

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра Прикладная математика и информатика
(наименование)

09.03.03 «Прикладная информатика»
(код и наименование направления подготовки, специальности)

Прикладная информатика в социальной сфере
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка приложения для автоматизации формирования отчетности
seo-специалиста в ООО «Феникс» на платформе 1С.

Студент

С.В. Маслов.

(И.О. Фамилия)

(личная подпись)

Руководитель

д.т.н., доцент С.В. Мкртычев.

(ученая степень, звание, И.О. Фамилия)

Консультант

И.Ю. Усатова

(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

Аннотация

Бакалаврская работа на тему: «Разработка приложения для автоматизации формирования отчетности seo-специалиста в ООО «Маслов Информационные Системы» на платформе 1С.

Структура работы представлена введением, 4 главами, заключением, списком литературы и приложением.

Работа изложена на 52 страницах с приложением и включает 39 рисунков, 4 таблицы и 23 источника.

Введение определяет концепции клиент - серверного - приложения, актуальность темы, цели и задачи заданные в работе, объект и предмет исследования.

В первой главе представлен анализ предметной области, функциональные требования к разрабатываемому продукту его назначение, функциональные требования, системные требования и интерфейс.

Во второй главе идет обоснование выбора платформы для разработки, описание архитектуры и принципов функционирования проектируемой системы, разработка конфигурации и логической модели.

В третьей главе осуществляется: Разработка компонентов объекта справочника, разработка пользовательского интерфейса системы, разработка обработки автоматизации «Яндекс. Метрика», разработка компонента «Отчет Яндекс. Метрики», тестирование приложения.

В четвертой главе идет расчет экономической целесообразности разработки данного программного обеспечения, определения затрат на разработку и отладку, расчет суммарных затрат на разработку.

Данная работа представляет интерес для широкого круга web, seo и 1С - разработчиков, студентов ИТ - направлений, и тех, кто просто интересуется платформой 1С и seo - продвижением.

Abstract

Bachelor's work on the topic: "Development of an application for automating the formation of reports of an seo specialist in LLC "Phoenix" on the 1C platform. The paper consists 4 tables, 39 images, 23 sources. The total volume of bachelor's work is 52 pages.

The structure of the work is presented: Introduction, 4 chapters, conclusion, list of references and appendix. The introduction defines the concepts of the client-server application, the relevance of the topic, the goals and objectives set in the work, the object and subject of research.

The first Chapter presents an analysis of the subject area, functional requirements for the product being developed, its purpose, functional requirements, system requirements, and interface. The second Chapter explains the choice of platform for development, describes the architecture and principles of operation of the designed system, and develops the configuration and logical model.

The third Chapter includes: development of components of the reference object, development of the user interface of the system, development of automation processing "Yandex.Metrica, development of the Yandex Report component. Metrics", testing the app.

In the fourth Chapter, we calculate the economic feasibility of developing this software, determine the cost of development and debugging, and calculate the total cost of development.

This work is of interest to a wide range of web, seo and 1C developers, students of it areas, and those who are simply interested in the 1C platform and seo promotion.

Оглавление

Глава 1 Анализ предметной области.....	6
1.1. Анализ предпроектной ситуации.....	6
1.2. Техническое задание	6
1.2.1 Назначение разработки.....	6
Надежность и сохранность данных:.....	8
Глава 2 Проектирование информационной системы	9
2.1. Описание системы «1С: Предприятие»	9
2.2 Разработка и описание архитектуры и принципов функционирования проектируемой системы	10
2.3 Разработка конфигурации системы.....	15
2.4 Разработка логической модели	17
Глава 3 Разработка компонентов информационной системы	20
3.1 Разработка компонента «Справочники».....	20
3.4 Разработка пользовательского интерфейса системы	22
3.5 Разработка обработки «Авторизация Яндекс Метрика».....	27
3.6 Разработка компонента «Отчет Яндекс метрики».....	30
3.7.1 Тестирование суммы метрики за период.....	36
3.7.2 Тестирование метрики по группировке	37
3.7.3 Тестирование сохранения.....	38
Глава 4 Экономическая часть	39
4.1 Характеристики результатов разработки	39
Заключение	42
Список используемой литературы	43
Приложение А Фрагменты программного кода	45

Введение

В современных условиях экономики оптимальное функционирование осуществляется за счет автоматизированных информационных систем. Применение автоматизированных систем обусловлено усложнением экономики, увеличением количества и видов информации. Особое значение информационные системы имеют в продвижении интернет-услуг, как работа seo-специалиста целиком происходит в различных системах. От удобства, практичности и функциональности систем зависит сколько времени сотрудник будет затрачивать на выполнение административных функций и тайм менеджмента. Так же с развитием и появлением новых ресурсов должны разрабатываться и системы работы с ними. На данный момент формирование отчетности занимает значительное время - около двух часов в день, что можно сократить при использовании более удобной системы.

