

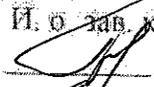
Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.03.02 – Информационные системы и технологии
Направленность (профиль) образовательной программы Безопасность
информационных систем

ДОПУСТИТЬ К ЗАЩИТЕ

И. о. зав. кафедрой

 А.В. Бушманов

« 08 » 07 2020 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка модуля автоматических систем для работы с
маркированными лекарственными препаратами в системе IС.

Исполнитель

студент группы 655-об


(подпись, дата)

А.Ф. Мышалов

Руководитель

доцент, канд. техн. наук


(подпись, дата)

А.В. Бушманов

Консультант

по безопасности и
экологичности

доцент, канд. техн. наук


(подпись, дата)

А.Б. Булгаков

Нормоконтроль

доцент, канд. техн. наук

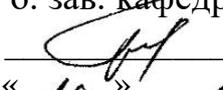

(подпись, дата)

О.В. Жилиндина

Благовещенск 2020

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

ДОПУСТИТЬ К ЗАЩИТЕ
И. о. зав. кафедрой
 А.В. Бушманов
« 10 » 02 2020 г.

ЗАДАНИЕ

К бакалаврской работе студента Мышалова Александра Федоровича.

1. Тема бакалаврской работы: Разработка модуля автоматических систем для работы с маркированными лекарственными препаратами в системе 1С

(утверждено приказом от 30.04.2020 № 810-уч)

2. Срок сдачи студентом законченной работы 26.06.2020 г.

3. Исходные данные к бакалаврской работе: отчет по преддипломной практике.

4. Содержание бакалаврской работы: анализ деятельности предприятия, разработка модуля автоматических систем, описание разработанного модуля, рассмотрение аспектов безопасности жизнедеятельности.

5. Перечень материалов приложения: 2 таблицы, 18 рисунков, 2 приложения.

6. Консультант по безопасности и экологичности Булгаков Андрей Борисович, доцент, канд. техн. наук.

7. Дата выдачи задания 20.02.2020 г.

Руководитель бакалаврской работы Бушманов Александр Вениаминович, доцент, канд. техн. наук.

Задание принял к исполнению (дата): 19.06.2020 Мышалов А.Ф.



РЕФЕРАТ

Бакалаврская работа содержит 59 с., 18 рисунков, 2 таблицы, 11 источников, 2 приложения.

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ, МОДУЛЬ
ИНФОРМАЦИОННОЙ СИСТЕМЫ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ,
АППАРАТНОЕ ОБЕСПЕЧЕНИЕ, DATAMARIX, МАРКИРОВКА

Для данной бакалаврской работы объектом исследования была выбрана разработка модуля информационной системы.

Целью работы является разработка модуля для работы с маркированными лекарственными препаратами в системе «1С: Предприятие».

Работа выполнялась последовательно в соответствии со следующими этапами: анализ предприятия, проектирование информационной подсистемы, разработка информационной подсистемы, рассмотрение вопросов информационной безопасности приложения, а также исследование аспектов безопасности жизнедеятельности.

Результатом выполнения бакалаврской работы является модуль автоматических систем для работы с маркированными лекарственными препаратами в системе «1С: предприятие».

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты и нормативные документы:

ГОСТ Р 50739-95 Средства вычислительной техники. Защита от несанкционированного доступа к информации. Общие технические требования.

ГОСТ Р ИСО 6385-2007 Эргономика. Применение эргономических принципов при проектировании производственных систем

СП 12.13130.2009 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности РД от 30.03.1992 СВТ защита от НСД. Показатели защищенности от НСД к информации

РД от 30.03.1992 АС защита от НСД. Классификация АС и требования по ЗИ

СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АО – аппаратное обеспечение;

БД – база данных;

БЖД – безопасность жизнедеятельности;

ИБ – информационная безопасность;

ИС – информационная система;

ИПС – информационная подсистема;

ОИБ – Отдел информационной безопасности;

ОПО – Отдел программного обеспечения;

ОС – операционная система;

ОТОиОП – Отдел технического обеспечения и оперативной полиграфии;

ПО – программное обеспечение;

РФ – Российская Федерация;

СНиП – санитарные нормы и правила;

СОДЕРЖАНИЕ

Введение	8
1 Анализ деятельности предприятия	10
1.1 Общая характеристика	10
1.2 Организационная структура	10
1.3 Обоснование необходимости создания модуля	12
1.4 DataMatrix код	14
1.4.1 Отличие DataMatrix от QR Code	16
1.4.2 Особенности штрихкода DataMatrix	16
1.4.3 Способы нанесения DataMatrix	18
2 Разработка модуля автоматических систем	20
2.1 Анализ требований	20
2.2 Характеристика функциональных подсистем	21
2.3 Проектирование модуля	21
3 Описание разработанного модуля	29
4 Угрозы информационной безопасности	33
4.1 Виды и описание угроз	33
4.2 Противодействие угрозам	34
5 Безопасность и экологичность	37
5.1 Безопасность	37
5.2 Экологичность	40
5.3 Комплексы физических упражнений для сохранения и укрепления индивидуального здоровья и обеспечения полноценной профессиональной деятельности	41
Библиографический список	44
Приложение	45

ВВЕДЕНИЕ

Проблема нелегальной продукции остается крайне острой для России: доля незаконного оборота в легкой промышленности достигает 35%, на парфюмерном рынке 20%, на рынке лекарственных средств до 10%.

В первом квартале 2018 года доля нелегального рынка сигарет в России увеличилась до 7,7% против 4,6% годом ранее, следует из исследования аналитического агентства Kantar TNS.

В декабре 2017 г. Президент РФ В.В. Путин одобрил решение правительства о создании до 2024 г. национальной системы цифровой маркировки товаров на базе Центра развития перспективных технологий (ЦРПТ).

ЦРПТ является совместным проектом ЮСМ («ЮЭСЭМ Технологии», 50%), госкорпорации Ростех (концерн «Автоматика», 25%) и «Элвис-Плюс групп» (25%).

Создаваемая центром система, получившая название Честный ЗНАК, позволяет эффективно бороться с контрафактом и контрабандой, защищает легальный бизнес, бренд добросовестного производителя и потребителей.

Благодаря внедрению системы маркировки легальный бизнес получит защиту от контрафакта, а его честное имя на рынке будет ограждено от репутационных потерь, связанных с некачественной продукцией, реализуемой под его брендом.

После внедрения маркировки стоимость товаров легальных производителей снизится.

На потребителя не лягут расходы на приобретение оборудования для маркировки, их возьмет на себя бизнес. При этом экономический эффект от маркировки покроет эти расходы. Благодаря системе производитель будет четко знать, сколько он продает товаров и как работают логистические цепочки. Это позволит оптимизировать бизнес-процессы и снизить издержки.

При самом консервативном сценарии после внедрения маркировки цены на товары могут снизиться на 10% за счет снижения доли контрафакта на рынке и других позитивных эффектов.

За год пилотного эксперимента по маркировке лекарств в России было выявлено нарушений на полмиллиарда рублей. Маркировка также позволила обнаружить факт вторичной продажи дорогостоящих препаратов, которые государство закупало для больниц по федеральной программе «7 нозологий».

Обязательная маркировка шуб, запущенная в 2016 году, также доказала свою эффективность: за год доля легального оборота меховых изделий выросла в 10 раз.

Если в 2015 году было продано 300 тысяч шуб, то в 2017 - уже 2 млн меховых изделий.

Опыт других стран тоже демонстрирует позитивные эффекты от внедрения маркировки. Например, в Бразилии был снижен объём поддельных табачных изделий, закрыты 7 нелегальных заводов, а собираемость налогов в табачном сегменте выросла с 2008 по 2010 гг на 40%.

1 АНАЛИЗ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ

1.1 Общая характеристика

ООО «Гармония» осуществляет деятельность по реализации ЛС на основании существующего законодательства, которое ставит обязательным условием получение лицензии на фармацевтическую деятельность — именно в этом документе указываются разрешенные виды деятельности. Помимо наличия лицензии нужно знать и следовать требованиям, установленным государственными стандартами, санитарными, противопожарными правилами и другими нормативными документами, правилами охраны труда и техники безопасности.

Виды деятельности, которые осуществляет ООО «Гармония»:

-Торговля розничная лекарственными средствами в специализированных магазинах (аптеках);

-Торговля оптовая фармацевтической продукцией;

-Торговля розничная изделиями, применяемыми в медицинских целях, ортопедическими изделиями в специализированных магазинах(аптеках);

-Торговля розничная косметическими и товарами личной гигиены в специализированных магазинах(аптеках);

1.2 Организационная структура ООО «Гармония»

Организационная структура — документ, схематически отражающий состав и иерархию подразделений предприятия. Организационная структура устанавливается исходя из целей деятельности и необходимых для достижения этих целей подразделений, выполняющих функции, составляющие бизнес-процессы организации.

Организационная структура определяет распределение ответственности и полномочий внутри организации. Структура ООО «Гармония» подразделяется на руководство и отделы (таблица 1).

Таблица 1 – Структура ООО «Гармония»

№№ п/п	Наименование	Численность (ед.)
I.	РУКОВОДСТВО	
1.	Генеральный директор	1
2.	Директор по развитию аптечной сети	1
3.	Главный бухгалтер	1
4.	Технический директор	1
II.	ОТДЕЛЫ	
2.	Отдел маркетинга	3
3.	Отдел автоматизации систем управления (Отдел АСУ)	3
4.	Гараж	
5.	Административно-хозяйственный отдел	4
6.	Отдел закупок ЛС и парафармацевтики	4
7.	Отдел ассортиментного анализа	5
8.	Бухгалтерия	4
	Итого:	24

На рис. 1 представлена схема организационной структуры ООО «Гармония»



Рисунок 1 – Схема организационной структуры ООО «Гармония»

Описание организационной структуры ООО «Гармония»:

-Руководство состоит из генерального директора, директора по развитию аптечной сети, главного бухгалтера и технического директора

-Генеральный директор управляет остальными членами руководства напрямую;

-В свою очередь помощники генерального директора управляют работой отделов, как показано на рисунке 1.

1.3 Обоснование необходимости создания модуля

Необходимость создания модуля для работы с маркированными лекарственными препаратами регулируется постановлением правительства №1556 от 14 декабря 2018 года, которое гласит:

В соответствии с частями 5 - 7 статьи 67 Федерального закона "Об обращении лекарственных средств" Правительство Российской Федерации постановляет:

1. Утвердить прилагаемое Положение о системе мониторинга движения лекарственных препаратов для медицинского применения.

2. Реализация настоящего постановления осуществляется соответствующими федеральными органами исполнительной власти в пределах установленной Правительством Российской Федерации предельной численности их работников и бюджетных ассигнований, предусмотренных им в федеральном бюджете на руководство и управление в сфере установленных функций.

3. Настоящее постановление вступает в силу со дня его официального опубликования, за исключением положений, указанных в пункте 4 настоящего постановления.

4. Положение о системе мониторинга движения лекарственных препаратов для медицинского применения, утвержденное настоящим постановлением, в отношении производителей лекарственных средств в части нанесения на первичную упаковку (в отношении лекарственных препаратов для медицинского применения, для которых не предусмотрена вторичная упаковка) и вторичную (потребительскую) упаковку лекарственных препаратов для медицинского применения средства идентификации, а также в отношении юридических лиц и индивидуальных предпринимателей, осуществляющих производство, хранение, ввоз в Российскую Федерацию, отпуск, реализацию, передачу, применение и уничтожение лекарственных препаратов для медицинского применения, в части внесения информации о лекарственных препаратах для медицинского применения в систему мониторинга движения лекарственных препаратов для медицинского применения вступает в силу со дня вступления в силу абзацев второго и пятого подпункта "б" пункта 7 статьи 1 Федерального закона от 28 декабря 2017 г. N

425-ФЗ "О внесении изменений в Федеральный закон "Об обращении лекарственных средств".

Так же будет использована система маркировки «Честный знак» с нанесением на упаковки препаратов DataMatrix кодов

А также постановлением правительства Российской Федерации №1118 от 30 августа 2019 года, постановлением правительства Российской Федерации №1557 от 14 декабря 2018 года, постановлением правительства Российской Федерации № 1558 от 14 декабря 2018 года, Федеральным законом №61 от 12 апреля 2010 года (ред. От 02. 08. 2019), постановлением правительства Российской Федерации №557 от 8 мая 2019 года и иными нормативно-правовыми документами, приведенными в разделе «Документы» на сайте «Честный ЗНАК».

1.4 DataMatrix код

DataMatrix — двумерный матричный штрихкод, имеющий два измерения (оси X и Y), для печати на упаковках товаров. Представляет из себя группу монохромных (черно-белых) квадратов, объединенных в группу, как правило, квадратной формы.

Предназначен для кодирования информации (текстовой или иной).

Описан в международном стандарте ISO/IEC 16022:2006. Он позволяет закодировать до 3Кб информации.

DataMatrix содержит избыточную структуру, которая позволяет восстановить до 30% полезной информации при частичном повреждении кода. Все коды используют коррекцию ошибок стандарта ECC200, который, в свою очередь, использует алгоритм Рида-Соломона(Reed-Solomon) для кодирования/декодирования данных.

DataMatrix это 2d штрих код (двухмерный), в который помещается гораздо больше данных, чем в линейный штрих код 1D. DataMatrix выбран для обязательной маркировки товара в системе Честный ЗНАК.

Кроме этого, код DataMatrix занимает меньше места, чем обычный штрихкод. Для чтения кода используют 2d сканер.

DataMatrix может использоваться как на упаковках небольшого размера (сигареты, лекарства, косметика, ювелирные изделия) так и на больших коробках.

Штрих код Data Matrix сейчас печатают на лекарствах, табачных изделиях, шубах.

По мере развития проекта Честный ЗНАК код DataMatrix будет почти на всех товарах, которые должны сканироваться на кассе.

DataMatrix ECC 200 — это последняя версия штрих кода, которая используется в системе GS1 и системе обязательной маркировки товаров Честный ЗНАК.

В штрих код DataMatrix можно закодировать:

- код товара GTIN GS1
- свойства товара — вес, срок годности, номер серии, номер партии, дата изготовления и розничную цену
- атрибуты товара — размер, цвет, и т.д.
- веб-ссылки, адреса электронной почты, телефонные номера
- код ТН ВЭД
- крипто хвост для проверки подлинности штрих кода

Честный ЗНАК использует штрих код DataMatrix для упрощения и удешевления проекта национальной системы маркировки и прослеживаемости товаров.

Вначале хотели использовать RFID метки и чипирование товаров, но вовремя отказались от этой идеи.

Честный ЗНАК кодирует в DataMatrix:

- Код товара GTIN
- Серийный номер, уникальный для каждой единицы товара
- Дополнительные поля, в зависимости от категории товара
- Код проверки, который генерирует ЦРПТ и таким образом криптографически подписывает каждый DataMatrix код

1.4.1 Отличие DataMatrix от QR Code

Визуально barcode GS1 DataMatrix, как и QR Code, выглядит как квадрат, состоящий из множества светлых и темных маленьких квадратиков. От QR-Code он отличается изображением шаблона поиска, который расположен слева в символе «L» (в QR коде это три угловых квадратных символа).

