параметры.Текст = Формат (НаДату
+ ВыборкаДнейОсталось.ДнейОсталось*24*60*60, "ДФ=dd.ММ") + " (" +
ВыборкаДнейОсталось.ДнейОсталось + " дн.)";
            КонецЕсли;

            ТабДок.Вывести (Область,
ВыборкаДнейОсталось.Уровень ());

```

```

        ВыборкаДетали =
ВыборкаДнейОсталось.Выбрать (ОбходРезультатаЗапроса.ПоГруппировка
М) ;
        Пока ВыборкаДетали.Следующий() Цикл
            Область =
Макет.ПолучитьОбласть ("СтрокаДетали") ;

            ЗаполнитьЗначенияСвойств (Область.Параметры, ВыборкаДетали) ;
                Область.Параметры.ГодРождения = Формат
(Год (ВыборкаДетали.ДатаРождения) , "ЧГ=") ;
                ТабДок.Вывести (Область,
ВыборкаДетали.Уровень ()) ;
                КонецЦикла;

        КонецЦикла;

        КонецЦикла;

        ТабДок.ЗакончитьАвтогруппировкуСтрок ();

        КонецПроцедуры

```

## **В7. Модуль формы обработки «ПриемКлиентов»**

```

&НаСервере
Процедура ОбновитьДанныеТабДокумент ()

        ТабличныйДокумент.Очистить ();

        Макет = Обработки.ПриемКлиентов.ПолучитьМакет ("Макет") ;

        НачПериода = Объект.Период.ДатаНачала;
        КонПериода = Объект.Период.ДатаОкончания;

        //Подготовка данных Дата, Врач, График
        ПостроительЗапроса = Новый ПостроительЗапроса;

        ДатаПериода = НачПериода;
        Счетчик = 0;
        ПолныйТекстЗапроса = "";
        Пока ДатаПериода < КонПериода Цикл
            Счетчик = Счетчик + 1;
            ИмяПараметраДата = "НаДату" + Счетчик;
            ИмяПараметраНечетныйДень = "НечетныйДень" +
Счетчик;

            Если Не ПустаяСтрока (ПолныйТекстЗапроса) Тогда
                ПолныйТекстЗапроса = ПолныйТекстЗапроса + "
                |      ОБЪЕДИНИТЬ ВСЕ
                |      ";
            КонецЕсли;
            ТекстЗапроса = "ВЫБРАТЬ

```

```

|      &НаДату КАК Дата,
|
ГрафикиРаботыСотрудниковСрезПоследних.Сотрудник КАК
Сотрудник,
|
ГрафикиРасписаниеРаботы.ВремяНачала КАК ВремяНачала,
|
ГрафикиРасписаниеРаботы.ВремяОкончания КАК ВремяОкончания
|ИЗ
|
РегистрСведений.ГрафикиРаботыСотрудников.СрезПоследних (&НаДа
ту, ) КАК ГрафикиРаботыСотрудниковСрезПоследних
|      ЛЕВОЕ СОЕДИНЕНИЕ
Справочник.Графики.РабочиеДни КАК ГрафикиРасписаниеРаботы
|      ПО
ГрафикиРаботыСотрудниковСрезПоследних.График =
ГрафикиРасписаниеРаботы.Ссылка
|ГДЕ
|      ВЫБОР
|      КОГДА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Нечетные)
|      И
&НечетныйДень
|      ТОГДА ИСТИНА
|      КОГДА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Четные)
|      И (НЕ
&НечетныйДень)
|      ТОГДА ИСТИНА
|      КОГДА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Понедельник)
|      И
ДЕНЬНЕДЕЛИ (&НаДату) = 1
|      ТОГДА ИСТИНА
|      КОГДА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Вторник)
|      И
ДЕНЬНЕДЕЛИ (&НаДату) = 2
|      ТОГДА ИСТИНА
|      КОГДА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Среда)
|      И
ДЕНЬНЕДЕЛИ (&НаДату) = 3
|      ТОГДА ИСТИНА
|      КОГДА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Четверг)

```



ДЕНЬНЕДЕЛИ (&НаДату) = 4		И
		ТОГДА ИСТИНА
ГрафикиРасписаниеРаботы.Дни =		КОГДА
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Пятница)		
ДЕНЬНЕДЕЛИ (&НаДату) = 5		И
		ТОГДА ИСТИНА
ГрафикиРасписаниеРаботы.Дни =		КОГДА
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Суббота)		
ДЕНЬНЕДЕЛИ (&НаДату) = 6		И
		ТОГДА ИСТИНА
ГрафикиРасписаниеРаботы.Дни =		КОГДА
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Воскресенье)		
ДЕНЬНЕДЕЛИ (&НаДату) = 7		И
		ТОГДА ИСТИНА
		КОНЕЦ
		И &НаДату В
		(ВЫБРАТЬ
		&НаДату КАК
Дата		ИЗ
Справочник.Графики.РабочиеДни КАК		ГрафикиРасписаниеРаботы
ПРАВОЕ СОЕДИНЕНИЕ Константы КАК Константы		ПО
ГрафикиРасписаниеРаботы.Ссылка =		
Константы.РабочаяОрганизация.ГрафикРаботы		ГДЕ
		ВЫБОР
		КОГДА
ГрафикиРасписаниеРаботы.Дни =		
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Нечетные)		
&НечетныйДень		И
		ТОГДА
ИСТИНА		КОГДА
ГрафикиРасписаниеРаботы.Дни =		
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Четные)		
(НЕ &НечетныйДень)		И
		ТОГДА
ИСТИНА		КОГДА
ГрафикиРасписаниеРаботы.Дни =		
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Понедельник)		

```

ДЕНЬНЕДЕЛИ (&НаДату) = 1
ИСТИНА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Вторник)
ДЕНЬНЕДЕЛИ (&НаДату) = 2
ИСТИНА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Среда)
ДЕНЬНЕДЕЛИ (&НаДату) = 3
ИСТИНА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Четверг)
ДЕНЬНЕДЕЛИ (&НаДату) = 4
ИСТИНА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Пятница)
ДЕНЬНЕДЕЛИ (&НаДату) = 5
ИСТИНА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Суббота)
ДЕНЬНЕДЕЛИ (&НаДату) = 6
ИСТИНА
ГрафикиРасписаниеРаботы.Дни =
ЗНАЧЕНИЕ (Перечисление.ВидыДней.Воскресенье)
ДЕНЬНЕДЕЛИ (&НаДату) = 7
ИСТИНА
КОНЕЦ)
{ ГДЕ
ГрафикиРаботыСотрудниковСрезПоследних.Сотрудник.* КАК
Врач}";
ТекстЗапроса = СтрЗаменить (ТекстЗапроса,
"&НаДату", "&" + ИмяПараметраДата);

```

```
ТекстЗапроса = СтрЗаменить(ТекстЗапроса,  
"&НечетныйДень", "&" + ИмяПараметраНечетныйДень);  
ПолныйТекстЗапроса = ПолныйТекстЗапроса +  
ТекстЗапроса;
```

```
ПостроительЗапроса.Параметры.Вставить(ИмяПараметраДата,  
ДатаПериода);
```

```
ПостроительЗапроса.Параметры.Вставить(ИмяПараметраНечетныйДе  
нь, Булево (День(ДатаПериода) % 2));
```