Наша разработка ориентирована для внутреннего использования, автоматизации работы seo-специалиста в ООО "Феникс".

Объектом исследования является формирования отчетности seo-специалиста в ООО "Феникс".

Предметом исследования является клиент-серверное приложение.

Цель бакалаврской работы - разработка информационной системы seo-специалиста в ООО "Маслов Информационные Системы".

Для достижения поставленной цели выпускной квалификационной работы нужно решить следующие задачи:

- Произвести анализ предметной области.
- Сформулировать требования к клиент-серверному приложению.
- Изучить работу API Яндекс Метрика и OAuth.
- Реализовать и протестировать приложение.

Глава 1 Анализ предметной области

1.1. Анализ предпроектной ситуации

Для обеспечения эффективной работы специалиста в настоящем и будущем необходима разработка информационных систем по повышению эффективности.

Для автоматизации работы seo-специалиста используются следующие объекты учета:

1. Список профилей. Для каждой компании или сайта может использоваться отдельный аккаунт. Для их хранения используются различные сервисы или бумажный вариант.
2. Работа происходит только в режиме «онлайн». Фактически, при отсутствующем интернете работа будет остановлена.
3. Форматирование полученных отчетов ограничено ресурсом в котором происходит работа и часто бывает недоступно.
4. В работе используются ресурсы, которые не имеют интеграции.

1.2. Техническое задание

1.2.1 Назначение разработки

Перед нами стоит задача создания практичной, быстродействующей информационной системы предназначенной для автоматизации процессов сохранения аутентификационных данных рекламных проектов, получения и записи данных метрики.

Разрабатываемый программный продукт должен выполнять следующие функции:

- Ввод и хранение данных seo-специалистов.
- Ввод и хранение данных ID приложения, токена, номера счетчиков.

- Переход на страницы получения данных.
- Получение данных метрик в различных параметрах.
- Отображение данных метрик.
- Форматирование полученного отчета.
- Сохранение отчета в базу 1С.
- Просмотр отчета.
- Сохранение отчета в память компьютера.

В качестве входных данных в конфигурации:

- Информация о seo-специалистах.
- Информация ID приложения, токена, номера счетчиков.
- Информация доступных метрик и группировок.

На выходе получаем следующую информацию:

- Получение отчета метрик.

Системные требования для пользовательского компьютера:

- Процессор Intel Pentium Celeron 1800 МГц и выше
- Оперативная память 256 Мб и выше
- Жесткий диск 40Гб и выше
- Устройство чтения компакт-дисков
- USB-порт
- SVGA-видеокарта

Компьютеры должны быть укомплектованы мышью, клавиатурой, сетевыми шнурами. Компьютер должен предоставляться для сертификации в той программно-аппаратной конфигурации, в которой он будет поставляться потребителю. Вместе с компьютером должен поставляться комплект необходимых драйверов под соответствующие операционные системы.

Конфигурация компьютера должна соответствовать спецификации (Hardware Compatibility List) для используемых операционных систем.

Автоматизированная система должна обладать дружелюбным, интуитивно понятным интерфейсом, быть максимально защищенной от

неправильных действий пользователя. Для работы требуются минимальные знания пользователя ПК и базовые знания работы в 1С.

Надежность и сохранность данных:

– Надежность программного продукта определяется его отказоустойчивостью и возможностью быстрого восстановления. Программный продукт 1с за счет внутрисистемных механизмов обеспечивают высокий уровень надёжности.

– Система предназначена для постоянной, ежедневной работы основных подразделения пищевого бизнеса. Пользователи работают в диалоговом режиме в реальном масштабе времени с системой, функционирующей на сервере базы данных.

– Сервер базы данных Системы должен работать в непрерывном круглосуточном режиме, кроме периодов восстановительных работ.

– Диагностика осуществляется как внутрисистемным механизмом, обеспечивающим сбор информации о возникающих ошибках.

Выводы к главе 1

Исходя из этих данных можно сделать выводы, что часть данных хранится в сторонних сервисах, либо на бумажных носителях. Так же создаётся привязка к определённому специалисту, что создаст трудности при его уходе или замене. Так же все данные получаются в режиме «онлайн» и могут быть сохранены только на внешние носители. Для учета данных и автоматизации процессов на основе платформы 1С будет создана конфигурация «Формирование отчетности SEO-специалиста».