В стандартных символиках системы GS1 кодируются цифры, латинские буквы и некоторые избранные символы. Минимальный размер квадратного кода – 10x10 модулей, максимальный – 144x144, шаг увеличения – 2 модуля в каждом направлении. Для QR-Code шаг составляет 4 модуля.

1.4.2 Особенности штрихкода DataMatrix:

Сертифицирован по международному стандарту ISO/IEC16022.

Сканеры GS1 DataMatrix обладают высокой скоростью распознавания метки и извлечения информации.

Допустимы две графические формы расположения модулей – квадратная и прямоугольная, что позволяет легче разместить код на изделии.

Не требует свободного пространства вокруг метки, в то время как для QR-Code обязательно нужно пространство размером от 2 до 4 модулей.

Метка может быть нанесена не только на идеально ровную поверхность, требования к качеству основы для размещения кода довольно низкие.

Распознавание кода не зависит от фона, на котором размещены модули.

Большинство DataMatrix-ов квадратные, но в целом можно использовать и прямоугольные коды.

Кодируемые данные располагаются внутри прямоугольного шаблона, представляющего собой L-образный угол для ориентации считывающего устройства, и набор чередующихся чёрных и белых модулей по периметру символа.

Самым существенным достоинством DataMatrix является возможность на минимальной площади закодировать небольшие последовательности данных (до 10 символов).

Это преимущество объясняется тем, что в штрих-коде DataMatrix содержится очень мало служебной информации, описывающей размеры и структуру данных штрихового кода.

DataMatrix применяется в основном для маркировки небольших объектов, например, микросхем, позволяя закодировать 50 символов в изображении размером 2-3 мм².

Данная символика популярна в таких сферах применения как: медицинская промышленность; почтовые перевозки; электронная промышленность; автомобилестроение; пищевая промышленность; авиакосмическая и оборонная промышленность; энергетическое машиностроение.

DataMatrix адаптирован под считывание видеосистемами. Другими словами, считывателями данного кода могут служить видеокамеры со встроенными декодирующими программами, а также ручные и стационарные сканеры, оснащенные внутренними и программными декодерами.

Расстояние считывания зависит от таких факторов, как разрешение, размер кода и вид используемого оборудования.

В зависимости от них оно может варьироваться от 0 мм до 91 см.

1.4.3 Способы нанесения DataMatrix

-Методом прямой печати на изделии: струйной печатью, лазерной гравировкой, гравировкой иглами

-Печатью на бумажных и полипропиленовых материалах при помощи: Термо- и термотрансферных принтеров

-Матричных принтеров

-Струйных принтеров

-Лазерных принтеров

Технология DataMatrix подходит для интеграции в существующие системы, например, для его нанесения можно применять готовые приложения по печати этикеток (BarTender) и SDK, а для считывания – модели сканеров, с физическим или программным интерфейсом «разрыв клавиатуры».

Преимущества DataMatrix: За счет того, что код имеет небольшие размеры, можно использовать меньшие этикетки, а значит сократить стоимость затрат на расходные материалы.

Можно копировать большой объем информации (например, помимо основных данных, включить такие как ID производителя, серийный номер товара и др.).

Код считывается независимо от его ориентации относительно считывателя, что ускоряет процесс идентификации. Маркировка может наноситься непосредственно на само изделие.

Какую информацию содержит код Data Matrix?

Код маркировки содержит следующие обязательные поля: серийный номер, глобальный идентификационный номер торговой единицы (GTIN). В код также могут быть включены (необязательно) следующие поля: дата истечения срока годности, номер серии, в установленных законодательством форматах. Также код маркировки в обязательном порядке содержит идентификатор ключа и код проверки для обеспечения криптографической защиты информации по алгоритму шифрования с открытым ключом (ГОСТ 34.10-2012). Состав информации определяется нормативным правовым актом Правительства Российской Федерации.

2 РАЗРАБОТКА МОДУЛЯ АВТОМАТИЧЕСКИХ СИСТЕМ

2.1 Анализ требований

Любая информационная система предназначена для хранения, изменения, обработки и выдачи информации. Все эти операции над информацией требуется регулировать. В наше время компании стараются автоматизировать процессы взаимодействия с информацией и таким образом ускорить рабочий процесс, облегчить работу сотрудникам.

Объектом разработки данной работы является модуль автоматических систем для работы с маркированными лекарственными препаратами. Прежде всего он нацелен на сотрудников аптеки, которые будут взаимодействовать с поступающими данными, заносить их в базу, изменять, редактировать и удалять(продавать) их. Поэтому для работы с модулем информационной системы требуются люди, умеющие оперировать входящими данными.

Готовый модуль будет внедряться в имеющуюся информационную систему, с возможным использованием старой БД, но с внесёнными в неё изменениями, автоматизируя учет поступающих маркированных лекарственных препаратов. Программное обеспечение можно будет развернуть на компьютерах, установленных непосредственно в здании аптеки.

Обобщая вышесказанное, можно сформулировать требования:

1. Легкий в понимании и освоении интерфейс, поддерживающий русский язык.
2. Понятное представление данных для работы с ними.
3. Небольшой размер.
4. Низкая требовательность к производительности системы компьютера и ее компонентам.
5. Отсутствие критических ошибок, влияющих на работу.
6. Совместимость с последними версиями ОС Windows.

Однозначным достоинством разрабатываемого модуля является то, что разработка велась с нуля. Готовых решений на рынке, на момент разработки

не было. Использовались некоторые модули системы 1С, для проверки работоспособности разработанного модуля. На этапе тестирования ошибок выявлено не было.

2.2 Характеристика функциональных подсистем

В данном подразделе более подробно будут рассмотрены функциональные подсистемы, с целью более полного понимания решаемых ими задач и, тем самым, выбора наиболее точного способа их дальнейшей реализации.

Подсистема «Маркировка» должна представлять следующие функции:

- возможность внесения в БД новых маркированных препаратов;
- возможность внесения изменений в существующие списки препаратов;

Подсистема «Программное обеспечение» должна представлять следующие функции:

- возможность внесения изменений в существующее ПО

2.3 Проектирование модуля

Реализация сформированных требований может быть произведена по нескольким вариантам, поэтому одной из главных задач данного этапа является выбор наиболее подходящего из предложенных. Первым шагом важно определить структуру модуля. Затем предложить варианты реализации требований к продукту, а после проанализировать каждый и сделать выбор в пользу наиболее подходящего.

В качестве информационной системы и платформы разработки была выбрана система 1С: Предприятие, так как вся работа организации построена именно на этой системе, а также она соответствует требованиям организации и разрабатываемому модулю. Модуль должен работать в реальном времени, поэтому должна присутствовать быстрая обработка входящих данных и вывод необходимой информации пользователю.

В начале была написана функция преобразования строк из формата UTF8, в формат понятный программе (рисунок8):

```

Функция ПреобразоватьСтрокуИзUTF8 (Стр)
    Длина=СтрДлина (Стр);
    Имог="";
    Для N=1 По Длина Цикл
        Знак=Сред(Стр, N, 1);
        Код=КодСимв(Знак);
        Если Код<128 Тогда
            Имог=Имог+Знак;
        ИначеЕсли (Код>=128)И(Код<192) Тогда
            Иначе
                N=N+1;
                Знак2=Сред(Стр, N, 1);
                Код2=КодСимв(Знак2);
                Если Код=208 Тогда
                    Если Код2=129 Тогда Имог=Имог+"Ё";
                    Иначе Имог=Имог+Симв(КодСимв("А")+Код2-144);
                    КонечЕсли;
                ИначеЕсли Код=209 Тогда
                    Если Код2=145 Тогда Имог=Имог+"ё";
                    Иначе Имог=Имог+Симв(КодСимв("р")+Код2-128);
                    КонечЕсли;
                КонечЕсли;
            Иначе
                КонечЕсли;
        КонечЦикла;
    Возврат Имог;
КонечФункции
    
```

Рисунок 2- Преобразование строк

Затем были сформированы процедуры авторизации по фирме в базе данных системы ЧестныйЗнак (рисунок 3):

The screenshot shows a window titled "Конфигуратор - Группа компаний 'Здоровье' Аптека 2004". The menu bar includes "Файл", "Действия", "Текст", "Конфигурация", "Конструкторы", "Администрирование", "Сервис", "Окна", and "Помощь". The toolbar contains various icons for file operations and configuration. The main area displays a code editor with the following content:

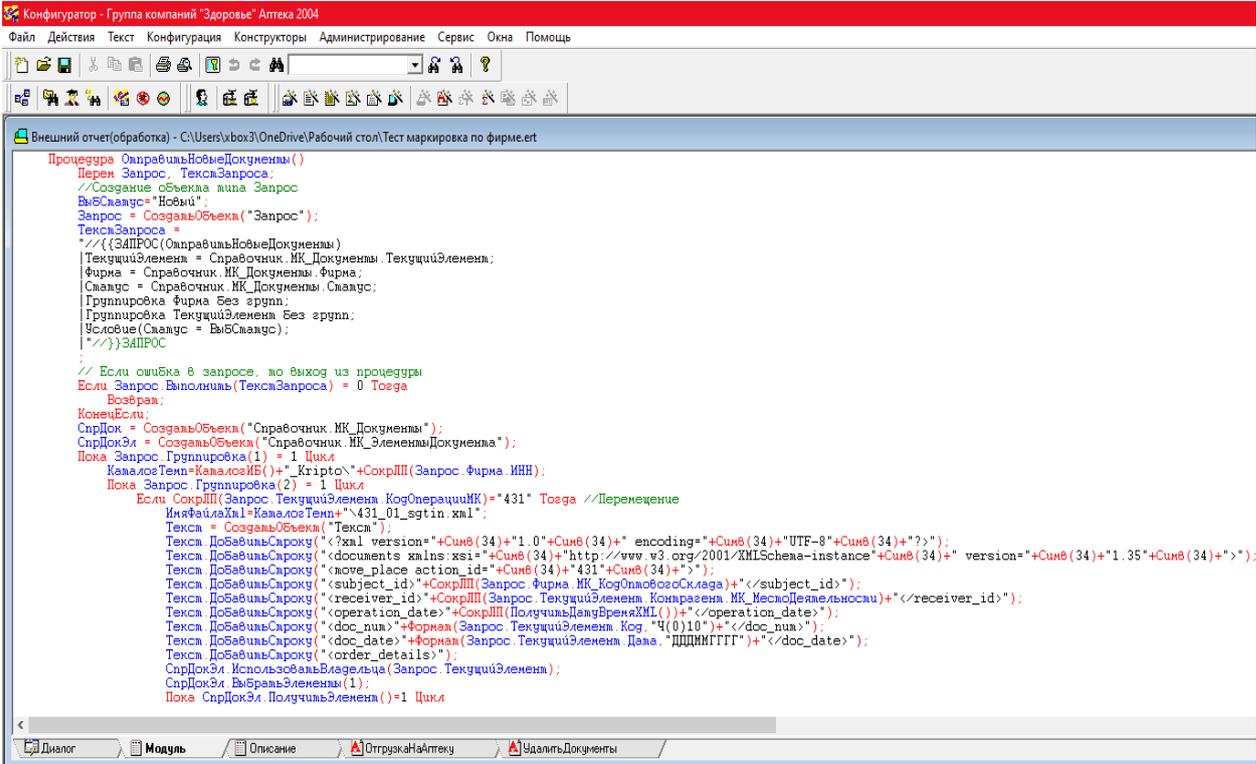
```

КонечФункции
Процедура АвторизацияПоФирме()
    Если ВыбФирма.Выбран()=0 Тогда
        Предупреждение("Не выбрана фирма!!!");
        Возврат;
    КонечЕсли;
    Если ПустаяСтрока(ВыбФирма.МК_КодУчетнойСистемы)=1 Тогда
        Предупреждение("Не задан код учетной системы!!!");
        Возврат;
    КонечЕсли;
    Если ПустаяСтрока(ВыбФирма.МК_СекретныйКлюч)=1 Тогда
        Предупреждение("Не задан секретный ключ!!!");
        Возврат;
    КонечЕсли;
    Если ПустаяСтрока(ВыбФирма.МК_Сертификат)=1 Тогда
        Предупреждение("Не задан сертификат!!!");
        Возврат;
    КонечЕсли;
    Если ПустаяСтрока(ВыбФирма.ИНН)=1 Тогда
        Предупреждение("Не задан ИНН!!!");
        Возврат;
    КонечЕсли;

    Если ФС.СуществуетФайл(КаталогИБ()+"_Kripto")=0 Тогда
        ФС.СоздатьКаталог(КаталогИБ()+"_Kripto");
    КонечЕсли;
    Если ФС.СуществуетФайл(КаталогИБ()+"_Kripto"+СокрЛП(ВыбФирма.ИНН))=0 Тогда
        ФС.СоздатьКаталог(КаталогИБ()+"_Kripto"+СокрЛП(ВыбФирма.ИНН));
    КонечЕсли;
    КаталогТемп=КаталогИБ()+"_Kripto"+СокрЛП(ВыбФирма.ИНН);

    СпрФирма = СоздатьОбъект("Справочник.Фирмы");
    СпрФирма.НайтиЭлемент(ВыбФирма);
    WinHttp = СоздатьОбъект("Msxml2.XMLHTTP.6.0");
    
```