```
ДатаПериода = ДатаПериода + 24*60*60;  
КонецЦикла;
```

```
ПолныйТекстЗапроса = ПолныйТекстЗапроса + "  
| УПОРЯДОЧИТЬ ПО Дата, Сотрудник  
| ИТОГИ МИНИМУМ(ВремяНачала),  
МАКСИМУМ(ВремяОкончания)  
| ПО ОБЩИЕ, Дата, Сотрудник  
| АВТОУПОРЯДОЧИВАНИЕ";
```

```
ПостроительЗапроса.Текст = ПолныйТекстЗапроса;
```

```
Если Не Объект.Сотрудник.Пустая() Тогда  
НовыйОтборВрач =  
ПостроительЗапроса.Отбор.Добавить("Сотрудник");  
НовыйОтборВрач.Использование = Истина;  
НовыйОтборВрач.Значение = Объект.Сотрудник;  
КонецЕсли;
```

```
ПостроительЗапроса.Выполнить();
```

```
Результат = ПостроительЗапроса.Результат;
```

```
ОбщийИтог =  
Результат.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);  
Если ОбщийИтог.Следующий() Тогда  
РабочееВремя_Макс = ОбщийИтог.ВремяОкончания;  
РабочееВремя_Мин = ОбщийИтог.ВремяНачала  
Иначе  
Возврат;  
КонецЕсли;
```

```
ВременнойИнтервал = 30*60;  
ЧислоИнтервалов = (РабочееВремя_Макс -  
РабочееВремя_Мин) / ВременнойИнтервал;
```

```
ТабличныйДокумент.НачатьАвтогруппировкуСтрок();
```

```
//Вывод шапки
```

```

        Область = Макет.ПолучитьОбласть ("Шапка | ШапкаЛ");
        Область.Параметры.Период =
ПредставлениеПериода (НачалоДня (НачПериода),
КонецДня (КонПериода), "ФП = Истина");
        ТабличныйДокумент.Вывести (Область, 1);
        //
        Время = РабочееВремя_Мин;
        Пока Время < РабочееВремя_Макс Цикл
            Область = Макет.ПолучитьОбласть ("Шапка |
Столбец");
            Область.Параметры.Время = Формат (Время,
"ДФ=чч:мм");
            ТабличныйДокумент.Присоединить (Область, 1);

            Время = Время + ВременнойИнтервал;
        КонецЦикла;

        ВыборкаДата =
ОбщийИтог.Выбрать (ОбходРезультатаЗапроса.ПоГруппировкам);
        Пока ВыборкаДата.Следующий () Цикл

            //Вывод группировка день
            Область = Макет.ПолучитьОбласть ("СтрокаЖ |
ШапкаЛ");
            Область.Параметры.Текст = Формат
(ВыборкаДата.Дата, "ДФ=dd.ММ.уууу") + "
("+ДеньНеделиСтрокой (ВыборкаДата.Дата) +)";

            ТабличныйДокумент.Вывести (Область, 1);

            Для Сч = 1 По ЧислоИнтервалов Цикл
                Область = Макет.ПолучитьОбласть ("СтрокаЖ |
Столбец");
                Область.Параметры.Знач = "";
                ТабличныйДокумент.Присоединить (Область, 1);
            КонецЦикла;

            ВыборкаВрачи =
ВыборкаДата.Выбрать (ОбходРезультатаЗапроса.ПоГруппировкам);
            Пока ВыборкаВрачи.Следующий () Цикл

                Область = Макет.ПолучитьОбласть ("Строка |
ШапкаЛ");
                Область.Параметры.Сотрудник =
ВыборкаВрачи.Сотрудник;
                Область.Параметры.Сотрудник =
ВыборкаВрачи.Сотрудник;
                ТабличныйДокумент.Вывести (Область, 2);

                //Присоединение
                Время = РабочееВремя_Мин;
                Пока Время < РабочееВремя_Макс Цикл

```

```

        Если ЭтоРабочийИнтервалВремени
(ВыборкаВрачи, Время, Время + ВременнойИнтервал) Тогда

                ЗаписьИнтервала =
ПолучитьЗаписьИнтервала (ВыборкаВрачи, Время, Время +
ВременнойИнтервал) ;

                Область =
Макет.ПолучитьОбласть ("Строка | Столбец") ;
                Если ЗаписьИнтервала = Неопределено
Тогда
                        СтруктураРасшифровка = Новый
Структура ;

                СтруктураРасшифровка.Вставить ("Дата", ВыборкаВрачи.Дата) ;

                СтруктураРасшифровка.Вставить ("Сотрудник",
ВыборкаВрачи.Сотрудник) ;

                СтруктураРасшифровка.Вставить ("Время", Время) ;
                Область.Параметры.Документ =
СтруктураРасшифровка ;
                Область.Параметры.Знач = "";
                Иначе
                Если ЗаписьИнтервала.флПрием
Тогда
                        Область =
Макет.ПолучитьОбласть ("Строка | ИдетПрием") ;
                        КонецЕсли ;

                ЗаполнитьЗначенияСвойств (Область.Параметры,
ЗаписьИнтервала) ;
                Область.Параметры.Знач =
ЗаписьИнтервала.Клиент ;
                КонецЕсли ;
                Иначе
                Область =
Макет.ПолучитьОбласть ("Строка | НерабочееИнтервал") ;
                КонецЕсли ;

                ТабличныйДокумент.Присоединить (Область,
2) ;

                Время = Время + ВременнойИнтервал ;

        КонецЦикла ;

        КонецЦикла ;

        ДатаПериода = НачалоДня (ДатаПериода + 24*60*60) ;
        КонецЦикла ;

```

```

ТабличныйДокумент.ЗакончитьАвтогруппировкуСтрок ();

ТабличныйДокумент.Вывести
(Макет.ПолучитьОбласть ("Подвал" ));

//пройдемся по документу и объединим ячейки

Для Строка = 1 По ТабличныйДокумент.ВысотаТаблицы Цикл
    Для Столбец = 1 По ТабличныйДокумент.ШиринаТаблицы
- 1 Цикл
        ТекстЯчейки =
ТабличныйДокумент.Область (Строка, Столбец) .Текст;
        Если Не ПустаяСтрока (ТекстЯчейки) Тогда

            СтолбецНачало = Столбец;
            Пока ТекстЯчейки =
ТабличныйДокумент.Область (Строка, Столбец + 1) .Текст Цикл
                Столбец = Столбец + 1;
            КонечЦикла;

            ОбъедОбласть =
ТабличныйДокумент.Область (Строка, СтолбецНачало, Строка,
Столбец) ;

            ОбъедОбласть.Объединить ();

        КонечЕсли;

    КонечЦикла;
КонечЦикла;

//Фиксация шапки
ТабличныйДокумент.ФиксацияСлева = 4;
ТабличныйДокумент.ФиксацияСверху = 2;

КонечПроцедуры

&НаСервере
Функция ЭтоРабочийИнтервалВремени (Выборка, НачИнтервала,
КонИнтервала)

    Если НачИнтервала < Выборка.ВремяНачала ИЛИ
КонИнтервала > Выборка.ВремяОкончания Тогда
        Возврат Ложь; //Вне графика врача (время не
рабочее);
    КонечЕсли;