После реализации данных методов, были реализованы процедуры по получению и удалению документов, а так же их отправки и контролирования отправки (рисунки 6-8)



```
Процедура ОтправитьНовыеДокументы()  
  Лерек Запрос, ТекстЗапроса;  
  //Создание объекта типа Запрос  
  ВыбСмалус = "Новый";  
  Запрос = СоздатьОбъект("Запрос");  
  ТекстЗапроса =  
  /*{{ЗАПРОС (ОтправитьНовыеДокументы)  
  |ТекущийЭлемент = Справочник МК_Документы. ТекущийЭлемент;  
  |Фирма = Справочник МК_Документы.Фирма;  
  |Смалус = Справочник МК_Документы.Смалус;  
  |Группировка Фирма Без групп;  
  |Группировка ТекущийЭлемент Без групп;  
  |Условие(Смалус = ВыбСмалус);  
  |/*}}ЗАПРОС  
  ;  
  // Если ошибка в запросе, но выход из процедуры  
  Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда  
    Возврат;  
  КонецЕсли;  
  СпрДок = СоздатьОбъект("Справочник МК_Документы");  
  СпрДокЭл = СоздатьОбъект("Справочник МК_ЭлементыДокумента");  
  Пока Запрос.Группировка(1) = 1 Цикл  
    КавалосТемп=КавалосИБ()+_Крипто^+СокрПП(Запрос.Фирма.ИНН);  
    Пока Запрос.Группировка(2) = 1 Цикл  
      Если СокрПП(Запрос.ТекущийЭлемент.КодОперацииМК)="431" Тогда //Переименование  
        ИмяФайлаXml=КавалосТемп+"431_01_sgtin.xml";  
        Текст = СоздатьОбъект("Текст");  
        Текст.ДобавитьСлowo("<?xml version="+Синв(34)+"1.0"+Синв(34)+" encoding="+Синв(34)+"UTF-8"+Синв(34)+"?>");  
        Текст.ДобавитьСлowo("<documents xmlns:xsi="+Синв(34)+"http://www.w3.org/2001/XMLSchema-instance"+Синв(34)+" version="+Синв(34)+"1.35"+Синв(34)+">");  
        Текст.ДобавитьСлowo("<doc_place action_id="+Синв(34)+"431"+Синв(34)+">");  
        Текст.ДобавитьСлowo("<subject_id"+СокрПП(Запрос.Фирма.МК.КодОпловогоСклада)+"</subject_id");  
        Текст.ДобавитьСлowo("<receiver_id"+СокрПП(Запрос.ТекущийЭлемент.Компания.МК.Месюлетельность)+"</receiver_id");  
        Текст.ДобавитьСлowo("<operation_date"+СокрПП(ПолучитьЦелуюВремяXML())+"</operation_date");  
        Текст.ДобавитьСлowo("<doc_num"+Формат(Запрос.ТекущийЭлемент.Код, "Ч(0)10")+"</doc_num");  
        Текст.ДобавитьСлowo("<doc_date"+Формат(Запрос.ТекущийЭлемент.Дата, "ДДММГГГГ")+"</doc_date");  
        Текст.ДобавитьСлowo("<order_details");  
        СпрДокЭл.ИспользоватьВладельца(Запрос.ТекущийЭлемент);  
        СпрДокЭл.ВыбратьЭлементы(1);  
        Пока СпрДокЭл.ПолучитьЭлемент()=1 Цикл
```

Рисунок 6-реализация процедуры отправки документов

```

Конфигуратор - Группа компаний "Здоровье" Аптека 2004
Файл Действия Текст Конфигурация Конструкторы Администрирование Сервис Окна Помощь

Внешний отчет(обработка) - C:\Users\hbox3\OneDrive\Рабочий стол\Тест маркировка по фирме.ert

Если СпрДокЭл.Элемент.Вуз()="МК_КодыУпаковок" Тогда
    Текст.ДобавитьСтроку("<sgtin>"+"СокрЛП(СпрДокЭл.Элемент.КодУпаковки)+"</sgtin>");
КонецЕсли;
КонецЦикла;
Текст.ДобавитьСтроку("</order_details>");
Текст.ДобавитьСтроку("</move_place>");
Текст.ДобавитьСтроку("</documents>");
Текст.Записать(ИмяФайла.xml);
КодДокумента=ОтправитьДокументыНаСайт(Запрос.Фирма,"431_01_sgtin.xml");
Если ПустаяСтрока(КодДокумента)=0 Тогда
    СпрДок.НайтиЭлемент(Запрос.ТекущийЭлемент);
    СпрДок.КодДокумента=КодДокумента;
    СпрДок.Смалус="Отправлен";
    СпрДок.Записать();
КонецЕсли;
КонецЕсли;
КонецЦикла;
КонецПроцедуры

Процедура КонтрольОтправленныхДокументов()
    ВыбСмалус="Отправлен";
    Запрос = СоздатьОбъект("Запрос");
    ТекстЗапроса =
    "://{ЗАПРОС(ОтправитьНовыеДокументы)
    |ТекущийЭлемент = Справочник.МК_Документы.ТекущийЭлемент;
    |Фирма = Справочник.МК_Документы.Фирма;
    |Смалус = Справочник.МК_Документы.Смалус;
    |Группировка Фирма Без групп;
    |Группировка ТекущийЭлемент Без групп;
    |Условие(Смалус = ВыбСмалус);
    |"}{ЗАПРОС
    ;
    // Если ошибка в запросе, то выход из процедуры
    Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда
        Возврат;
    КонецЕсли;
    СпрЭл = СоздатьОбъект("Справочник.МК_КодыУпаковок");
    СпрДок = СоздатьОбъект("Справочник.МК_Документы");

```

Рисунок 7-Реализация процедурправки и контроля отправки документов

```

Конфигуратор - Группа компаний "Здоровье" Аптека 2004
Файл Действия Текст Конфигурация Конструкторы Администрирование Сервис Окна Помощь

Внешний отчет(обработка) - C:\Users\hbox3\OneDrive\Рабочий стол\Тест маркировка по фирме.ert

Процедура УдалитьДокументы()
    Перецм Запрос, ТекстЗапроса, Таб;
    //Создание объекта типа Запрос
    Запрос = СоздатьОбъект("Запрос");
    ТекстЗапроса =
    "://{ЗАПРОС(УдалитьДокументы)
    |МК_Документы = Справочник.МК_Документы.ТекущийЭлемент;
    |Группировка МК_Документы;
    |"}{ЗАПРОС
    ;
    // Если ошибка в запросе, то выход из процедуры
    Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда
        Возврат;
    КонецЕсли;

    // Подготовка к заполнению выходных форм данными запроса
    Спр = СоздатьОбъект("Справочник.МК_Документы");
    Таб = СоздатьОбъект("Таблица");
    Таб.ИсходнаяТаблица("УдалитьДокументы");
    // Заполнение полей "Заголовок"
    Таб.ВывестиСекцию("Заголовок");
    Состояние("Заполнение выходной таблицы...");
    Таб.Опции(0, 0, Таб.ВысотаТаблицы(), 0);
    Пока Запрос.Группировка(1) = 1 Цикл
        // Заполнение полей МК_Документы
        Таб.ВывестиСекцию("МК_Документы");
        Спр.НайтиЭлемент(Запрос.МК_Документы);
        Спр.Удалить(1);
    КонецЦикла;
    // Вывод заполненной формы
    Таб.ТолькоПросмотр(1);
    Таб.Показать("УдалитьДокументы", "");
КонецПроцедуры

```

Рисунок 8-реализация процедуры удаления документов

Так же были реализованы процедуры выведения информации по упаковке и расформирование кода агрегации. Информация по упаковке считывается из DataMatrix кода, декодируется и записывается в файл, с последующей отправкой в систему ЧестныйЗНАК и выведением на рабочее место кассира всей необходимой информации по продаваемому препарату. А код агрегации сохраняется в виде текстового файла с именем вида «Справочник.МК_КодыУпаковок» (рисунок 9, 10)

```

Процедура ИнформацияПоУпаковке()
Если ВыESGTIN.Выбран()=0 Тогда
    Предупреждение("Не выбран элемент с кодом упаковки!!!");
    Возврат;
КонецЕсли;
ВыБФирма=ВыESGTIN.ФирмаОптовая;
Если ВыБФирма.Выбран()=0 Тогда
    Предупреждение("У кода агрегации не определена фирма!!!");
    Возврат;
КонецЕсли;
Канал.ЛогТемп=Канал.ЛогИБ()+ "_Крипто\"+СокрIII(ВыБФирма.ИНН);
ИмяФайлаXml=Канал.ЛогТемп+"220_01_sgtin.xml";
Текст = СоздатьОбъект("Текст");
Текст.ДобавитьСлужку("<?xml version="+Симв(34)+"1.0"+Симв(34)+" encoding="+Симв(34)+"UTF-8"+Симв(34)+"?>");
Текст.ДобавитьСлужку("<documents xmlns:xsi="+Симв(34)+"http://www.w3.org/2001/XMLSchema-instance"+Симв(34)+" version="+Симв(34)+"1.35"+Симв(34)+">");
Текст.ДобавитьСлужку("<query_hierarchy_info action_id="+Симв(34)+"220"+Симв(34)+">");
Текст.ДобавитьСлужку("<subject_id'+СокрIII(ВыБФирма.МК_КодОптовоеСклада)+'</subject_id'");
Текст.ДобавитьСлужку("<sgtin'+СокрIII(ВыESGTIN.КодУпаковки)+'</sgtin'");
Текст.ДобавитьСлужку("</query_hierarchy_info action_id'");
Текст.ДобавитьСлужку("</documents'");
Текст.Записать(ИмяФайлаXml);

КодДокумента=ОтправитьДокументНаСайт(ВыБФирма,"220_01_sgtin.xml");
СрКВ.Византия=СкачатьКВ.ВизантияПоИдентификатору(ВыБФирма.КодДокумента);
КонецПроцедуры

```

Рисунок 9-Информация по упаковке

```

Процедура РасформированиеКодаАгрегации()
Если ВыSSCC.Выбран()=0 Тогда
    Предупреждение("Не выбран элемент с кодом агрегации!!!");
    Возврат;
КонецЕсли;
ВыБФирма=ВыSSCC.ФирмаОптовая;
Если ВыБФирма.Выбран()=0 Тогда
    Предупреждение("У кода агрегации не определена фирма!!!");
    Возврат;
КонецЕсли;
Счетчик = 0;
СрКВ.Виз = СоздатьОбъект("Справочник.МК_КодыУпаковок");
СрКВ.Виз.ВыбратьЭлементыПоРеквизиту("КодАгрегирования", ВыSSCC.КодАгрегирования, 0, 0);
Пока СрКВ.Виз.ПолучитьЭлемент()=1 Цикл
    Счетчик=Счетчик+1;
КонецЦикла;
Если ВыSSCC.Количество()<Счетчик Тогда
    Предупреждение("У кода агрегации не соответствует кол-во упаковок!!!");
    Возврат;
КонецЕсли;
СрSSCC = СоздатьОбъект("Справочник.МК_КодыАгрегирования");
СрSSCC.НайтиЭлемент(ВыSSCC);
Канал.ЛогТемп=Канал.ЛогИБ()+ "_Крипто\"+СокрIII(ВыБФирма.ИНН);
ИмяФайлаXml=Канал.ЛогТемп+"912_01_sgtin.xml";
Текст = СоздатьОбъект("Текст");
Текст.ДобавитьСлужку("<?xml version="+Симв(34)+"1.0"+Симв(34)+" encoding="+Симв(34)+"UTF-8"+Симв(34)+"?>");
Текст.ДобавитьСлужку("<documents xmlns:xsi="+Симв(34)+"http://www.w3.org/2001/XMLSchema-instance"+Симв(34)+" version="+Симв(34)+"1.35"+Симв(34)+">");
Текст.ДобавитьСлужку("<unit_unpack action_id="+Симв(34)+"912"+Симв(34)+">");
Текст.ДобавитьСлужку("<subject_id'+СокрIII(ВыБФирма.МК_КодОптовоеСклада)+'</subject_id'");
Текст.ДобавитьСлужку("<operation_date'+СокрIII(ПолучитьПлатьеВремяXML())+'</operation_date'");
Текст.ДобавитьСлужку("<sscc'+СокрIII(ВыSSCC.КодАгрегирования)+'</sscc'");
Текст.ДобавитьСлужку("</unit_unpack'");
Текст.ДобавитьСлужку("</documents'");
Текст.Записать(ИмяФайлаXml);

КодДокумента=ОтправитьДокументНаСайт(ВыБФирма,"912_01_sgtin.xml");
СрКВ.Византия=СкачатьКВ.ВизантияПоИдентификатору(ВыБФирма.КодДокумента);
КонецПроцедуры

```

Рисунок 10-Расформирования кода агрегации

В дальнейшем все функции и процедуры были внесены в диалоговую форму (рисунок 11):

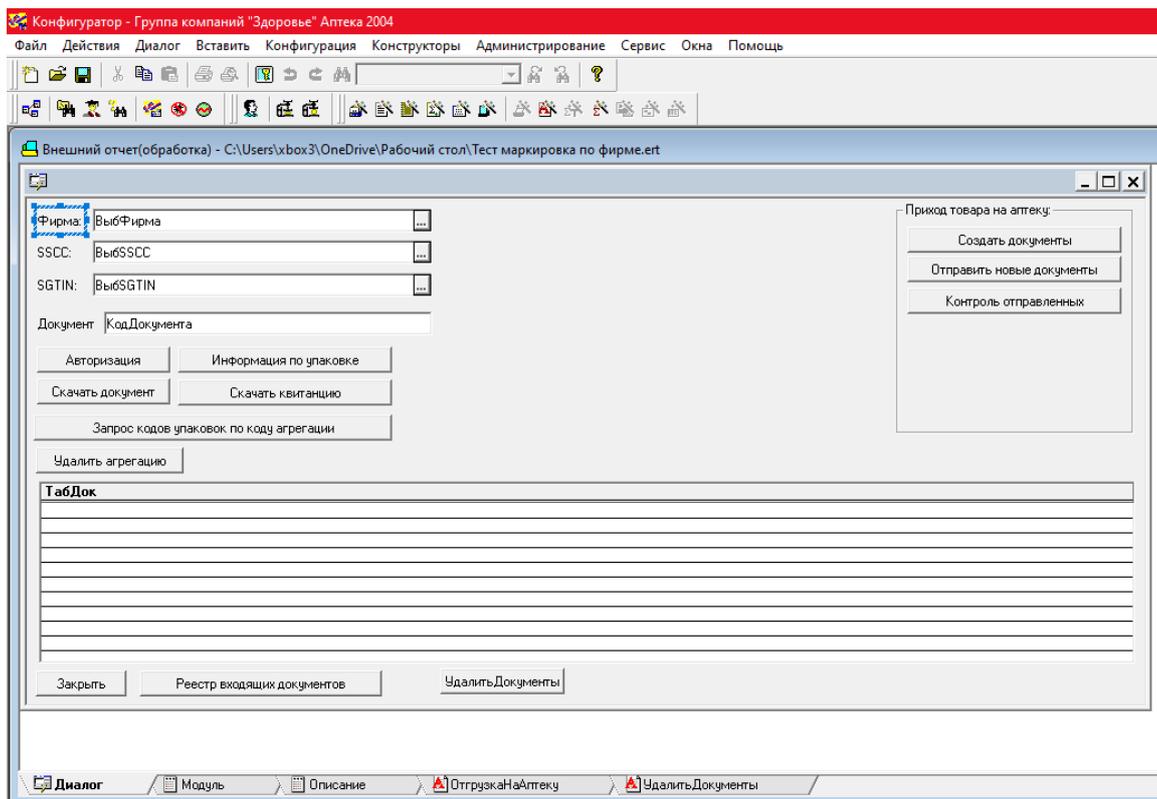


Рисунок 11-форма для работы с модулем

3 ОПИСАНИЕ РАЗРАБАТЫВАЕМОГО МОДУЛЯ

Модуль предназначен для работы с маркированными препаратами и базой данных системы ЧестныйЗНАК. Маркировка наносится производителями лекарств и препаратов самостоятельно. Логические компании и дистрибьюторы должны передавать данные о полученных от поставщика и проданных лекарственных препаратах в систему ЧестныйЗНАК. Аптеки так же в свою очередь передают все данные полученные от поставщика в систему ЧестныйЗНАК, а также о проданных упаковках через ОФД с помощью онлайн-касс. Медицинские учреждения должны отправлять данные о выбытии лекарств при помощи регистраторов выбытия.