    ВыборкаДетали =
Выборка.Выбрать (ОбходРезультатаЗапроса.ПоГруппировкам) ;
    Пока ВыборкаДетали.Следующий () Цикл
        Если НачИнтервала >= ВыборкаДетали.ВремяНачала И
КонИнтервала <= ВыборкаДетали.ВремяОкончания Тогда
            Возврат Истина; //входит в интервал рабочего
времени

```

```

        КонечЕсли;
    КонечЦикла;

    Возврат Ложь;

КонечФункции

&НаСервере
Функция ПолучитьЗаписьИнтервала (Выборка, НачИнтервала,
КонИнтервала)

    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
        | ПриемНаряд.Ссылка КАК Документ,
        | ПриемНаряд.флПрием,
        | ПриемНаряд.Клиент
    | ИЗ
        | Документ.ПриемНаряд КАК ПриемНаряд
    | ГДЕ
        | ПриемНаряд.Проведен = ИСТИНА
        | И ПриемНаряд.Сотрудник = &Сотрудник
        | И ПриемНаряд.Дата МЕЖДУ
НАЧАЛОПЕРИОДА (&Дата, ДЕНЬ) И КОНЕЦПЕРИОДА (&Дата, ДЕНЬ)
        | И ПриемНаряд.ВремяНачала <=
&НачИнтервала
        | И ПриемНаряд.ВремяОкончания >=
&КонИнтервала";
    Запрос.УстановитьПараметр ("Дата", Выборка.Дата);
    Запрос.УстановитьПараметр ("Сотрудник",
Выборка.Сотрудник);
    Запрос.УстановитьПараметр ("НачИнтервала",
НачИнтервала);
    Запрос.УстановитьПараметр ("КонИнтервала",
КонИнтервала);

    Результат = Запрос.Выполнить ();

    Если Результат.Пустой() Тогда
        Возврат Неопределено;
    Иначе
        Возврат Результат.Выгрузить () [0];
    КонечЕсли;

КонечФункции

&НаКлиенте
Процедура ПриОткрытии (Отказ)

    Объект.Период.Вариант =
ВариантСтандартногоПериода.Следующие7Дней;
    ОбновитьДанныеТабДокумент ();

КонечПроцедуры

```

&НаКлиенте  
Процедура ОбработкаОповещения (ИмяСобытия, Параметр,  
Источник)

Если ИмяСобытия = "ИзмененыДанныеПриема" Тогда  
ОбновитьДанныеТабДокумент ();  
КонецЕсли;

КонецПроцедуры

&НаКлиенте  
Процедура ТабличныйДокументВыбор (Элемент, Область,  
СтандартнаяОбработка)

Расшифровка = Область.Расшифровка;

Если ТипЗнч(Расшифровка) = Тип ("Структура") Тогда  
СтандартнаяОбработка = Ложь;  
СформироватьИОткрытьНаряд (Расшифровка);  
КонецЕсли;

КонецПроцедуры

&НаСервере  
Процедура СформироватьИОткрытьНаряд (Данные)

НовыйДокумент = Документы.ПриемНаряд.СоздатьДокумент ();  
ФормаДокумента =  
НовыйДокумент.ПолучитьФорму ("Документ.ПриемНаряд.ФормаОбъекта");

ДокументНаряд = ФормаДокумента.Объект;  
ЗаполнитьЗначенияСвойств (ДокументНаряд, Данные);

ДокументНаряд.ВремяНачала = Данные.Время;  
ДокументНаряд.ВремяОкончания =  
ДокументНаряд.ВремяНачала + 30\*60;

ФормаДокумента.Открыть ();

КонецПроцедуры

&НаКлиенте  
Процедура ПериодПриИзменении (Элемент)  
ОбновитьДанныеТабДокумент ();  
КонецПроцедуры

&НаКлиенте  
Процедура Обновить (Команда)  
ОбновитьДанныеТабДокумент ();  
КонецПроцедуры

&НаКлиенте



Процедура Печать (Команда)

```
ТабДок = ПодготовитьТабДокДляПечати();  
ПодготовкаВПечатиТабДок (ТабДок, "Расписание за период:  
" + ПредставлениеПериода (Объект.Период.ДатаНачала,  
Объект.Период.ДатаОкончания, "ФП=Истина"));
```

КонецПроцедуры

&НаСервере

Функция ПодготовитьТабДокДляПечати()

```
ТабДокДляПечати = Новый ТабличныйДокумент;  
ТабДокДляПечати.Вывести (ТабличныйДокумент);
```

```
Возврат ТабДокДляПечати;
```

КонецФункции

&НаКлиенте

Процедура ВрачПриИзменении (Элемент)

```
ОбновитьДанныеТабДокумент ();
```

КонецПроцедуры

## **V8 Модуль формы обработки «Уведомления»**

&НаСервере

Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)

```
АвтоЗаголовок = Ложь;
```

```
ОбновитьДанные ();
```

КонецПроцедуры

&НаКлиенте

Процедура Уведомить (Команда)

```
МассивВыделенныхСтрок =  
Элементы.Уведомления.ВыделенныеСтроки;  
СоздатьДокументУведомление (МассивВыделенныхСтрок);
```

КонецПроцедуры

&НаСервере

Процедура

```
СоздатьДокументУведомление (КлиентыМассивИдентификаторов,  
НеУведомлять = Ложь)
```

```
НовыйДокумент =  
Документы.Уведомление.СоздатьДокумент ();
```

```
ФормаДокумента =  
НовыйДокумент.ПолучитьФорму ("Документ.Уведомление.ФормаОбъекта")  
;
```

```
НовыйДокумент = ФормаДокумента.Объект;
```

```
НовыйДокумент.НеУведомлять = НеУведомлять;
```

```
Если НеУведомлять Тогда
```

```
    НовыйДокумент.Текст = "Не уведомлять никогда";
```

```
Иначе
```

```
    НовыйДокумент.Текст = "Уведомлен (ы)";
```

```
КонецЕсли;
```

```
НовыйДокумент.Дата = НачалоДня(ТекущаяДата());
```

```
Для Каждого ИдентификаторСтроки Из
```

```
КлиентыМассивИдентификаторов Цикл
```

```
    НоваяСтрока = НовыйДокумент.Клиенты.Добавить();
```

```
    НоваяСтрока.Клиент =
```

```
Уведомления.НайтиПоИдентификатору(ИдентификаторСтроки).Клиент;
```

```
КонецЦикла;
```

```
ФормаДокумента.Открыть();
```

```
КонецПроцедуры
```

```
&НаКлиенте
```

```
Процедура НеУведомлять(Команда)
```

```
МассивВыделенныхСтрок =
```

```
Элементы.Уведомления.ВыделенныеСтроки;
```

```
СоздатьДокументУведомление(МассивВыделенныхСтрок,  
Истина);
```

```
КонецПроцедуры
```

```
&НаСервере
```

```
Процедура ОбновитьДанные()
```

```
ПостроительЗапроса = Новый ПостроительЗапроса;
```

```
ПостроительЗапроса.Текст = "ВЫБРАТЬ
```

```
|
```

```
УведомлениеОПолугодномПосещенииСтоматологаСрезПоследних.Клиент  
КАК Клиент,
```

```
|
```

```
УведомлениеОПолугодномПосещенииСтоматологаСрезПоследних.Период  
КАК ДатаПоследнегоПосещенияУведомления,
```

```
|
```

```
РАЗНОСТЬДАТ(НАЧАЛОПЕРИОДА(УведомлениеОПолугодномПосещенииСтоматологаСрезПоследних.Период,  
ДЕНЬ), &ТекДата, ДЕНЬ) - 182 КАК  
Дней
```

```
|ИЗ
```

```

|
РегистрСведений.УведомлениеОПолугодномПосещенииСтоматолога.С
резПоследних(&ТекДата, ) КАК
УведомлениеОПолугодномПосещенииСтоматологаСрезПоследних
| ГДЕ
|
РАЗНОСТЬДАТ(УведомлениеОПолугодномПосещенииСтоматологаСрезПо
следних.Период, &ТекДата, ДЕНЬ) > 182
| И
УведомлениеОПолугодномПосещенииСтоматологаСрезПоследних.Уведомля
ть = ИСТИНА
|
| УПОРЯДОЧИТЬ ПО
| Дней УБЫВ";
ПостроительЗапроса.Параметры.Вставить("ТекДата",
КонецДня(ТекущаяДата()));
ПостроительЗапроса.Выполнить();

ДеревоРезультат =
ПостроительЗапроса.Результат.Выгрузить(ОбходРезультатаЗапроса.По
ГруппировкамСИерархией);

ЗначениеВРеквизитФормы(ДеревоРезультат, "Уведомления");

КонецПроцедуры

&НаКлиенте
Процедура ОбработкаОповещения(ИмяСобытия, Параметр,
Источник)

Если ИмяСобытия = "ИзмененыДанныеУведомлений" Тогда
ОбновитьДанные();
КонецЕсли;

КонецПроцедуры

```

## **В10. Модуль формы обработки «ЖурналЗаписиНаПрием»**

```

#Область ПрограммныйИнтерфейс

// Обработка оповещения
&НаКлиенте
Процедура УстановитьОтборЗавершение(СтруктураВозврата,
Параметры) Экспорт

Если ЗначениеЗаполнено(СтруктураВозврата) Тогда
СписокЗначенийОтбора.Добавить(СтруктураВозврата);
ЗаголовокДекорацииОтбор = СтрЗаменить(НСтр("ru =
'%1: %2'", "%1", СотрудникКабиентПереключатель);
ЗаголовокДекорацииОтбор =
СтрЗаменить(ЗаголовокДекорацииОтбор, "%2",
СписокЗначенийОтбора);

```

```

        Элементы.ДекорацияОтбор.Заголовок =
ЗаголовокДекорацииОтбор;

        КонецЕсли;
        ОбновитьТабличныйДокумент ();

        Элементы.ГруппаОтбор.Видимость =
ЗначениеЗаполнено (СписокЗначенийОтбора);

        КонецПроцедуры

// Обработка оповещения
&НаКлиенте
Процедура НастройкаЗавершение (СтруктураВозврата, Параметры)
Экспорт

        ОбновитьИнтерфейс ();
        ЗагрузитьНастройки ();
        ОбновитьТабличныйДокумент ();

        КонецПроцедуры

#КонецОбласти

#Область ОбработчикиСобытийФормы

&НаСервере
Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)

        Если Параметры.Свойство ("АвтоТест") Тогда // Возврат
при получении формы для анализа.
                Возврат;
        КонецЕсли;

        ЗагрузитьНастройки ();

        Период.ДатаНачала =
? (ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"Период") = Неопределено, ТекущаяДатаСеанса (),
ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"Период"));

        Период.Переключатель =
? (ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ПериодПереключатель") = Неопределено, "День",
ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ПериодПереключатель"));

        СотрудникКлиент.Переключатель =
? (ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"СотрудникКлиент.Переключатель") = Неопределено, "Сотрудник",
ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"СотрудникКлиент.Переключатель"));

        Если Период.Переключатель = "День" Тогда

```

```

        Период.ДатаНачала      =
НачалоДня (Период.ДатаНачала) ;
        Период.ДатаОкончания =
КонецДня (Период.ДатаНачала) ;
        ИначеЕсли ПериодПереключатель = "Неделя" Тогда
        Период.ДатаНачала      =
НачалоНедели (Период.ДатаНачала) ;
        Период.ДатаОкончания =
КонецНедели (Период.ДатаНачала) ;
        ИначеЕсли ПериодПереключатель = "Месяц" Тогда
        Период.ДатаНачала      =
НачалоМесяца (Период.ДатаНачала) ;
        Период.ДатаОкончания =
КонецМесяца (Период.ДатаНачала) ;
        КонецЕсли;

```

```

        Заголовок = СформироватьЗаголовокФормы (Период,
ПериодПереключатель) ;

```

```

        Элементы.ГруппаОтбор.Видимость =
ЗначениеЗаполнено (СписокЗначенийОтбора) ;

```

```

        ОбновитьТабличныйДокумент () ;

```

```

КонецПроцедуры

```

```

&НаКлиенте
Процедура ПриОткрытии (Отказ)

```

```

        Элементы.ТабличныйДокумент.ТекущаяОбласть =
ТабличныйДокумент.Область (3, 2) ;

```

```

КонецПроцедуры

```

```

&НаКлиенте
Процедура ОбработкаОповещения (ИмяСобытия, Параметр,
Источник)

```

```

        Если ИмяСобытия = "Запись_ПредварительнаяЗапись"
        Или ИмяСобытия = "Удаление_ПредварительнаяЗапись"
        Или ИмяСобытия =
"Запись_ПредварительныйГрафикРаботыСотрудника" Тогда

```

```

        ОбновитьТабличныйДокумент () ;
        УстановитьВидимостьКонтекстногоМеню () ;

```

```

        КонецЕсли;

```

```

КонецПроцедуры

```

```

#КонецОбласти

```

```

#Область ОбработчикиСобытийЭлементовФормы

```

&НаКлиенте

Процедура ПериодПереключательПриИзменении (Элемент)

```
      Если ПериодПереключатель = "День" Тогда
        Период.ДатаНачала =
НачалоДня (Период.ДатаНачала) ;
        Период.ДатаОкончания =
КонецДня (Период.ДатаНачала) ;
      ИначеЕсли ПериодПереключатель = "Неделя" Тогда
        Период.ДатаНачала =
НачалоНедели (Период.ДатаНачала) ;
        Период.ДатаОкончания =
КонецНедели (Период.ДатаНачала) ;
      ИначеЕсли ПериодПереключатель = "Месяц" Тогда
        Период.ДатаНачала =
НачалоМесяца (Период.ДатаНачала) ;
        Период.ДатаОкончания =
КонецМесяца (Период.ДатаНачала) ;
      КонецЕсли;
```

```
      Заголовок = СформироватьЗаголовокФормы (Период,
ПериодПереключатель) ;
```

```
      ОбновитьТабличныйДокумент () ;
```

```
      СохранитьНастройки () ;
```

КонецПроцедуры

&НаКлиенте

Процедура ДатаНачалаПриИзменении (Элемент)