Модуль внедряется в уже рабочую систему организации, которая работает на базе 1С: Предприятие.

Реестр маркированных упаковок загружается из системы ЧестныйЗНАК (рисунок 12)

Идентификатор	Код упаковки	Производитель	Год эксп. до	Поставщик	Инициалы фирмы	Статус
31	36033999007311_ДЕ117193L21 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
32	36033999007311_ДЕ117194L3K 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
33	36033999007311_ДЕ117194L3D 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
34	36033999007311_ДЕ117194L3H 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
35	36033999007311_ДЕ117194L4T 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
36	36033999007311_ДЕ117194L4D 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
37	36033999007311_ДЕ117194L49 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
38	36033999007311_ДЕ117194L4E 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
39	36033999007311_ДЕ117194L4F 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
40	36033999007311_ДЕ117194L4G 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
41	36033999007311_ДЕ117194L4H 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
42	36033999007311_ДЕ117194L4I 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
43	36033999007311_ДЕ117194L4J 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
44	36033999007311_ДЕ117194L4K 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
45	36033999007311_ДЕ117194L4L 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
46	36033999007311_ДЕ117194L4M 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
47	36033999007311_ДЕ117194L4N 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
48	36033999007311_ДЕ117194L4O 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
49	36033999007311_ДЕ117194L4P 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
50	36033999007311_ДЕ117194L4Q 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
51	36033999007311_ДЕ117194L4R 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
52	36033999007311_ДЕ117194L4S 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
53	36033999007311_ДЕ117194L4T 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
54	36033999007311_ДЕ117194L4U 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
55	36033999007311_ДЕ117194L4V 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
56	36033999007311_ДЕ117194L4W 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
57	36033999007311_ДЕ117194L4X 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
58	36033999007311_ДЕ117194L4Y 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова
59	36033999007311_ДЕ117194L4Z 36033999000872909	Общество ПРК 41219	01.01.22	МФ Обществу ПРК АО (серия 01.12.16)	Александр ООУ	Медерсова

Рисунок 12-Реестр маркированных упаковок

Затем происходит агрегация полученных данных и загрузка их в систему

модуля

1С

(рисунок

13)

Номенклатура	Код агрегирования	Производитель	Серия	Год от до	Поставщик	Количество	Отправ. фирма	Запрос по времени	Запрос по упаковке	Статус
Вино 500мл бутылка ИЗО	3531143001738847	Белтаг Ридинг с/поборозно П/д	890089	01.02.22	Досон 24 Либароско (серия 30.09.18)	30	Талисман ООО			Сделан
Вино: 1000мл 154 бутылка ИЗО	146350779900073482	Оболенское ФП ЗАО	2130719	30.07.23	Нача на брн 8507-0007/200 (24.10.19)	72	Талисман ООО	35345762484e-41d8488483548-3a30		Сделан
Суплевант изотон.0,5% (пленка 2л)	346139880003872880	Оболенское ПФК	41219	01.01.22	Нача на брн 8507-0007/200 (24.10.19)	300	Талисман ООО	c00fa283-8035-4a6e-9167-d8076a072d4		Маркировка

Рисунок 13-Агрегированные упаковки

Для обмена информацией с системой ЧестныйЗНАК, используются документы, составляемые сотрудниками организации. Эти документы записываются в реестр для более быстрого поиска при необходимости (рисунок 14)

Документ МК	Код операции	Код клиента	Фирма	Код документа в РЕ	Статус
Дата		Место деятельности	Контрагент	Комментарий	
15.10.19	607	51579420544-4eb5-5050-11e404416eb2b	Талисман ООО	6455049F-916a-4605-5026-15e54d4eb2b	Принят
24.10.19	607	00000000000000	Илгрен Р Либароско (серия 30.09.18)	551254a2-6900-41494-3e412104c9e0f	Принят
22.05.20	431	0014805c484-4205-ac85-c507617a1500	Талисман ООО	6012a64c-3076-4344-601b-155eeeb4eb2	Сделан
23.05.20	607	00000000010743	Талисман ООО	52ae0270-8934-4095-963c-15e0140048f3	Новый
11.05.20	431	164114495-a26e-43e0-5042-0cc85047eb1a	Илгрен Р Либароско (серия 30.09.18)	8096f144-4844-4127-932c-61912121600	Сделан
		000000000140476	Илгрен Р Либароско (серия 30.09.18)	Вспомогательное завершение операции	
			Илгрен Р Либароско (серия 30.09.18)	Вспомогательное завершение операции	

Рисунок 14-Реестр документов

Накладная или разборочный лист с перечнем препаратов приходит на склад от поставщика. В этом документе содержится вся информация о препаратах, начиная с наименования, заканчивая регистрационным

Приемка товара Аптека 2020 Маркировка!!!

Ввод уникальных кодов по маркировке

Товар: Венарус 1000мг тбл п/п об №60

Кол-во: 00002 из 00005

04605077012480010001K996A04
04605077012480010001K98EJ8G

[F2] - Продолжить разбор Штрих-код

Рисунок 17-Ввод упаковок по принятому товару

4 УГРОЗЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Что такое безопасность? Часто считается, что для обеспечения безопасности достаточно задать пароль учетной записи или зашифровать данные. В некоторых случаях это так и есть. Но правильной будет уточнить, что данные меры усиливают защиту информации на некоторых «участках» функционирования системы. Обеспечение защиты данных понятие комплексное. Часть мер по организации контроля доступа к информации осуществляется техническими средствами, в том числе платформой 1С.

4.1 Угрозы ИБ

В таблице представлены вероятные угрозы безопасности и их характеристики.

Таблица 2 – Аудит безопасности системы

Вероятная угроза безопасности	Последствия	Вероятный злоумышленник
Доступ пользователей к административным действиям конфигулятора	Копирование всей информации	Программист, продвинутый пользователь
Использование старых логинов	Последствия зависят от прав учетной за-писи	Уволенный сотрудник
Наличие уязвимостей операционной системы, СУБД	Повышение прав с помощью взломов, опубликованных в интернете, возможно полное копирование информации	Администратор, Продвинутый пользователь
Вирусы, трояны, логгеры	Порча информации, получение данных, раскрытие паролей пользователей	Администратор, Продвинутый пользователь
Пароли на мониторах, слабые пароли	Несанкционированный доступ к учетным записям пользователей с последующим доступом к данным	Продвинутый пользователь, уволенный сотрудник

Первой угрозой безопасности является доступ пользователя к административным действиям конфигуратора.

Каждая учетная запись пользователя принадлежит к одной или нескольким ролям. Роль – это набор прав. Некоторые права предоставляют возможность копирования информации или выполнения административных действий. Такие права должны быть только у пользователей, которым они необходимы по служебным обязанностям, например, у администратора информационной системы.

Так же актуальна такая угроза как использование старых логинов. Среди учетных записей, используемых системой могут оставаться те, которые принадлежали уволенным сотрудникам. Необходимо организовать регулярную процедуру проверки учетных записей.

Угроза уязвимости операционной системы выявляется отслеживанием информации о уязвимостях операционной системы. При появлении официальных сообщений от производителя, пользоваться предлагаемыми обновлениями. Если есть возможность, пользоваться автоматической загрузкой и установкой обновлений.

Обычно такие угрозы как вирусы, трояны и логгеры не нацелены на взлом приложений 1С. Однако, нарушение работы «1С: Предприятие» может стать побочным эффектом деятельности вируса, поэтому пренебрегать этой опасностью не следует.

Следующей угрозой является слабые пароли и нахождение паролей на мониторе. Наличие действующих учетных записей уволенных сотрудников или не использующихся действующими сотрудниками больше месяца. Даже если сотрудник находится в отпуске, на время его отсутствия лучшим средством снизить риск является временно отключение пользователя или смена пароля. Серьезность данной угрозы прямо зависит от прав записи данного сотрудника.

4.2 Противодействие угрозам

Начнем с угрозы доступа пользователя к административным действиям конфигулятора.

Для защиты от данной угрозы нужно ограничить права, назначать административные права только тем пользователям «1С: Предприятие», которым это действительно необходимо. Необходимо выделить права, которыми должен обладать только администратор и лишить этих прав всех остальных пользователей. Обычно, к таким правам относятся:

- административные функции;
- обновление конфигурации;
- внешнее соединение;
- интерактивное открытие внешних обработок;
- интерактивное открытие внешних отчетов.

Для решения проблемы отсутствия разграничений доступа в режиме «1С: Предприятие». Необходимо привести права доступа в соответствие с должностными обязанностями каждого сотрудника. Эта задача, в общем случае, является достаточно трудоемкой. Имеет смысл начинать проверку прав доступа и приведение их в соответствие с наиболее критичных областей данных.

Для избегания угрозы использования «старых» паролей необходимо проводить регулярное администрирование учетных записей. В данном случае обеспечение безопасности не является операцией, которая раз и навсегда решит проблему. Для поддержания необходимого уровня безопасности на данном участке необходимо организовать и проводить регулярную проверку всех учетных записей системы.

Во избежание уязвимостей ОС необходимо своевременно выполнять ее обновление.

Клиентская «1С: Предприятие» имеет наибольший риск заражения вирусами. Поэтому используя антивирусную программу нужно выбрать разумный компромисс между степенью защиты и быстродействием.

Можно использовать следующие варианты настройки антивируса клиентской части:

1) добавить в исключения проверяемых файлов:

- файлы с расширениями *.1CD, *.1cl, *.log, *.elf, *.pfl, *.usr, *.v8i, *.lst, *.st, *.snp, *.epf, *.dat;

- исполняемые приложения: 1cv8.exe, ragent.exe, rmngr.exe, rphost.exe, 1CV8 Servers.msc из папки C:\Program Files\1cv77\bin\ (C:\Program Files (x86)\1cv77 - для 32битной версии в 64 битной среде) или C:\Program Files\1cv77\bin\ (C:\Program Files (x86)\1cv77 - для 32 битной версии в 64 битной среде);

2) исключить из проверки подкаталоги с установленными «1С: Предприятие», обычно:

- C:\Program Files\1cv77;
- C:\Program Files (x86)\1cv77;
- C:\Program Files\1cv77;
- C:\Program Files (x86)\1cv77;

Пользователи могут забывать пароли и записывать их на видном месте. Чтобы избежать этого, на предприятии можно использовать устройства автоматической аутентификации, например, смарт-карты.

5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

В этом разделе будут рассмотрены вопросы безопасности жизнедеятельности сотрудников технического отдела ООО «Гармония». Будут определены правила работы за ПК, способы безопасной утилизации носителей информации и элементов ПК, а также рассмотрена противопожарная безопасность на основе СанПиН 2.2.2/2.4.1340-03.

5.1 Безопасность

В СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к ПЭВМ и организации работы» представлены ряд правил и норм для рабочих мест с ПЭВМ.

В кабинете АИС выполняются следующие требования:

1) требования к ПЭВМ:

- все ПЭВМ соответствуют требованиям СанПиН 2.2.2/2.4.1340-03;
- конструкция ПЭВМ обеспечивает возможность поворота корпуса в горизонтальной и вертикальной плоскости с фиксацией в заданном положении;
- корпуса ПЭВМ, клавиатура и другие блоки и устройства ПЭВМ окрашены в серый и черный цвета, имеют матовую поверхность;
- конструкцией ВДТ предусмотрено регулирование яркости и контрастности.

2) требования к помещениям для работы с ПЭВМ:

- окна в помещении, где эксплуатируется вычислительная техника, ориентированы на запад. Оконные проемы оборудованы жалюзи;
- площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе плоских дискретных экранов составляет более 4,5 м²;

Расположение рабочих мест сотрудников технического блока представлено на рисунке 24. После проведения расчетов, было выявлено, что все рабочие места располагаются согласно нормам СанПиН.

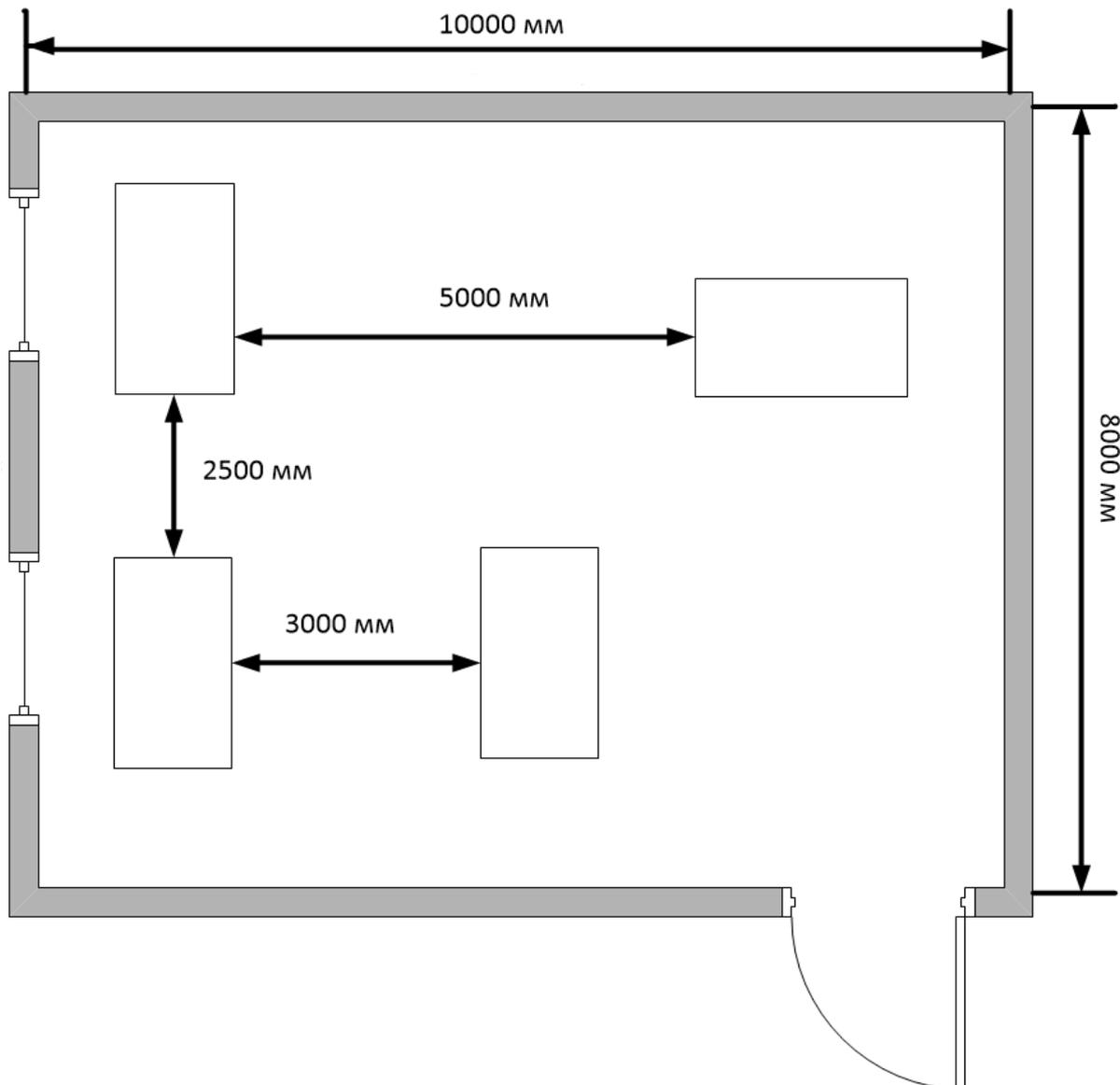


Рисунок 18 – Расположение рабочих мест отдела АИС ООО
«Гармония»

3) требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ:

- шумящее оборудование, а именно серверное оборудование, уровни шума которого превышают нормативные, размещается в отдельном помещении здания.