```
      Если ПериодПереключатель = "День" Тогда
        Период.ДатаНачала = НачалоДня (Период.ДатаНачала) ;
        Период.ДатаОкончания =
КонецДня (Период.ДатаНачала) ;
      ИначеЕсли ПериодПереключатель = "Неделя" Тогда
        Период.ДатаНачала =
НачалоНедели (Период.ДатаНачала) ;
        Период.ДатаОкончания =
КонецНедели (Период.ДатаНачала) ;
      ИначеЕсли ПериодПереключатель = "Месяц" Тогда
        Период.ДатаНачала =
НачалоМесяца (Период.ДатаНачала) ;
        Период.ДатаОкончания =
КонецМесяца (Период.ДатаНачала) ;
      КонецЕсли;
```

```
      Заголовок = СформироватьЗаголовокФормы (Период,
ПериодПереключатель) ;
```

ОбновитьТабличныйДокумент ( ) ;

СохранитьНастройки ( ) ;

КонецПроцедуры

&НаКлиенте

Процедура СотрудникКабиентПереключательПриИзменении (Элемент)

СписокЗначенийОтбора.Очистить ( ) ;

Элементы.ГруппаОтбор.Видимость =  
ЗначениеЗаполнено (СписокЗначенийОтбора) ;

ОбновитьТабличныйДокумент ( ) ;

СохранитьНастройки ( ) ;

КонецПроцедуры

&НаКлиенте

Процедура ТабличныйДокументВыбор (Элемент, Область,  
СтандартнаяОбработка)

Если Не

ЗначениеЗаполнено (Элемент.ТекущаяОбласть.Расшифровка) Тогда

ОткрытьФорму ("Документ.ПредварительнаяЗапись.Форма.ФормаДоку  
мента", Новый Структура ("ЗначенияЗаполнения",  
ПолучитьПараметрыОткрытияПредварительнойЗаписи ( ) ), ЭтаФорма) ;

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ТабличныйДокументОбработкаРасшифровки (Элемент,  
Расшифровка, СтандартнаяОбработка)

СтандартнаяОбработка = Ложь ;

Если ТипЗнч (Расшифровка) =

Тип ("ДокументСсылка.ПредварительнаяЗапись") Тогда

ПараметрыФормы = Новый Структура ;

ПараметрыФормы.Вставить ("Ключ", Расшифровка) ;

ОткрытьФорму ("Документ.ПредварительнаяЗапись.Форма.ФормаДоку  
мента", ПараметрыФормы, ЭтаФорма) ;

ИначеЕсли ТипЗнч (Расшифровка) =

Тип ("СправочникСсылка.Контрагенты") Тогда

ПараметрыФормы = Новый Структура ;

ПараметрыФормы.Вставить ("Ключ", Расшифровка) ;

ОткрытьФорму ("Справочник.Контрагенты.Форма.ФормаЭлемента",  
ПараметрыФормы, ЭтаФорма);

ИначеЕсли ТипЗнч(Расшифровка) =  
Тип ("СправочникСсылка.Склады") Тогда

ПараметрыФормы = Новый Структура;  
ПараметрыФормы.Вставить ("Ключ", Расшифровка);

ОткрытьФорму ("Справочник.Склады.Форма.ФормаЭлемента",  
ПараметрыФормы, ЭтаФорма);

КонецЕсли;

КонецПроцедуры

&НаКлиенте  
Процедура ТабличныйДокументПриАктивизацииОбласти (Элемент)

УстановитьВидимостьКонтекстногоМеню ();

КонецПроцедуры

#КонецОбласти

#Область ОбработчикиКомандФормы

&НаКлиенте  
Процедура УменьшитьПериод (Команда)

Если ПериодПереключатель = "День" Тогда  
Период.ДатаНачала = Период.ДатаНачала - 24  
\* 60 \* 60;

Период.ДатаОкончания =  
КонецДня (Период.ДатаНачала);

ИначеЕсли ПериодПереключатель = "Неделя" Тогда  
Период.ДатаНачала = Период.ДатаНачала - 7 \*  
24 \* 60 \* 60;

Период.ДатаОкончания =  
КонецНедели (Период.ДатаНачала);

ИначеЕсли ПериодПереключатель = "Месяц" Тогда  
Период.ДатаНачала =  
ДобавитьМесяц (Период.ДатаНачала, -1);

Период.ДатаОкончания =  
КонецМесяца (Период.ДатаНачала);

КонецЕсли;

Заголовок = СформироватьЗаголовокФормы (Период,  
ПериодПереключатель);

СохранитьНастройки ();



```

ОбновитьТабличныйДокумент ( ) ;

КонецПроцедуры

&НаКлиенте
Процедура УвеличитьПериод (Команда)

    Если ПериодПереключатель = "День" Тогда
        Период.ДатаНачала = Период.ДатаНачала + 24
    * 60 * 60;
        Период.ДатаОкончания =
КонецДня (Период.ДатаНачала) ;
    ИначеЕсли ПериодПереключатель = "Неделя" Тогда
        Период.ДатаНачала = Период.ДатаНачала + 7 *
24 * 60 * 60;
        Период.ДатаОкончания =
КонецНедели (Период.ДатаНачала) ;
    ИначеЕсли ПериодПереключатель = "Месяц" Тогда
        Период.ДатаНачала =
ДобавитьМесяц (Период.ДатаНачала, 1) ;
        Период.ДатаОкончания =
КонецМесяца (Период.ДатаНачала) ;
    КонецЕсли;

    Заголовок = СформироватьЗаголовокФормы (Период,
ПериодПереключатель) ;

    СохранитьНастройки ( ) ;

    ОбновитьТабличныйДокумент ( ) ;

КонецПроцедуры

&НаКлиенте
Процедура ДобавитьЗапись (Команда)

    ОткрытьФорму ("Документ.ПредварительнаяЗапись.Форма.Форм
аДокумента", Новый Структура ("ЗначенияЗаполнения",
ПолучитьПараметрыОткрытияПредварительнойЗаписи ( ) ), ЭтаФорма) ;

КонецПроцедуры

&НаКлиенте
Процедура ИзменитьЗапись (Команда)

    Расшифровка =
Элементы.ТабличныйДокумент.ТекущаяОбласть.Расшифровка;

    Если ТипЗнч (Расшифровка) =
Тип ("ДокументСсылка.ПредварительнаяЗапись") Тогда

        ПараметрыОткрытия = Новый Структура ( ) ;
        ПараметрыОткрытия.Вставить ("Ключ", Расшифровка) ;

```

ОткрытьФорму ("Документ.ПредварительнаяЗапись.Форма.ФормаДокумента", ПараметрыОткрытия, ЭтаФорма);

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура УдалитьЗапись (Команда)

Расшифровка =  
Элементы.ТабличныйДокумент.ТекущаяОбласть.Расшифровка;

Если ТипЗнч (Расшифровка) =  
Тип ("ДокументСсылка.ПредварительнаяЗапись") Тогда

Если УдалитьДокументНаСервере (Расшифровка) Тогда

ОбновитьТабличныйДокумент ();

КонецЕсли;

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура Обновить (Команда)

ОбновитьТабличныйДокумент ();

КонецПроцедуры

&НаКлиенте

Процедура УстановитьОтбор (Команда)

ПараметрыОткрытия = Новый Структура;  
Оповещение = Новый  
ОписаниеОповещения ("УстановитьОтборЗавершение", ЭтаФорма);  
Если СотрудникКабинетПереключатель = "Сотрудник" Тогда  
ПараметрыОткрытия.Вставить ("Сотрудники");  
ПараметрыОткрытия.Вставить ("Отбор", Новый  
Структура ("Сотрудник", Истина));

ОткрытьФорму ("Справочник.Контрагенты.Форма.ФормаВыбора",  
ПараметрыОткрытия, ЭтаФорма, , , , Оповещение);

Иначе

ОткрытьФорму ("Справочник.Склады.Форма.ФормаВыбора",  
ПараметрыОткрытия, ЭтаФорма, , , , Оповещение);

КонецЕсли;

```

КонецПроцедуры

&НаКлиенте
Процедура СнятьОтбор (Команда)

    СписокЗначенийОтбора.Очистить ();
    Элементы.ГруппаОтбор.Видимость =
ЗначениеЗаполнено (СписокЗначенийОтбора);

    ОбновитьТабличныйДокумент ();

КонецПроцедуры

&НаКлиенте
Процедура Настройки (Команда)

    ПараметрыОткрытия = Новый Структура;
    Оповещение = Новый
ОписаниеОповещения ("НастройкаЗавершение", ЭтаФорма);
    ОткрытьФорму ("Обработка.ЖурналЗаписиНаПрием.Форма.Настр
ойка", ПараметрыОткрытия, ЭтаФорма, , , , Оповещение);

КонецПроцедуры

#КонецОбласти

#Область СлужебныеПроцедурыИФункции

&НаСервере
Процедура ЗагрузитьНастройки ()

    ВремяНачалаРаботы =
? (ХранилищеОбщихНастроек.Загрузить ("ВремяРаботы",
"ВремяНачалаРаботы") = Неопределено, '000101011000',
ХранилищеОбщихНастроек.Загрузить ("ВремяРаботы",
"ВремяНачалаРаботы"));
    ВремяОкончанияРаботы =
? (ХранилищеОбщихНастроек.Загрузить ("ВремяРаботы",
"ВремяОкончанияРаботы") = Неопределено, '000101011900',
ХранилищеОбщихНастроек.Загрузить ("ВремяРаботы",
"ВремяОкончанияРаботы"));
    КратностьВремениЗаписи =
? (ХранилищеОбщихНастроек.Загрузить ("ВремяРаботы",
"КратностьВремениЗаписи") = Неопределено, 30,
ХранилищеОбщихНастроек.Загрузить ("ВремяРаботы",
"КратностьВремениЗаписи"));

    ЦветЗаписиКлиента =
? (ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ЦветЗаписиКлиента") = Неопределено, Новый Цвет (255,
232, 179),
ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ЦветЗаписиКлиента"));

```

```

        ЦветПервичнойЗаписи =
? (ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ЦветПервичнойЗаписи") = Неопределено, Новый Цвет (255,
165, 179),
    ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ЦветПервичнойЗаписи") ); ;
        ЦветПодтвержденнойЗаписи =
? (ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ЦветПодтвержденнойЗаписи") = Неопределено, Новый Цвет (0, 255,
0),
    ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ЦветПодтвержденнойЗаписи") ); ;
        ЦветНеявки =
? (ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ЦветНеявки") = Неопределено, Новый Цвет (255,
140, 105),
    ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ЦветНеявки") ); ;
        ЦветПрочейЗаписи =
? (ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ЦветПрочейЗаписи") = Неопределено, Новый Цвет (255,
165, 0),
    ХранилищеОбщихНастроек.Загрузить ("ЖурналЗаписиНаПрием",
"ЦветПрочейЗаписи") ); ;

```

КонецПроцедуры

&НаСервере

Процедура ОбновитьТабличныйДокумент ()

ТабличныйДокумент.Очистить ();

ТабличныйДокумент.ФиксацияСверху = 2;

ТабличныйДокумент.ФиксацияСлева = 1;

СтруктураАдресов =

ПредварительнаяЗапись.ПолучитьРасписаниеПриема (Период.ДатаНачала  
, Период.ДатаОкончания, КратностьВремениЗаписи,  
СотрудникКабиентПереключатель, СписокЗначенийОтбора);

ТаблицаИзмерений =

ПолучитьИзВременногоХранилища (СтруктураАдресов.АдресИзмерений);

ТаблицаЭлементов =

ПолучитьИзВременногоХранилища (СтруктураАдресов.АдресЭлементов);

Макет =

Обработки.ЖурналЗаписиНаПрием.ПолучитьМакет ("Макет");

ОбластьШапкаШкала =

Макет.ПолучитьОбласть ("Шапка | Шкала");

ОбластьШапкаИзмерение =

Макет.ПолучитьОбласть ("Шапка | Измерение");

```

        ОбластьШкала =
Макет.ПолучитьОбласть ("Кратность" + КратностьВремениЗаписи +
"|Шкала");
        ОбластьИзмерение =
Макет.ПолучитьОбласть ("Кратность" + КратностьВремениЗаписи +
"|Измерение");

        СплошнаяЛиния = Новый
Линия (ТипЛинииЯчейкиТабличногоДокумента.Сплошная);

        ТабличныйДокумент.Вывести (ОбластьШапкаШкала);

        // Вывод дат записей.
        ТекДата = Период.ДатаНачала;
        ТекКолонка = 2;

        Пока ТекДата <= НачалоДня (Период.ДатаОкончания) Цикл

            Отбор = Новый Структура;
            Отбор.Вставить ("Период", ТекДата);
            НайденныеЭлементы =
ТаблицаИзмерений.НайтиСтроки (Отбор);

            Если НайденныеЭлементы.Количество () > 0 Тогда
                КолонкаНачалаПериода = ТекКолонка;
                Для Каждого ТекИзмерение из НайденныеЭлементы
Цикл

                    ОбластьШапкаИзмерение.Параметры.Измерение = ТекДата;

                    ТабличныйДокумент.Присоединить (ОбластьШапкаИзмерение);
                        ТекКолонка = ТекКолонка + 1;
                    КонечЦикла;
                Если ТекКолонка - КолонкаНачалаПериода > 1
Тогда
                    ТабличныйДокумент.Область (1,
КолонкаНачалаПериода, 1, ТекКолонка - 1).Объединить ();
                    ТабличныйДокумент.Область (1,
КолонкаНачалаПериода, 1, ТекКолонка - 1).Обвести (СплошнаяЛиния,
СплошнаяЛиния, СплошнаяЛиния, СплошнаяЛиния);
                    КонечЕсли;
                Иначе
                    ОбластьШапкаИзмерение.Параметры.Измерение =
ТекДата;

                    ТабличныйДокумент.Присоединить (ОбластьШапкаИзмерение);
                        ТекКолонка = ТекКолонка + 1;
                    КонечЕсли;

                    ТекДата = ТекДата + 24 * 60 * 60;
                КонечЦикла;

        ТабличныйДокумент.Вывести (ОбластьШапкаШкала);

```

```

// Вывод измерений (сотрудников или кабинетов).
ТекДата = Период.ДатаНачала;