4) требования к освещению на рабочих местах, оборудованных ПЭВМ:

- искусственное освещение в помещении для эксплуатации ПЭВМ

осуществляется системой общего равномерного освещения.

5) общие требования к организации рабочих мест пользователей ПЭВМ.

- экран видеомонитора находится от глаз пользователя на расстоянии 60- 70 см;

- конструкция рабочего стола обеспечивает оптимальное размещение на рабочей поверхности используемого оборудования, а также обеспечивает поддержание рациональной рабочей позы при работе на ПЭВМ;

- рабочий стул - подъемно-поворотный, регулируется по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья. Поверхность сиденья, спинки полумягкая, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

б) требования к организации и оборудованию рабочих мест с ПЭВМ для взрослых пользователей.

- высота рабочей поверхности не регулируется, но соответствует указанно норме в 72 см. Рабочий стол имеет пространство для ног высотой 65 см, шириной – 60 см;

- клавиатура располагается на поверхности стола на расстоянии 30 см от края, обращенного к пользователю.

ООО “Гармония” 8-ми часовая рабочая смена поэтому перерывы следует устанавливать:

- 1 категория работ - через 2 часа от начала смены и через 2 часа после обеденного перерыва продолжительностью 15 минут каждый;

- 2 категория работ - через 2 часа от начала рабочей смены и через 1,5- 2,0 часа после обеденного перерыва продолжительностью 15 минут каждый или продолжительностью 10 минут через каждый час работы;

- 3 категория работ - через 1,5- 2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

5.2 Экологичность

Для обеспечения экологичности на предприятии ООО «Гармония» существует Федеральный закон №89 «Об отходах производства и потребления» от 24.06.1998 (ред. От 28.12.2016). Данным законом регулируются способы утилизации отходов.

Самостоятельная утилизация бумажных отходов, то есть сжигание, закапывание не допускается. Поэтому данные отходы необходимо складировать и после передавать на утилизацию предприятию, занимающемуся переработкой.

Светильники, содержащие ртуть, необходимо не следует утилизировать как обычные отходы, необходимо хранить, выработавшие свой ресурс лампы в специальном помещении.

Пожарная безопасность при работе с ЭВМ

Помещения, где эксплуатируются ЭВМ, относятся к категории В - пожароопасные помещения, согласно НПБ 105-03. В следствии этого обеспечение именно пожарной безопасности является наиважнейшей задачей при рассмотрении безопасности жизнедеятельности на предприятии.

Согласно ГОСТ 12.4.026 – 2015 «Цвета сигнальные, знаки безопасности и разметка сигнальная» от 01 марта 2017 года. План эвакуации – это знак пожарной безопасности, указывающий путь эвакуации от места его расположения на плане помещения, а также эвакуационные выходы, места размещения средств противопожарной защиты, медицинских средств, средств спасения и связи, правила поведения людей, порядок и последовательность их действий в условиях пожара и\или чрезвычайной ситуации.

На предприятии имеется как план эвакуации при пожаре, так и средства пожаротушения, в виде огнетушителей в каждом помещении, где эксплуатируются ЭВМ. Присутствуют средства пожарной сигнализации, позволяющие оповестить персонал о пожаре и своевременно вызвать экстренные службы.

Персонал периодически проходит инструктаж по технике безопасности, в том числе и противопожарной, а также проводятся учения, для отработки

действий при пожаре.

5.3 Комплексы физических упражнений для сохранения и укрепления индивидуального здоровья и обеспечения полноценной профессиональной деятельности

Регламентированные микропаузы и перерывы целесообразно использовать для выполнения комплексов упражнений и гимнастики для глаз, для снятия утомления с плечевого пояса и рук, для улучшенного мозгового кровообращения. Через 2-3 недели следует менять комплексы упражнений

Комплексы упражнений для глаз:

Упражнения выполняются сидя или стоя, отвернувшись от экрана, при ритмичном дыхании, с максимальной амплитудой движения глаз.

Вариант 1:

1) Закрывать глаза, сильно напрягая глазные мышцы, на счет 1 - 4, затем раскрыть глаза, расслабив мышцы глаз, посмотреть вдаль на счет 1 - 6. Повторить 4 - 5 раз.

2) Посмотреть на переносицу и задержать взор на счет 1 - 4. До усталости глаза не доводить. Затем открыть глаза, посмотреть вдаль на счет 1 - 6. Повторить 4 - 5 раз.

3) Не поворачивая головы, посмотреть направо и зафиксировать взгляд на счет 1 - 4, затем посмотреть вдаль прямо на счет 1 - 6. Аналогичным образом проводятся упражнения, но с фиксацией взгляда влево, вверх и вниз. Повторить 3 - 4 раза.

4) Перенести взгляд быстро по диагонали: направо вверх - налево вниз, потом прямо вдаль на счет 1 - 6; затем налево вверх направо вниз и посмотреть вдаль на счет 1 - 6. Повторить 4 - 5 раз

Физкультминутка способствует снятию локального утомления. По содержанию Физкультминутки различны и предназначаются для конкретного воздействия на ту или иную группу мышц или систему организма в зависимости от самочувствия и ощущения усталости.

Физкультминутка общего воздействия может применяться, когда

физкультпаузу по каким-либо причинам выполнить нет возможности. Существует определённые физкультминутки.

Комплекс общего воздействия № 1:

1) Исходное положение (и.п.) - основная стойка (о.с.) 1 - 2 - встать на носки, руки вверх-наружу, потянуться вверх за руками. 3 - 4 - дугами в стороны руки вниз и расслабленно скрестить перед грудью, голову наклонить вперед. Повторить 6 - 8 раз. Темп быстрый.

2) И.п. - стойка ноги врозь, руки вперед, 1 - поворот туловища направо, мах левой рукой вправо, правой назад за спину. 2 и.п. 3 - 4 - то же в другую сторону. Упражнения выполняются размашисто, динамично. Повторить 6 - 8 раз. Темп быстрый.

3) И.п. 1 - согнуть правую ногу вперед и, обхватив голень руками, притянуть ногу к животу. 2 - приставить ногу, руки вверх-наружу. 3 - 4 - то же другой ногой. Повторить 6 - 8 раз. Темп средний.

Комплекс общего воздействия № 2:

1) И.п. - о.с. 1 - 2 - дугами внутрь два круга руками в лицевой плоскости. 3 - 4 - то же, но круги наружу. Повторить 4 - 6 раз. Темп средний.

2) И.п. - стойка ноги врозь, правую руку вперед, левую на пояс. 1 - 3 - круг правой рукой вниз в боковой плоскости с поворотом туловища направо. 4 - заканчивая круг, правую руку на пояс, левую вперед. То же в другую сторону. Повторить 4 - 6 раз. Темп средний.

3) И.п. - о.с. 1 - с шагом вправо руки в стороны. 2 - два пружинящих наклона вправо. Руки на пояс. 4 - и.п. 1 - 4 - то же влево. Повторить 4 - 6 раз в каждую сторону. Темп средний.

Наклоны и повороты головы оказывают механическое воздействие на стенки шейных кровеносных сосудов, повышают их эластичность; раздражение вестибулярного аппарата вызывает расширение кровеносных сосудов головного мозга. Все это усиливает мозговое кровообращение, повышает его интенсивность и облегчает умственную деятельность.

Физкультурная пауза - повышает двигательную активность,

Физкультурная пауза:

1) Исходное положение (и.п.) - основная стойка (о.с.). 1 - руки вперед, ладони книзу. 2 - руки в стороны, ладони кверху, 3 - встать на носки, руки вверх, прогнуться. 4 - и.п. Повторить 4 - 6 раз. Темп медленный.

2) И.п. - ноги врозь, немного шире плеч. 1 - 3 наклон назад, руки за спину. 3 - 4 - и.п. Повторить 6 - 8 раз. Темп средний.

3) И.п. - ноги на ширине плеч. 1 - руки за голову, поворот туловища направо. 2 - туловище в и.п., руки в стороны, наклон вперед, голову назад. 3 - выпрямиться, руки за голову, поворот туловища налево. 4 - и.п. 5 - 8 - то же в другую сторону. Повторить 6 раз. Темп средний.

4) И.п. - руки к плечам. 1 - выпад вправо, руки в стороны. 2 - и.п. 3 - присесть, руки вверх. 4 - и.п. 5 - 8 - то же в другую сторону. Повторить 6 раз. Темп средний.

5) И.п. - ноги врозь, руки на пояс. 1 - 4 - круговые движения туловищем вправо. 5 - 8 - круговые движения туловищем влево. Повторить 4 раза. Темп средний.

6) И.п. - о.с. 1 - мах правой ногой назад, руки в стороны. 2 - и.п. 3 - 4 - то же левой ногой. Повторить 6 - 8 раз. Темп средний.

7) И.п. - ноги врозь, руки на пояс. 1 - голову наклонить вправо. 2 - не выпрямляя головы, наклонить ее назад. 3 - голову наклонить вперед. 4 - и.п. 5 - 8 - то же в другую сторону. Повторить 4 - 6 раз. Темп средний.

стимулирует деятельность нервной, сердечно-сосудистой, дыхательной и мышечной систем, снимает общее утомление, повышает умственную работоспособность.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Автоматизированные информационные системы: учебник для студ. учреждений сред. проф. образования / К.Н.Мезенцев. – 4-е изд., стер. – М. : Издательский центр «Академия», 2013. – 176 с.

2 Маглинец, Ю.А. Анализ требований к автоматизированным информационным системам. М. Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 191 с.

3 Информационные системы: учебник для вузов / Ю. Избачков [и др.]. – 3-е изд. – СПб.: Питер, 2011. – 544 с.

4 1С: Предприятие [электронный ресурс] // solutions.1c.ru: отраслевые и специализированные решения «1С: Предприятие». Режим доступа: <https://1c.ru/rus/products/1c/predpr/buh44.htm>.

5 1С:ИТС [электронный ресурс]. – Режим доступа: <https://its.1c.ru/db/metod8dev#content:5816:hdoc@1f67734d>.

6 Информационная безопасность и защита информации [электронный ре-сурс]. – Режим доступа: <http://bezopasnik.org>article/book/23.pdf>.

7 Шаньгин, В.Ф. Информационная безопасность и защита информации. Саратов: Профобразование, 2017. –315 с.

8 Маркировка лекарств [электронный ресурс]. – Режим доступа: <https://честныйзнак.рф/business/projects/medicines/>.

9 Краткая инструкция по быстрому старту для изучения API [электронный ресурс]. – Режим доступа: https://честныйзнак.рф/upload/iblock/25b/Kratkaya_instruktsiya_po_bystromu_startu_dlya_izucheniya_API.pdf

10 Паспорта процессов [электронный ресурс]. – Режим доступа: https://честныйзнак.рф/upload/Pasporta_protseessov.pdf

11 Методические рекомендации по работе с маркированными лекарственными препаратами [электронный ресурс]. – Режим доступа:

https://честныйзнак.рф/upload/iblock/ff6/Methodicheskie_rekomendatsii_po_rabote_s_markirovannoy_produktsiey_opublikovano.pdf

ПРИЛОЖЕНИЕ А

Листинг модуля для работы с маркированными лекарственными препаратами

Перем tcr,МК_КлючСессии;

Перем ВремяКлюча;

Функция ПреобразоватьСтрокуИзУТФ8(Стр)

Длина=СтрДлина(Стр);

Итог="";

Для Н=1 По Длина Цикл

Знак=Сред(Стр,Н,1);

Код=КодСимв(Знак);

Если Код<128 Тогда

Итог=Итог+Знак;

ИначеЕсли (Код>=128)И(Код<192) Тогда

Иначе

Н=Н+1;

Знак2=Сред(Стр,Н,1);

Код2=КодСимв(Знак2);

Если Код=208 Тогда

Если Код2=129 Тогда Итог=Итог+"Ё";

Иначе Итог=Итог+Симв(КодСимв("А")+Код2-144);

КонецЕсли;

ИначеЕсли Код=209 Тогда

Если Код2=145 Тогда Итог=Итог+"ё";

Иначе Итог=Итог+Симв(КодСимв("р")+Код2-128);

КонецЕсли;

КонецЕсли;

КонецЕсли;

КонецЦикла;

Возврат Итог;

КонецФункции

Функция ПолучитьДатуВремяXML()

ТекДата=ТекущаяДата();

ТекГод=ДатаГод(ТекДата);

ТекМес=ДатаМесяц(ТекДата);

ТекДень=ДатаЧисло(ТекДата);

ДатаXML=Формат(ТекГод,"Ч(0)4")+ "-" +Формат(ТекМес,"Ч(0)2")+ "-" +Формат(ТекДень,"Ч(0)2")+ "Т"+ТекущееВремя()+ "+09:00";

Возврат ДатаXML;

КонецФункции

Процедура АвторизацияПоФирме()

```
Если ВыбФирма.Выбран()=0 Тогда
    Предупреждение("Не выбрана фирма!!!");
    Возврат;
КонецЕсли;
Если ПустаяСтрока(ВыбФирма.МК_КодУчетнойСистемы)=1 Тогда
    Предупреждение("Не задан код учетной системы!!!");
    Возврат;
КонецЕсли;
Если ПустаяСтрока(ВыбФирма.МК_СекретныйКлюч)=1 Тогда
    Предупреждение("Не задан секретный ключ!!!");
    Возврат;
КонецЕсли;
Если ПустаяСтрока(ВыбФирма.МК_Сертификат)=1 Тогда
    Предупреждение("Не задан сертификат!!!");
    Возврат;
КонецЕсли;
```