Пока ТекДата <= НачалоДня(Период.ДатаОкончания) Цикл

    Отбор = Новый Структура;
    Отбор.Вставить ("Период", ТекДата);
    НайденныеЭлементы =
ТаблицаИзмерений.НайтиСтроки (Отбор);

    Если НайденныеЭлементы.Количество() > 0 Тогда

        Для Каждого ТекИзмерение из НайденныеЭлементы
Цикл

            ЗаполнитьЗначенияСвойств (ОбластьШапкаИзмерение.Параметры,
ТекИзмерение);

            ТабличныйДокумент.Присоединить (ОбластьШапкаИзмерение);
                КонецЦикла;
            Иначе
                ОбластьШапкаИзмерение.Параметры.Измерение =
"";

            ТабличныйДокумент.Присоединить (ОбластьШапкаИзмерение);
                КонецЕсли;

            ТекДата = ТекДата + 24 * 60 * 60;
КонецЦикла;

// Вывод шкалы времени и вывод граф.
ТекВремя = ВремяНачалаРаботы;

Пока Час (ТекВремя) < Час (ВремяОкончанияРаботы) Цикл

    ОбластьШкала.Параметры.Час = Формат (ТекВремя,
"ДФ=ЧЧ:мм");
    ТабличныйДокумент.Вывести (ОбластьШкала);

    ТекДата = Период.ДатаНачала;

    Пока ТекДата <= НачалоДня(Период.ДатаОкончания)
Цикл

        Отбор = Новый Структура;
        Отбор.Вставить ("Период", ТекДата);
        НайденныеЭлементы =
ТаблицаИзмерений.НайтиСтроки (Отбор);

        Если НайденныеЭлементы.Количество() > 0 Тогда

```

```

        Для Каждого ТекИзмерение из
НайденныеЭлементы Цикл

        ТабличныйДокумент.Присоединить (ОбластьИзмерение) ;
            КонечЦикла;
        Иначе

        ТабличныйДокумент.Присоединить (ОбластьИзмерение) ;
            КонечЕсли;

            ТекДата = ТекДата + 24 * 60 * 60;

            КонечЦикла;

            ТекВремя = ТекВремя + 60 * 60;

            КонечЦикла;

        // Заполнение расшифровки и объединение ячеек.
        Если ТаблицаЭлементов.Количество() = 0 Тогда
            Возврат;
        КонечЕсли;

        ТекДата = Период.ДатаНачала;
        ТекКолонка = 2;

        Пока ТекДата <= НачалоДня (Период.ДатаОкончания) Цикл

            Отбор = Новый Структура;
            Отбор.Вставить ("Период", ТекДата);
            НайденныеЭлементы =
ТаблицаИзмерений.НайтиСтроки (Отбор) ;

            Если НайденныеЭлементы.Количество() > 0 Тогда

                Для Каждого ТекИзмерение из НайденныеЭлементы
Цикл

                    ТекСтрока = 3;
                    ОбластьПоиска =
ТабличныйДокумент.Область (ТекСтрока - 1, ТекКолонка) ;
                    ТекОбластьШапкиИзмерения =
ТабличныйДокумент.НайтиЗначение (ТекИзмерение.Измерение, , ОбластьПоиска) ;

                    Если ТекОбластьШапкиИзмерения =
Неопределено Тогда

                        Возврат;
                    КонечЕсли;

                    ТекЗапись = Неопределено;
                    ТекВремя = ВремяНачалаРаботы;

```

```

        Пока Час (ТекВремя) <
Час (ВремяОкончанияРаботы) Цикл

        Отбор = Новый Структура;

        Отбор.Вставить (СотрудникКабиентПереключатель,
ТекИзмерение.Измерение);

        Отбор.Вставить ("Период",
ТекДата);

        Отбор.Вставить ("ВремяНачала",
ТекВремя);

        НайденныеЭлементы =
ТаблицаЭлементов.НайтиСтроки (Отбор);

        Если НайденныеЭлементы.Количество ()
> 0 Тогда

            ТекЗапись =
НайденныеЭлементы[0];

            ОбластьЯчеекЗаписи =
ТабличныйДокумент.Область (ТекСтрока, ТекКолонка);

            // Проверка наложения ячеек.
            Если Не
ОбластьЯчеекЗаписи.Верх = ОбластьЯчеекЗаписи.Низ И ТекСтрока <>
2 Тогда

                ОбластьЯчеекЗаписи.Разъединить ();

                ТабличныйДокумент.Область (ОбластьЯчеекЗаписи.Верх,
ТекКолонка, ТекСтрока -1, ТекКолонка).Объединить ();

                ТабличныйДокумент.Область (ОбластьЯчеекЗаписи.Верх,
ТекКолонка, ТекСтрока -1, ТекКолонка).Обвести (СплошнаяЛиния,
СплошнаяЛиния, СплошнаяЛиния, СплошнаяЛиния);

                ОбластьЯчеекЗаписи =
ТабличныйДокумент.Область (ТекСтрока, ТекКолонка);

                КонецЕсли;

                ОбластьЯчеекЗаписи.Расшифровка
= ТекЗапись.Регистратор;

                ОбластьЯчеекЗаписи.Текст =
СформироватьТекстЗаписи (ТекЗапись);

                Если ТекЗапись.Неявка Тогда

                    ОбластьЯчеекЗаписи.ЦветФона = ЦветНеявки;

                    ИначеЕсли
ТекЗапись.Подтверждение Тогда

                        ОбластьЯчеекЗаписи.ЦветФона = ЦветПодтвержденнойЗаписи;

```



```

Тогда
    ОбластьЯчеекЗаписи.ЦветФона = ЦветПервичнойЗаписи;
    ИначеЕсли ТекЗапись.Первичная
    =
    ПредопределенноеЗначение ("Перечисление.ТипыПредварительнойЗаписи
    .ПрочаяЗапись") Тогда

    ОбластьЯчеекЗаписи.ЦветФона = ЦветПрочейЗаписи;
    Иначе

    ОбластьЯчеекЗаписи.ЦветФона = ЦветЗаписиКлиента;
    КонецЕсли;

    ВысотаСтрокЗаписи =
    (ТекЗапись.ВремяОкончания - ТекЗапись.ВремяНачала) /
    (КратностьВремениЗаписи * 60);

    Если ВысотаСтрокЗаписи > 1
    Тогда

    ТабличныйДокумент.Область (ТекСтрока, ТекКолонка, ТекСтрока +
    ВысотаСтрокЗаписи - 1, ТекКолонка).Объединить ();

    ТабличныйДокумент.Область (ТекСтрока, ТекКолонка, ТекСтрока +
    ВысотаСтрокЗаписи - 1, ТекКолонка).Обвести (СплошнаяЛиния,
    СплошнаяЛиния, СплошнаяЛиния, СплошнаяЛиния);
    Иначе

    ТабличныйДокумент.Область (ТекСтрока, ТекКолонка, ТекСтрока,
    ТекКолонка).Обвести (СплошнаяЛиния, СплошнаяЛиния, СплошнаяЛиния,
    СплошнаяЛиния);

    КонецЕсли;

    КонецЕсли;

    ТекСтрока = ТекСтрока + 1;
    ТекВремя = ТекВремя +
    КратностьВремениЗаписи * 60;