Продолжение ПРИЛОЖЕНИЯ А

```
Если ПустаяСтрока(ВыбФирма.ИНН)=1 Тогда
    Предупреждение("Не задан ИНН!!!");
    Возврат;
КонецЕсли;

Если ФС.СуществуетФайл(КаталогИБ()+"_Kripto")=0 Тогда
    ФС.СоздатьКаталог(КаталогИБ()+"_Kripto");
КонецЕсли;
Если
ФС.СуществуетФайл(КаталогИБ()+"_Kripto\"+СокрЛП(ВыбФирма.ИНН))=0
Тогда

    ФС.СоздатьКаталог(КаталогИБ()+"_Kripto\"+СокрЛП(ВыбФирма.ИНН
));
КонецЕсли;
КаталогТемп=КаталогИБ()+"_Kripto\"+СокрЛП(ВыбФирма.ИНН);

СпрФирма = СоздатьОбъект("Справочник.Фирмы");
СпрФирма.НайтиЭлемент(ВыбФирма);
WinHttp = СоздатьОбъект("Msxml2.XMLHTTP.6.0");

//Метод для получения кода аутентификации
СтрЗапрос="http://api.mdlp.crpt.ru/api/v1/auth";
СтрJson="{";
СтрJson=СтрJson+Симв(34)+"client_id"+Симв(34)+": "+Симв(34)+СокрЛ
П(ВыбФирма.МК_КодУчетнойСистемы)+Симв(34)+"; // Учетные системы:
ВАЖНО!!! = идентификатор клиента
```

```

СтрJson=СтрJson+Симв(34)+"client_secret"+Симв(34)+":"+Симв(34)+Со
крЛП(ВыбФирма.МК_СекретныйКлюч)+Симв(34)+"; // Учетные системы:
секретный код
СтрJson=СтрJson+Симв(34)+"user_id"+Симв(34)+":"+Симв(34)+СокрЛ
П(ВыбФирма.МК_Сертификат)+Симв(34)+"; // Сертификат
организации: отпечаток
СтрJson=СтрJson+Симв(34)+"auth_type"+Симв(34)+":"+Симв(34)+"SIG
NED_CODE"+Симв(34);
СтрJson=СтрJson+"}";
WinHttp.Open("POST", СокрЛП(СтрЗапрос), 0,,);
WinHttp.setRequestHeader("Content-type","application/json;charset=UTF-
8");
WinHttp.Send(СтрJson);
Если WinHttp.Status <> 200 Тогда
    Сообщить("Ошибка авторизации: " + WinHttp.Status + " " +
WinHttp.StatusText);
    Сообщить(WinHttp.ResponseText);
    Возврат;
КонецЕсли;
СтрJson=WinHttp.ResponseText;
СтрJson=СтрЗаменить(СтрJson,"{"+Симв(34)+"code"+Симв(34)+":"+Си
мв(34), "");
СтрJson=СтрЗаменить(СтрJson,Симв(34)+"} ", "");
СтрКод=СтрJson;
Текст = СоздатьОбъект("Текст");
Текст.ДобавитьСтроку(СтрJson);
Текст.Записать(КаталогТемп+"\InCode64.txt");
Если ФС.СуществуетФайл(КаталогТемп+"\OutCode64.txt")=1 Тогда
    ФС.УдалитьФайл(КаталогТемп+"\OutCode64.txt");
КонецЕсли;
СтрКоманда=Симв(34)+"C:\Program Files (x86)\Crypto
Pro\CSP\csptest.exe"+Симв(34)+" -sfsign -sign -in
"+КаталогТемп+"\InCode64.txt -out "+КаталогТемп+"\OutCode64.txt -my
"+Симв(34)+СокрЛП(ВыбФирма.ИНН)+Симв(34)+" -detached -base64 -add";
ЗапуститьПриложение(СтрКоманда);
//Сообщить(СтрJson);
//Sert.bat полученный код:InCode64.txt переводит в сигнатуру:
OutCode64.txt
Пока ФС.СуществуетФайл(КаталогТемп+"\OutCode64.txt")=0 Цикл
    Состояние("Получение сигнатуры для кода...");
КонецЦикла;
СтрСерт="";
Текст = СоздатьОбъект("Текст");
Текст.Открыть(КаталогТемп+"\OutCode64.txt"); // Считывается
сигнатура

```

Для Счет=1 По Текст.КоличествоСтрок() Цикл
Стр=Текст.ПолучитьСтроку(Счет);
Продолжение ПРИЛОЖЕНИЯ А

```
СтрСерт=СтрСерт+Стр;  
КонецЦикла;  
СтрJson="{Симв(34)+\"code\"+Симв(34)+\": \"+Симв(34)+СтрКод+Симв(34)+\", \"+Симв(34)+\"signature\"+Симв(34)+\": \"+Симв(34)+СтрСерт+Симв(34)+\"}";  
//Метод для получения ключа сессии  
СтрЗапрос="http://api.mdlp.crpt.ru/api/v1/token";  
WinHttp.Open("POST", СокрЛП(СтрЗапрос), 0,,);  
WinHttp.setRequestHeader("Content-type","application/json;charset=UTF-8");  
WinHttp.Send(СтрJson);  
Если WinHttp.Status <> 200 Тогда  
    Сообщить("Ошибка получения ключа сессии: " + WinHttp.Status + " " + WinHttp.StatusText);  
    Сообщить(WinHttp.ResponseText);  
    Возврат;  
КонецЕсли;  
СтрJson=WinHttp.ResponseText;  
СтрJson=СтрЗаменить(СтрJson,"{","");  
СтрJson=СтрЗаменить(СтрJson,Симв(34),"");  
Поз=Найти(СтрJson,"life_time");  
СтрJson=Сред(СтрJson,1,Поз-1);  
СтрToken=СтрЗаменить(СтрJson,":"," ");  
МК_КлючСессии=СтрToken;  
СпрФирма.МК_КлючСессии=СтрToken;  
СпрФирма.МК_ДатаСессии=ТекущаяДата();  
СпрФирма.МК_ВремяСессии=ТекущееВремя();  
СпрФирма.Записать();  
ВыбФирма=СпрФирма.ТекущийЭлемент();  
Сообщить("Авторизация прошла успешно!!!");  
КонецПроцедуры
```

Процедура

```
СчитатьКодыУпаковокПоКодуАгрегирования(_ВыбSSCC,СтрXML)  
    СпрУп = СоздатьОбъект("Справочник.МК_КодыУпаковок");  
    DOMDocument = СоздатьОбъект("MSXML2.DOMDocument");  
    DOMDocument.loadXML(СтрXML);  
    Если DomDocument.parseError.errorCode <> 0 Тогда  
        Сообщить("Ошибка: XML файл не валидный");  
        Возврат;  
    КонецЕсли;
```

```

УзелTree = DomDocument.SelectNodes("//tree/sgtin/info_sgtin");
СчетКол=_ВыбSSCC.Количество;
Для Индекс = 0 По УзелTree.Length - 1 Цикл
    УзелSgtin=УзелTree.Item(Индекс);
    КодУпаковки=УзелSgtin.ChildNodes.Item(0).Text;
    СпрУп.ИспользоватьВладельца(_ВыбSSCC.Владелец);
    Если
СпрУп.НайтиПоРеквизиту("КодУпаковки",КодУпаковки,0)=0 Тогда
    СпрУп.Новый();
    СпрУп.КодУпаковки=КодУпаковки;
    СпрУп.КодАгрегирования=_ВыбSSCC.КодАгрегирования;
    СпрУп.Производитель=_ВыбSSCC.Производитель;
    СпрУп.Серия=_ВыбSSCC.Серия;
    СпрУп.СрокГодности=_ВыбSSCC.СрокГодности;
    СпрУп.Поставщик=_ВыбSSCC.Поставщик;

    СпрУп.ПриходныйДокумент=_ВыбSSCC.ПриходныйДокумент;
    СпрУп.ФирмаОптовая=_ВыбSSCC.ФирмаОптовая;
    СпрУп.Записать();
    КонецЕсли;
    СчетКол=СчетКол-1;
КонецЦикла;
Если СчетКол>0 Тогда
    Сообщить("Ошибка! Не определены коды упаковок в
количестве:"+СчетКол);
КонецЕсли;

```

Продолжение ПРИЛОЖЕНИЯ А

КонецПроцедуры

```

Процедура ПолучитьРеестрВхДокументовПоФирме()
    Если ВыбФирма.Выбран()=0 Тогда
        Предупреждение("Не выбрана фирма!!!");
        Возврат;
    КонецЕсли;
    МК_КлючСессии=ВыбФирма.МК_КлючСессии;
    Если ВыбФирма.МК_ДатаСессии<ТекущаяДата() Тогда
        МК_КлючСессии="";
    КонецЕсли;
    Если
Число(Сред(ТекущееВремя(),1,2))*60+Число(Сред(ТекущееВремя(),3,2))-
Число(Сред(ВыбФирма.МК_ВремяСессии,1,2))*60-
Число(Сред(ВыбФирма.МК_ВремяСессии,3,2))>ВремяКлюча Тогда
        МК_КлючСессии="";
    КонецЕсли;

```

```

Если ПустаяСтрока(МК_КлючСессии)=1 Тогда
    АвторизацияПоФирме();
    Возврат;
КонецЕсли;
СтрТокен=МК_КлючСессии;
КаталогТемп=КаталогИБ()+"_Kripto\"+СокрЛП(ВыбФирма.ИНН);

WinHttp = СоздатьОбъект("Msxml2.XMLHTTP.6.0");
СтрJson="{ "+Симв(34)+"filter"+Симв(34)+"":{" "+Симв(34)+"doc_status"+
Симв(34)+": "+Симв(34)+"PROCESSED_DOCUMENT"+Симв(34)+"}," "+Симв(
34)+"start_from"+Симв(34)+": 0," +Симв(34)+"count"+Симв(34)+":100}";
СтрЗапрос="https://api.mdlp.crpt.ru/api/v1/documents/income";
WinHttp.Open("POST", СокрЛП(СтрЗапрос), 0,,);
WinHttp.setRequestHeader("Content-type","application/json;charset=UTF-
8");
WinHttp.setRequestHeader("Authorization",СтрТокен);
WinHttp.Send(СтрJson);
Если WinHttp.Status <> 200 Тогда
    Сообщить("HTTP " + WinHttp.Status + " " + WinHttp.StatusText);
    Сообщить("Ошибка: " + WinHttp.ResponseText);
    Возврат;
КонецЕсли;
СтрJson=WinHttp.ResponseText;
СтрJson=СтрЗаменить(СтрJson,Симв(34),"");
ТабДок.Очистить();
НомерСтр=0;
Поз1=Найти(СтрJson,"[");
СтрJson=Сред(СтрJson,Поз1+1);
Поз1=Найти(СтрJson,"{");
Пока Поз1>0 Цикл
    НомерСтр=НомерСтр+1;
    ТабДок.НоваяСтрока();
    Поз2=Найти(СтрJson,"}");
    СтрДок=Сред(СтрJson,Поз1+1,Поз2-Поз1-1);
    Поз3=Найти(СтрДок,":");
    Пока Поз3>0 Цикл
        ИмяРекв=Сред(СтрДок,1,Поз3-1);
        Поз4=Найти(СтрДок,",");
        Если Поз4>0 Тогда
            ЗначРекв=Сред(СтрДок,Поз3+1,Поз4-Поз3-1);
            СтрДок=Сред(СтрДок,Поз4+1);
        Иначе
            ЗначРекв=Сред(СтрДок,Поз3+1);
            СтрДок="";
        КонецЕсли;
    СтрДок=СтрДок+ИмяРекв+ЗначРекв+";";
    Поз1=Найти(СтрДок,"}");
    СтрДок=Сред(СтрДок,Поз1+1);
    Поз1=Найти(СтрДок,"{");
КонецЦикла;

```

Продолжение ПРИЛОЖЕНИЯ А

Если НомерСтр=1 Тогда

ТабДок.НоваяКолонка(ИмяРекв,,,,Макс(СтрДлина(ИмяРекв),СтрДлина(ЗначРекв)),,);

КонецЕсли;

ТабДок.УстановитьЗначение(НомерСтр,ИмяРекв,ЗначРекв);

Поз3=Найти(СтрДок,":");

КонецЦикла;

СтрJson=Сред(СтрJson,Поз2+1);

Поз1=Найти(СтрJson,"{");

КонецЦикла;

КонецПроцедуры

Процедура СкачатьДокументПоИдентификатору()

Текст = СоздатьОбъект("Текст");

Если ВыбФирма.Выбран()=0 Тогда

Предупреждение("Не выбрана фирма!!!");

Возврат;

КонецЕсли;

МК_КлючСессии=ВыбФирма.МК_КлючСессии;

Если ВыбФирма.МК_ДатаСессии<ТекущаяДата() Тогда

МК_КлючСессии="";

КонецЕсли;

Если

Число(Сред(ТекущееВремя(),1,2))*60+Число(Сред(ТекущееВремя(),3,2))-

Число(Сред(ВыбФирма.МК_ВремяСессии,1,2))*60-

Число(Сред(ВыбФирма.МК_ВремяСессии,3,2))>ВремяКлюча Тогда

МК_КлючСессии="";

КонецЕсли;

Если ПустаяСтрока(МК_КлючСессии)=1 Тогда

АвторизацияПоФирме();

Возврат;

КонецЕсли;

СтрТокен=МК_КлючСессии;

WinHttp = СоздатьОбъект("Msxml2.XMLHTTP.6.0");

СтрЗапрос="https://api.mdlp.crpt.ru/api/v1/documents/download/"+СокрЛП(КодДокумента);

WinHttp.Open("GET", СокрЛП(СтрЗапрос), 0,,);

WinHttp.setRequestHeader("Content-type", "application/json;charset=UTF-8");

```

WinHttp.setRequestHeader("Authorization",СтрТокен);
WinHttp.Send();
Если WinHttp.Status <> 200 Тогда
    Сообщить("HTTP " + WinHttp.Status + " " + WinHttp.StatusText);
    Сообщить("Ошибка: " + WinHttp.ResponseText);
    Возврат;
КонецЕсли;
СтрJson=WinHttp.ResponseText;

СтрJson=СтрЗаменить(СтрJson,"{"+Симв(34)+"link"+Симв(34)+":"+Симв(34),
""");
СтрJson=СтрЗаменить(СтрJson,Симв(34)+"}", "");
Сообщить(СтрJson);
СтрЗапрос=СтрJson;
WinHttp.Open("GET", СокрЛП(СтрЗапрос), 0,,);
WinHttp.setRequestHeader("Content-type","application/json;charset=UTF-
8");
WinHttp.setRequestHeader("Authorization",СтрТокен);
WinHttp.Send();
Если WinHttp.Status <> 200 Тогда
    Сообщить("HTTP " + WinHttp.Status + " " + WinHttp.StatusText);
    Сообщить("Ошибка: " + WinHttp.ResponseText);
    Возврат;
КонецЕсли;
СтрJson=WinHttp.ResponseText;

```

Продолжение ПРИЛОЖЕНИЯ А

```

Сообщить("HTTP " + WinHttp.Status + " " + WinHttp.StatusText);
Текст.ДобавитьСтроку(СтрJson);
Текст.Показать();
КонецПроцедуры

```

Функция

```

СкачатьКвитанциюПоИдентификатору(_ВыбФирма, КодДокумента)
Если ПустаяСтрока(_КодДокумента)=1 Тогда
    Предупреждение("Не определен код документа!!!");
    Возврат "";
КонецЕсли;
КаталогТемп=КаталогИБ()+ "_Kripto\"+СокрЛП(_ВыбФирма.ИНН);
Текст = СоздатьОбъект("Текст");
Если _ВыбФирма.Выбран()=0 Тогда
    Предупреждение("Не выбрана фирма!!!");
    Возврат "";
КонецЕсли;
МК_КлючСессии=_ВыбФирма.МК_КлючСессии;

```

```

Если _ВыбФирма.МК_ДатаСессии<ТекущаяДата() Тогда
    МК_КлючСессии="";
    КонецЕсли;
Если
    Число(Сред(ТекущееВремя(),1,2))*60+Число(Сред(ТекущееВремя(),3,2))-
    Число(Сред(_ВыбФирма.МК_ВремяСессии,1,2))*60-
    Число(Сред(_ВыбФирма.МК_ВремяСессии,3,2))>ВремяКлюча Тогда
    МК_КлючСессии="";
    КонецЕсли;
Если ПустаяСтрока(МК_КлючСессии)=1 Тогда
    АвторизацияПоФирме();
    Возврат "";
    КонецЕсли;
СтрТокен=МК_КлючСессии;

WinHttp = СоздатьОбъект("Msxml2.XMLHTTP.6.0");

Для Счет=1 По 2 Цикл

    СтрЗапрос="https://api.mdlp.crpt.ru/api/v1/documents/"+СокрЛПП(_КодДокумента)+"/ticket";
    WinHttp.Open("GET", СокрЛПП(СтрЗапрос), 0,.);
    WinHttp.setRequestHeader("Content-type","application/json;charset=UTF-8");
    WinHttp.setRequestHeader("Authorization",СтрТокен);
    WinHttp.Send();
    Если WinHttp.Status <> 200 Тогда
        ОшибкаСтр=WinHttp.ResponseText;
        Если Найти(ОшибкаСтр,"Unauthorized")>0 Тогда
            АвторизацияПоФирме();
            Возврат "";
        Иначе
            Сообщить("Ошибка запроса квитанции: " +
WinHttp.ResponseText);
            Возврат "";
        КонецЕсли;
    КонецЕсли;
    СтрJson=WinHttp.ResponseText;

    СтрJson=СтрЗаменить(СтрJson,"{"+Симв(34)+"link"+Симв(34)+":"+Симв(34)+"}");
    СтрJson=СтрЗаменить(СтрJson,Симв(34)+"}","");
    Сообщить(СтрJson);
    СтрЗапрос=СтрJson;
    WinHttp.Open("GET", СокрЛПП(СтрЗапрос), 0,.);

```

```

WinHttp.setRequestHeader("Content-
type","application/json;charset=UTF-8");
WinHttp.setRequestHeader("Authorization",СтрТокен);
WinHttp.Send();
Если WinHttp.Status <> 200 Тогда
    Сообщить("Ошибка получения квитанции: " +
WinHttp.ResponseText);
        Продолжение ПРИЛОЖЕНИЯ А

        Возврат "";
КонецЕсли;
Сообщить("HTTP " + WinHttp.Status + " " + WinHttp.StatusText);

СтрJson=WinHttp.ResponseText;
Поз1=Найти(СтрJson,"<operation_comment>");
Поз2=Найти(СтрJson,"</operation_comment>");
СтрКомментарий=Сред(СтрJson,Поз1+19,Поз2-Поз1-19);
Сообщить(СтрКомментарий);
Текст.ДобавитьСтроку(СтрJson);
Текст.Записать("d:\test.xml");
Возврат СтрJson;

```

```

КонецЦикла;
КонецФункции

```

```

Функция ОтправитьДокументНаСайт(ВыбФирма,ИмяФайла)
    КаталогТемп=КаталогИБ()+"_Kripto\"+СокрЛП(ВыбФирма.ИНН);
    ИмяФайлаХml=КаталогТемп+"\\"+ИмяФайла;
    ИмяФайлаSig=СтрЗаменить(ИмяФайлаХml,"xml","sig");
    ИмяФайлаВ64=СтрЗаменить(ИмяФайлаХml,"xml","b64");
    МК_КлючСессии=ВыбФирма.МК_КлючСессии;
    Если ВыбФирма.МК_ДатаСессии<ТекущаяДата() Тогда
        МК_КлючСессии="";
    КонецЕсли;
    Если
        Число(Сред(ТекущееВремя(),1,2))*60+Число(Сред(ТекущееВремя(),3,2))-
        Число(Сред(ВыбФирма.МК_ВремяСессии,1,2))*60-
        Число(Сред(ВыбФирма.МК_ВремяСессии,3,2))>ВремяКлюча Тогда
        МК_КлючСессии="";
    КонецЕсли;
    Если ПустаяСтрока(МК_КлючСессии)=1 Тогда
        АвторизацияПоФирме();
        Возврат "";
    КонецЕсли;
    СтрТокен=МК_КлючСессии;
    try

```

```

fso=СоздатьОбъект("Scripting.FileSystemObject");
text=fso.OpenTextFile(ИмяФайлаВ64, 2, 1);
tcp.Base64_OpenFile(ИмяФайлаXml);
Пока 1=1 Цикл
    стр=tcp.Base64_ReadNextLine();
    Если стр="" Тогда
        Прервать;
    КонецЕсли;
    text.WriteLine(стр);
КонецЦикла;
text.Close();
ехсерт
    Сообщить(ОписаниеОшибки());
endtry;

Если ФС.СуществуетФайл(ИмяФайлаSig)=1 Тогда
    ФС.УдалитьФайл(ИмяФайлаSig);
КонецЕсли;
СтрКоманда=Симв(34)+"C:\Program Files (x86)\Crypto
Pro\CSP\csptest.exe"+Симв(34)+" -sfsign -sign -in "+ИмяФайлаXml+" -out
"+ИмяФайлаSig+" -my "+Симв(34)+СокрЛП(ВыбФирма.ИНН)+Симв(34)+" -
detached -base64 -add";
ЗапуститьПриложение(СтрКоманда);
Пока ФС.СуществуетФайл(ИмяФайлаSig)=0 Цикл
    Состояние("Получение сигнатуры для файла...");
КонецЦикла;

```

Продолжение ПРИЛОЖЕНИЯ А

```

Текст = СоздатьОбъект("Текст");
СтрСигнатура="";
Текст.Открыть(ИмяФайлаSig);
Для Счет=1 По Текст.КоличествоСтрок() Цикл
    Стр=Текст.ПолучитьСтроку(Счет);
    СтрСигнатура=СтрСигнатура+Стр;
КонецЦикла;

СтрДокумент="";
Текст.Открыть(ИмяФайлаВ64);
Для Счет=1 По Текст.КоличествоСтрок() Цикл
    Стр=Текст.ПолучитьСтроку(Счет);
    СтрДокумент=СтрДокумент+Стр;
КонецЦикла;
ObjV7 = СоздатьОбъект("AddIn.V7SysInfo");
СтрUUID = ObjV7.СоздатьGUID();

```

```

СтрJson="{ "+Симв(34)+"document"+Симв(34)+": "+Симв(34)+СтрДокум
ент+Симв(34)+", "+Симв(34)+"sign"+Симв(34)+": "+Симв(34)+СтрСигнатура+
Симв(34)+", "+Симв(34)+"request_id"+Симв(34)+": "+Симв(34)+СтрUUID+Си
мв(34)+"}";
WinHttp = СоздатьОбъект("Msxml2.XMLHTTP.6.0");
СтрЗапрос="https://api.mdlp.crpt.ru/api/v1/documents/send";
WinHttp.Open("POST", СокрЛП(СтрЗапрос), 0,,);
WinHttp.setRequestHeader("Content-type","application/json;charset=UTF-
8");
WinHttp.setRequestHeader("Authorization",СтрТокен);
WinHttp.Send(СтрJson);
Если WinHttp.Status <> 200 Тогда
    Сообщить("HTTP " + WinHttp.Status + " " + WinHttp.StatusText);
    Сообщить("Ошибка: " + WinHttp.ResponseText);
    Возврат "";
КонецЕсли;
СтрJson=WinHttp.ResponseText;
Сообщить(СтрJson);
СтрJson=СтрЗаменить(СтрJson,"{" "+Симв(34)+"document_id"+Симв(34)
+": "+Симв(34), "");
СтрJson=СтрЗаменить(СтрJson,Симв(34)+"} ", "");
Возврат СтрJson;
КонецФункции

```

Процедура ИнформацияПоУпаковке()

```

Если ВыбSGTIN.Выбран()=0 Тогда
    Предупреждение("Не выбран элемент с кодом упаковки!!!");
    Возврат;
КонецЕсли;
ВыбФирма=ВыбSGTIN.ФирмаОптовая;
Если ВыбФирма.Выбран()=0 Тогда
    Предупреждение("У кода агрегации не определена фирма!!!");
    Возврат;
КонецЕсли;
КаталогТемп=КаталогИБ()+"_Kripto\ "+СокрЛП(ВыбФирма.ИНН);
ИмяФайлаХмл=КаталогТемп+"\220_01_sgtin.xml";
Текст = СоздатьОбъект("Текст");
Текст.ДобавитьСтроку("<?xml version="+Симв(34)+"1.0"+Симв(34)+"
encoding="+Симв(34)+"UTF-8"+Симв(34)+"?>");
Текст.ДобавитьСтроку("<documents
xmlns:xsi="+Симв(34)+"http://www.w3.org/2001/XMLSchema-
instance"+Симв(34)+" version="+Симв(34)+"1.35"+Симв(34)+">");
Текст.ДобавитьСтроку("<query_hierarchy_info
action_id="+Симв(34)+"220"+Симв(34)+">");

```

Текст.ДобавитьСтроку("<subject_id>"+СокрЛП(ВыбФирма.МК_КодОп
товогоСклада)+"</subject_id>");
Текст.ДобавитьСтроку("<sgtin>"+СокрЛП(ВыбSGTIN.КодУпаковки)+"
</sgtin>");

Текст.ДобавитьСтроку("</query_hierarchy_info action_id>");

Текст.ДобавитьСтроку("</documents>");

Продолжение ПРИЛОЖЕНИЯ А

Текст.Записать(ИмяФайлаXml);

КодДокумента=ОтправитьДокументНаСайт(ВыбФирма,"220_01_sgtin.x
ml");

СтрКвитанция=СкачатьКвитанциюПоИдентификатору(ВыбФирма,Код
Документа);

КонецПроцедуры

Процедура РасформированиеКодаАгрегации()

Если ВыбSSCC.Выбран()=0 Тогда

Предупреждение("Не выбран элемент с кодом агрегации!!!");

Возврат;

КонецЕсли;

ВыбФирма=ВыбSSCC.ФирмаОптовая;

Если ВыбФирма.Выбран()=0 Тогда

Предупреждение("У кода агрегации не определена фирма!!!");

Возврат;

КонецЕсли;

СчетУп=0;

СпрКодыУп = СоздатьОбъект("Справочник.МК_КодыУпаковок");

СпрКодыУп.ВыбратьЭлементыПоРеквизиту("КодАгрегирования",Выб
SSCC.КодАгрегирования,0,0);

Пока СпрКодыУп.ПолучитьЭлемент()=1 Цикл

СчетУп=СчетУп+1;

КонецЦикла;

Если ВыбSSCC.Количество<>СчетУп Тогда

Предупреждение("У кода агрегации не соответствует кол-во
упаковок!!!");

Возврат;

КонецЕсли;

СпрSSCC = СоздатьОбъект("Справочник.МК_КодыАгрегирования");

СпрSSCC.НайтиЭлемент(ВыбSSCC);

КаталогТемп=КаталогИБ()+"_Kripto"+СокрЛП(ВыбФирма.ИНН);

ИмяФайлаXml=КаталогТемп+"\912_01_sgtin.xml";

Текст = СоздатьОбъект("Текст");

Текст.ДобавитьСтроку("<?xml version="+Симв(34)+"1.0"+Симв(34)+"
encoding="+Симв(34)+"UTF-8"+Симв(34)+"?>");

Текст.ДобавитьСтроку("<documents
xmlns:xsi="+Симв(34)+"http://www.w3.org/2001/XMLSchema-
instance"+Симв(34)+" version="+Симв(34)+"1.35"+Симв(34)+">");
Текст.ДобавитьСтроку("<unit_unpack
action_id="+Симв(34)+"912"+Симв(34)+">");
Текст.ДобавитьСтроку("<subject_id>"+СокрЛП(ВыбФирма.МК_КодОп
товогоСклада)+"</subject_id>");
Текст.ДобавитьСтроку("<operation_date>"+СокрЛП(ПолучитьДатуВре
мяXML())+"</operation_date>");
Текст.ДобавитьСтроку("<sscc>"+СокрЛП(ВыбSSCC.КодАгрегирования
)+"</sscc>");
Текст.ДобавитьСтроку("</unit_unpack>");
Текст.ДобавитьСтроку("</documents>");
Текст.Записать(ИмяФайлаXml);

КодДокумента=ОтправитьДокументНаСайт(ВыбФирма,"912_01_sgtin.x
ml");

СтрКвитанция=СкачатьКвитанциюПоИдентификатору(ВыбФирма,Код
Документа);

КонецПроцедуры

Процедура ЗапросКодовУпПоКодуАгрегации()

Если ВыбSSCC.Выбран()=0 Тогда

Предупреждение("Не выбран элемент с кодом агрегации!!!");

Возврат;

КонецЕсли;

ВыбФирма=ВыбSSCC.ФирмаОптовая;

Если ВыбФирма.Выбран()=0 Тогда

Предупреждение("У кода агрегации не определена фирма!!!");

Возврат;

КонецЕсли;

Продолжение ПРИЛОЖЕНИЯ А

Если ПустаяСтрока(ВыбSSCC.ЗапросПоУпаковкам)=1 Тогда

СпрSSCC =

СоздатьОбъект("Справочник.МК_КодыАгрегирования");

СпрSSCC.НайтиЭлемент(ВыбSSCC);

КаталогТемп=КаталогИБ()+ "_Kripto\"+СокрЛП(ВыбФирма.ИНН);

ИмяФайлаXml=КаталогТемп+"\210_01_sgtin.xml";

Текст = СоздатьОбъект("Текст");