    КонецЦикла;
    ТекКолонка = ТекКолонка + 1;
    КонецЦикла;
    Иначе
    ТекКолонка = ТекКолонка + 1;
    КонецЕсли;
    ТекДата = ТекДата + 24 * 60 * 60;
    КонецЦикла;

    КонецПроцедуры // ОбновитьТабличныйДокумент ()

    &НаКлиенте

```

Процедура УстановитьВидимостьКонтекстногоМеню()

Если

ЗначениеЗаполнено (ТабличныйДокумент.ТекущаяОбласть.Расшифровка)  
Тогда

Элементы.ТабличныйДокументКонтекстноеМенюДобавитьЗапись.Видимость = Ложь;

Элементы.ТабличныйДокументКонтекстноеМенюИзменитьЗапись.Видимость = Истина;

Элементы.ТабличныйДокументКонтекстноеМенюУдалитьЗапись.Видимость = Истина;

Иначе

Элементы.ТабличныйДокументКонтекстноеМенюДобавитьЗапись.Видимость = Истина;

Элементы.ТабличныйДокументКонтекстноеМенюИзменитьЗапись.Видимость = Ложь;

Элементы.ТабличныйДокументКонтекстноеМенюУдалитьЗапись.Видимость = Ложь;

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура СохранитьНастройки()

ХранилищеОбщихНастроек.Сохранить ("ЖурналЗаписиНаПрием",  
"Период", Период.ДатаНачала);

ХранилищеОбщихНастроек.Сохранить ("ЖурналЗаписиНаПрием",  
"ПериодПереключатель", ПериодПереключатель);

ХранилищеОбщихНастроек.Сохранить ("ЖурналЗаписиНаПрием",  
"СотрудникКлиентПереключатель",  
СотрудникКлиентПереключатель);

КонецПроцедуры

&НаСервере

Функция СформироватьТекстЗаписи(ТекЗапись)

Текст = "";

Если ТекЗапись.ТипЗаписи =  
ПредопределенноеЗначение ("Перечисление.ТипыПредварительнойЗаписи  
.ЗаписьКлиента") Тогда

Текст = СтрЗаменить (

```

        НСтр("ru = '%1 %2
            |%3
            |%4'",
            "%1",
Формат(ТекЗапись.Регистратор.ВремяНачала, "ДФ=ЧЧ:мм"));
        Текст = СтрЗаменить(Текст, "%2",
? (ЗначениеЗаполнено(ТекЗапись.Телефон), Строка(ТекЗапись.Клиент)
+ Символы.ПС + Строка(ТекЗапись.Телефон), ТекЗапись.Клиент));
        Текст = СтрЗаменить(Текст, "%3",
? (СотрудникКабинетПереключатель = "Сотрудник",
ТекЗапись.Кабинет, ТекЗапись.Сотрудник));
        Текст = СтрЗаменить(Текст, "%4",
ТекЗапись.Комментарий);
        Иначе
            Текст = СтрЗаменить(
                НСтр("ru = '%1 %2
                    |%3
                    |%4'",
                    "%1",
Формат(ТекЗапись.Регистратор.ВремяНачала, "ДФ=ЧЧ:мм"));
                Текст = СтрЗаменить(Текст, "%2",
ТекЗапись.ВидПрочейЗаписи);
                Текст = СтрЗаменить(Текст, "%3",
? (СотрудникКабинетПереключатель = "Сотрудник",
ТекЗапись.Кабинет, ТекЗапись.Сотрудник));
                Текст = СтрЗаменить(Текст, "%4",
ТекЗапись.Комментарий);
            КонечЕсли;

```

Возврат Текст;

КонечФункции

&НаСервереБезКонтекста

Функция УдалитьДокументНаСервере (ДокументСсылка)

Возврат

```

ПредварительнаяЗапись.УдалитьПредварительнуюЗапись (ДокументСсылка);

```

КонечФункции

&НаКлиентеНаСервереБезКонтекста

Функция СформироватьЗаголовокФормы (Период,
ПериодПереключатель)

```

        Если ПериодПереключатель = "День" Тогда
            Результат = СтрЗаменить(НСтр("ru = 'Журнал записи
на %1 (%2)'", "%1", Формат(Период.ДатаНачала, "ДЛФ=DD"));
            Результат = СтрЗаменить(Результат, "%2",
Формат(Период.ДатаНачала, "ДФ=dddd"));

```

```

        ИначеЕсли ПериодПереключатель = "Неделя" Или
ПериодПереключатель = "Месяц" Тогда
            Результат = СтрЗаменить (НСтр("ru = 'Журнал записи с
%1 по %2'"), "%1", Формат(Период.ДатаНачала, "ДЛФ=DD"));
            Результат = СтрЗаменить(Результат, "%2",
Формат(Период.ДатаОкончания, "ДЛФ=DD"));

        КонецЕсли;

        Возврат Результат;

КонецФункции

&НаСервереБезКонтекста
Функция ПолучитьОсновнойСкладСотрудника(Сотрудник)

    ОсновнойСклад = Справочники.Склады.ПустаяСсылка();

    Если ЗначениеЗаполнено(Сотрудник) Тогда

        ОсновнойСклад = Сотрудник.ОсновнойСклад;

    КонецЕсли;

    Возврат ОсновнойСклад;

КонецФункции

&НаКлиенте
Функция ПолучитьПараметрыОткрытияПредварительнойЗаписи()

    ТекущаяОбласть =
Элементы.ТабличныйДокумент.ТекущаяОбласть;
    ТекДата = ТабличныйДокумент.Область(1,
ТекущаяОбласть.Лево).Расшифровка;
    ТекИзмерение = ТабличныйДокумент.Область(2,
ТекущаяОбласть.Лево).Расшифровка;

    ДанныеЗаполнения = Новый Структура;

    // ДанныеЗаполнения.Вставить("Дата",
Дата(Прав(ТекДатаСтрокой, 4), Сред(ТекДатаСтрокой, 4, 2),
Лев(ТекДатаСтрокой, 2)));
    ДанныеЗаполнения.Вставить("Дата", ТекДата);
    Если ТипЗнч(ТекИзмерение) =
Тип("СправочникСсылка.Контрагенты") Тогда

        ДанныеЗаполнения.Вставить("Сотрудник",
ТекИзмерение);
        ДанныеЗаполнения.Вставить("Кабинет",
ПолучитьОсновнойСкладСотрудника(ТекИзмерение));

```

ИначеЕсли ТипЗнч(ТекИзмерение) =  
Тип("СправочникСсылка.Склады") Тогда

ДанныеЗаполнения.Вставить("Кабинет",  
ТекИзмерение);

КонецЕсли;

ВремяНачала = ВремяНачалаРаботы +  
(ТекущаяОбласть.Верх - 3) \* КратностьВремениЗаписи \* 60;

ВремяОкончания = ВремяНачалаРаботы +  
(ТекущаяОбласть.Низ - 2) \* КратностьВремениЗаписи \* 60;

ДанныеЗаполнения.Вставить("ВремяНачала",  
ВремяНачала);

ДанныеЗаполнения.Вставить("ВремяОкончания",  
ВремяОкончания);

Возврат ДанныеЗаполнения;

КонецФункции

#КонецОбласти