Текст.ДобавитьСтроку("<?xml

version="+Симв(34)+"1.0"+Симв(34)+" encoding="+Симв(34)+"UTF-
8"+Симв(34)+"?>");

```
Текст.ДобавитьСтроку("<documents
xmlns:xsi="+Симв(34)+"http://www.w3.org/2001/XMLSchema-
instance"+Симв(34)+" version="+Симв(34)+"1.35"+Симв(34)+">");
```

```
Текст.ДобавитьСтроку("<query_kiz_info
action_id="+Симв(34)+"210"+Симв(34)+">");
```

```
Текст.ДобавитьСтроку("<subject_id>"+СокрЛП(ВыбФирма.МК_КодОп
товогоСклада)+"</subject_id>");
```

```
Текст.ДобавитьСтроку("<sscc_down>"+СокрЛП(ВыбSSCC.КодАгрегир
ования)+"</sscc_down>");
```

```
Текст.ДобавитьСтроку("</query_kiz_info>");
```

```
Текст.ДобавитьСтроку("</documents>");
```

```
Текст.Записать(ИмяФайлаXml);
```

```
КодДокумента=ОтправитьДокументНаСайт(ВыбФирма,"210_01_sgtin.x
ml");
```

```
СпрSSCC.ЗапросПоУпаковкам=КодДокумента;
```

```
СпрSSCC.Записать();
```

```
ВыбSSCC=СпрSSCC.ТекущийЭлемент();
```

```
КонецЕсли;
```

```
СтрКвитанция=СкачатьКвитанциюПоИдентификатору(ВыбSSCC.Фир
маОптовая,ВыбSSCC.ЗапросПоУпаковкам);
```

```
Если ПустаяСтрока(СтрКвитанция)=0 Тогда
```

```
СчитатьКодыУпаковокПоКодуАгрегирования(ВыбSSCC,СтрКвитанци
я);
```

```
КонецЕсли;
```

```
КонецПроцедуры
```

```
Процедура ОтгрузкаНаАптеку()
```

```
Перем Запрос, ТекстЗапроса, Таб;
```

```
//Создание объекта типа Запрос
```

```
ВыбСтатус="Принят";
```

```
Запрос = СоздатьОбъект("Запрос");
```

```
ТекстЗапроса =
```

```
"//{{ЗАПРОС(ОтгрузкаНаАптеку)
```

```
|Статус = Справочник.МК_КодыУпаковок.Статус;
```

```
|ФирмаОптовая = Справочник.МК_КодыУпаковок.ФирмаОптовая;
```

```
|Аптека = Справочник.МК_КодыУпаковок.Аптека;
```

```
|ТекущийЭлемент = Справочник.МК_КодыУпаковок.ТекущийЭлемент;
```

```
|Группировка ФирмаОптовая без групп;
```

```
|Группировка Аптека без групп;
```

```
|Группировка ТекущийЭлемент без групп;  
|Условие(Статус = ВыбСтатус);  
|"/}}ЗАПРОС
```

```
;
```

```
// Если ошибка в запросе, то выход из процедуры
```

```
Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда
```

```
    Возврат;
```

```
КонецЕсли;
```

```
// Подготовка к заполнению выходных форм данными запроса
```

```
СпрУп = СоздатьОбъект("Справочник.МК_КодыУпаковок");
```

```
СпрДок = СоздатьОбъект("Справочник.МК_Документы");
```

```
СпрДокЭл = СоздатьОбъект("Справочник.МК_ЭлементыДокумента");
```

Продолжение ПРИЛОЖЕНИЯ А

```
Пока Запрос.Группировка(1) = 1 Цикл
```

```
    Пока Запрос.Группировка(2) = 1 Цикл
```

```
        НачатьТранзакцию();
```

```
        СпрДок.Новый();
```

```
        СпрДок.Дата=ТекущаяДата();
```

```
        СпрДок.Фирма=Запрос.ФирмаОптовая;
```

```
        СпрДок.Контрагент=Запрос.Аптека;
```

```
        Если Запрос.Аптека.ОснФирма=Запрос.ФирмаОптовая
```

```
Тогда
```

```
            СпрДок.КодОперацииМК="431"; // Перемещение
```

```
Иначе
```

```
            СпрДок.КодОперацииМК="416"; // Приемка на
```

```
аптеку
```

```
        КонецЕсли;
```

```
        СпрДок.Статус="Новый";
```

```
        СпрДок.Записать();
```

```
СпрДокЭл.ИспользоватьВладельца(СпрДок.ТекущийЭлемент());
```

```
    Пока Запрос.Группировка(3) = 1 Цикл
```

```
        СпрДокЭл.Новый();
```

```
        СпрДокЭл.Элемент=Запрос.ТекущийЭлемент;
```

```
        СпрДокЭл.Записать();
```

```
        СпрУп.НайтиЭлемент(Запрос.ТекущийЭлемент);
```

```
        СпрУп.Статус="Документ";
```

```
        СпрУп.Записать();
```

```
    КонецЦикла;
```

```
    ЗафиксироватьТранзакцию();
```

```
КонецЦикла;
```

```
КонецЦикла;
```

```
КонецПроцедуры
```

Процедура ОтправитьНовыеДокументы()

```
Перем Запрос, ТекстЗапроса;  
//Создание объекта типа Запрос  
ВыбСтатус="Новый";  
Запрос = СоздатьОбъект("Запрос");  
ТекстЗапроса =  
"//{{ЗАПРОС(ОтправитьНовыеДокументы)  
|ТекущийЭлемент = Справочник.МК_Документы.ТекущийЭлемент;  
|Фирма = Справочник.МК_Документы.Фирма;  
|Статус = Справочник.МК_Документы.Статус;  
|Группировка Фирма без групп;  
|Группировка ТекущийЭлемент без групп;  
|Условие(Статус = ВыбСтатус);  
|//}}ЗАПРОС  
;  
// Если ошибка в запросе, то выход из процедуры  
Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда  
    Возврат;  
КонецЕсли;  
СпрДок = СоздатьОбъект("Справочник.МК_Документы");  
СпрДокЭл = СоздатьОбъект("Справочник.МК_ЭлементыДокумента");  
Пока Запрос.Группировка(1) = 1 Цикл
```

```
КаталогТемп=КаталогИБ()+"_Kripto\"+СокрЛП(Запрос.Фирма.ИНН);  
Пока Запрос.Группировка(2) = 1 Цикл
```

Если

СокрЛП(Запрос.ТекущийЭлемент.КодОперацииМК)="431" Тогда

//Перемещение

```
ИмяФайлаХмл=КаталогТемп+"\431_01_sgtin.xml";
```

```
Текст = СоздатьОбъект("Текст");
```

```
Текст.ДобавитьСтроку("<?xml
```

```
version="+Симв(34)+"1.0"+Симв(34)+" encoding="+Симв(34)+"UTF-  
8"+Симв(34)+"?>");
```

Продолжение ПРИЛОЖЕНИЯ А

```
Текст.ДобавитьСтроку("<documents  
xmlns:xsi="+Симв(34)+"http://www.w3.org/2001/XMLSchema-  
instance"+Симв(34)+" version="+Симв(34)+"1.35"+Симв(34)+">");
```

```
Текст.ДобавитьСтроку("<move_place  
action_id="+Симв(34)+"431"+Симв(34)+">");
```

Текст.ДобавитьСтроку("<subject_id>"+СокрЛП(Запрос.Фирма.МК_Код
ОптовогоСклада)+"</subject_id>");

Текст.ДобавитьСтроку("<receiver_id>"+СокрЛП(Запрос.ТекущийЭлеме
нт.Контрагент.МК_МестоДеятельности)+"</receiver_id>");

Текст.ДобавитьСтроку("<operation_date>"+СокрЛП(ПолучитьДатуВре
мяXML())+"</operation_date>");

Текст.ДобавитьСтроку("<doc_num>"+Формат(Запрос.ТекущийЭлемент
.Код,"Ч(0)10")+"</doc_num>");

Текст.ДобавитьСтроку("<doc_date>"+Формат(Запрос.ТекущийЭлемент.
Дата,"ДДДММГГГГ")+"</doc_date>");

Текст.ДобавитьСтроку("<order_details>");

СпрДокЭл.ИспользоватьВладельца(Запрос.ТекущийЭлемент);

СпрДокЭл.ВыбратьЭлементы(1);

Пока СпрДокЭл.ПолучитьЭлемент()=1 Цикл

Если

СпрДокЭл.Элемент.Вид()=="МК_КодыУпаковок" Тогда

Текст.ДобавитьСтроку("<sgtin>"+СокрЛП(СпрДокЭл.Элемент.КодУпа
ковки)+"</sgtin>");

КонецЕсли;

КонецЦикла;

Текст.ДобавитьСтроку("</order_details>");

Текст.ДобавитьСтроку("</move_place>");

Текст.ДобавитьСтроку("</documents>");

Текст.Записать(ИмяФайлаXml);

КодДокумента=ОтправитьДокументНаСайт(Запрос.Фирма,"431_01_sgt
in.xml");

Если ПустаяСтрока(КодДокумента)=0 Тогда

СпрДок.НайтиЭлемент(Запрос.ТекущийЭлемент);

СпрДок.КодДокумента=КодДокумента;

СпрДок.Статус="Отправлен";

СпрДок.Записать();

КонецЕсли;

КонецЕсли;

КонецЦикла;

КонецЦикла;

КонецПроцедуры

```

Процедура КонтрольОтправленныхДокументов()
  ВыбСтатус="Отправлен";
  Запрос = СоздатьОбъект("Запрос");
  ТекстЗапроса =
  "///{{ЗАПРОС(ОтправитьНовыеДокументы)
|ТекущийЭлемент = Справочник.МК_Документы.ТекущийЭлемент;
|Фирма = Справочник.МК_Документы.Фирма;
|Статус = Справочник.МК_Документы.Статус;
|Группировка Фирма без групп;
|Группировка ТекущийЭлемент без групп;
|Условие(Статус = ВыбСтатус);
|"}}}ЗАПРОС
;
// Если ошибка в запросе, то выход из процедуры
        Продолжение ПРИЛОЖЕНИЯ А

Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда
  Возврат;
КонецЕсли;
СпрУп = СоздатьОбъект("Справочник.МК_КодыУпаковок");
СпрДок = СоздатьОбъект("Справочник.МК_Документы");
СпрДокЭл = СоздатьОбъект("Справочник.МК_ЭлементыДокумента");
Пока Запрос.Группировка(1) = 1 Цикл

  КаталогТемп=КаталогИБ()+"_Kripto\"+СокрЛП(Запрос.Фирма.ИНН);
  Пока Запрос.Группировка(2) = 1 Цикл

    СтрКвитанция=СкачатьКвитанциюПоИдентификатору(ВыбФирма,Запрос.ТекущийЭлемент.КодДокумента);
      Если ПустаяСтрока(СтрКвитанция)=0 Тогда
        СтрРезультат=""; СтрКомментарий="";
        Поз1=Найти(СтрКвитанция,"<operation_result>");
        Поз2=Найти(СтрКвитанция,"</operation_result>");
        Если Поз1>0 Тогда

          СтрРезультат=Сред(СтрКвитанция,Поз1+18,Поз2-Поз1-18);
          КонецЕсли;
          Поз1=Найти(СтрКвитанция,"<operation_comment>");
          Поз2=Найти(СтрКвитанция,"</operation_comment>");
          Если Поз1>0 Тогда

            СтрКомментарий=Сред(СтрКвитанция,Поз1+19,Поз2-Поз1-19);
            КонецЕсли;
            СпрДок.НайтиЭлемент(Запрос.ТекущийЭлемент);

```

```

        СпрДок.Комментарий=["+СтрРезультат+"]
"+СтрКомментарий;
        Если СтрРезультат="Accepted" Тогда
            Если
                СокрЛП(Запрос.ТекущийЭлемент.КодОперацииМК)="431" Тогда
//Перемещение
                    СпрДок.Статус="Одобен";

                СпрДокЭл.ИспользоватьВладельца(Запрос.ТекущийЭлемент);
                СпрДокЭл.ВыбратьЭлементы(1);
                Пока СпрДокЭл.ПолучитьЭлемент()=1
Цикл
                    Если
                        СпрДокЭл.Элемент.Вид()=="МК_КодыУпаковок" Тогда

                            СпрУп.НайтиЭлемент(СпрДокЭл.Элемент);
                                СпрУп.Статус="Одобен";
                                СпрУп.Записать();
                                    КонечЕсли;
                                        КонечЦикла;
                                            КонечЕсли;
                                                КонечЕсли;
                                                    СпрДок.Записать();
                                                        КонечЕсли;
                                                            КонечЦикла;
                                                                КонечЦикла;
                                                                    КонечПроцедуры

```

```

//*****
// Процедура генерации запроса УдалитьДокументы.
//

```

```

Процедура УдалитьДокументы()
    Перец Запрос, ТекстЗапроса, Таб;
    //Создание объекта типа Запрос
    Запрос = СоздатьОбъект("Запрос");
    ТекстЗапроса =
        "{ЗАПРОС(УдалитьДокументы)
        |МК_Документы = Справочник.МК_Документы.ТекущийЭлемент;
        Продолжение ПРИЛОЖЕНИЯ А

```

```

    |Группировка МК_Документы;
    |"}ЗАПРОС
    ;

```

```

// Если ошибка в запросе, то выход из процедуры
Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда

```

Возврат;
КонецЕсли;

```
// Подготовка к заполнению выходных форм данными запроса
Спр = СоздатьОбъект("Справочник.МК_Документы");
Таб = СоздатьОбъект("Таблица");
Таб.ИсходнаяТаблица("УдалитьДокументы");
// Заполнение полей "Заголовков"
Таб.ВывестиСекцию("Заголовков");
Состояние("Заполнение выходной таблицы...");
Таб.Опции(0, 0, Таб.ВысотаТаблицы(), 0);
Пока Запрос.Группировка(1) = 1 Цикл
    // Заполнение полей МК_Документы
    Таб.ВывестиСекцию("МК_Документы");
    Спр.НайтиЭлемент(Запрос.МК_Документы);
    Спр.Удалить(1);
КонецЦикла;
// Вывод заполненной формы
Таб.ТолькоПросмотр(1);
Таб.Показать("УдалитьДокументы", "");
КонецПроцедуры
```

ВремяКлюча=59;

```
Если ЗагрузитьВнешнююКомпоненту("Base64.dll")=0 Тогда
    Сообщить("Не удастся загрузить Base64.dll !!");
КонецЕсли;
```

Попытка

```
tcp = СоздатьОбъект("AddIn.Base64");
```

Исключение

```
Сообщить("Не удастся создать объект AddIn.Base64... !!");
```

КонецПопытки;

```
ЗагрузитьВнешнююКомпоненту(КаталогИБ()+"v7plus.dll");
```

ПРИЛОЖЕНИЕ Б

Справка-подтверждение разработки модуля

Общество с ограниченной
ответственностью

«Гармония»,

Отдел автоматизации

Информационных систем

Начальник отдела:

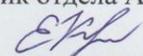
Канухин Е. Е. 

СПРАВКА-ПОДТВЕРЖДЕНИЕ

выдана Мышалову Александру Федоровичу, студенту 4-го курса ФГБОУ ВО «Амурский государственный университет», проходившему преддипломную практику в отделе автоматизации информационных систем с 16 мая 2020 года по 13 июня 2020 года, в том, что он действительно является разработчиком модуля для работы с маркированными лекарственными препаратами, маркировка которых вводится с 1-го июля 2020 года, в системе 1С:Предприятие.



Начальник отдела АИС

Канухин Е.Е. 

12.06.2020г