Глава 2 Проектирование информационной системы

2.1. Описание системы «1С: Предприятие»

В данный момент на рынке имеется огромное количество конфигураций, предназначенных для автоматизации процессов ведения хозяйственной деятельности предприятия. Большое распространение в России имеет платформа 1С:Предприятие. Гибкость и универсальность этой программы позволяет создать необходимую для организации систему управления и учета данных. Пользователи работают с уже готовой конфигурацией, разработанной в системе 1С:Предприятие, которая включает в себя несколько элементов:

- Технологическая платформа
- Прикладные решения
- Методология
- ИТС

Данная структура системы необходима для решения многих задач и имеет ряд преимуществ, приведенных ниже:

- Система имеет единый стандарт разработки, фундаментальный набор типовых решений.

- Платформа имеет большой уровень адаптации для решения большинства задач, поставленных пользователем. Принцип её построения позволяет оперативно реагировать на необходимые изменения потребностей в ходе её эксплуатации.

- Система удобна для изменения конфигурации программистом, не принимавшим участие в её разработке. Чаще всего это достоинство можно заметить при использовании прикладных решений в сфере экономики, где не нужно создавать новые, а усовершенствовать типовые решения.

- Данная программа имеет невысокий порог вхождения и простоту языка программирования, тем самым, она является доступной для

разработчиков, имеющих разный профессиональный уровень.

2.2 Разработка и описание архитектуры и принципов функционирования проектируемой системы

Платформа 1С состоит из различных объектов, подразделяющихся на два вида: общих и прикладных.

Общими объектами конфигурации являются:

- Подсистемы, которые объединяют различные объекты;
- Общие модули, хранящие информацию всех процедур и данных, которые можно получить из любой точки доступа в конфигурацию;
- Роли, распределяющие первоочередность получения доступа к информации данной конфигурации;
- Критерии отбора, которые отбирают необходимые данные по заданным условиям;
- Интерфейсы, осуществляющие взаимосвязь пользователей программы 1С с помощью меню и панели управления;
- Общие формы, многократно используемые другими элементами конфигурации, чтобы производить расчеты одинакового типа;
- Общие макеты, используемые в качестве стандартных форм, которые могут использоваться любыми элементами конфигурации;
- Общие картинки, которые на основе графических файлов создают наглядные изображения, которые заменяют надпись текстов более понятным интерфейсом;
- Стили, предназначенные для различного оформления элементов управления;
- Языки, помогающие осуществлять пользование программой на различных языках.

К прикладным объектам конфигурации относятся:

- Константы, хранящие постоянные значения конфигурации, которые используются при различных операциях;
- Справочники, содержащую информацию для разных объектов конфигурации;
- Документы, которые представляют собой различные формы документов, используемых в программе 1С;
- Журналы документов, объединяющие различные виды документов по определенным критериям;
- Перечисления, хранящие значения переменных для подстановки в данные для просчета;
- Отчеты, которые представляют собой обобщенную форму для анализа обрабатываемого материала;
- Обработки, представляющие собой отдельные элементы, используемые для специфических задач, которые не использованы в существующих объектах конфигурации;
- Планы видов характеристик, дающие возможность вести аналитический учет с помощью данной программы;
- Планы счетов, которые хранят информацию по ведению бухгалтерских счетов организации;
- Планы видов расчета, дающие возможность осуществлять расчет организации с различными субъектами;
- Регистры сведений, которые имеют функцию хранения данных объектов с учетом реального времени;
- Регистры накопления, используются в качестве накопителей цифровой информации с учетом реального времени;
- Регистры бухгалтерии, представляющие собой объекты для хранения проведенных бухгалтерских проводок по различным планам счетов.
- Конфигурация, созданная на основе вышеуказанных общих и

прикладных объектов, осуществляет ведение бухгалтерского учета и управление предприятием.

– Документы – это фундамент, на котором работает конфигурация. Поэтому важно правильно оформить документ для ввода данных в систему, чтобы после этого его можно было обработать. Верно сформированный документ дает возможность анализировать работу организации и делать результирующие выводы.

Любой документ представляет из себя форму или макет для ввода реквизитов, который система автоматически запрашивает. Документ содержит:

- Шапку, содержащую общие сведения о нумерации, датировании, наименовании, либо других данных документа;
- Список товаров, представленных в виде таблицы, содержащие данные о наименовании, количестве, цене, стоимости товара;
- Список услуг, который описывает расходы на транспортные услуги, их количество, стоимость, единицы измерения.
- Дополнительные сведения, которые представляют собой добавочные сведения к уже введенным.

В «1С-Предприятие 8.3» существуют следующие объекты, являющиеся моделями для управления документами:

- ДокументМенеджер дает возможность попасть в любой документ, его форме или макету, находящемуся в конфигурации;
- ДокументОбъект позволяет найти и изменить необходимый документ;
- ДокументСсылка выбирает необходимый для просмотра любой документ, ничего в нем не меняя;
- ДокументСписок упорядочивает список однотипных документов в виде таблицы;
- ДокументВыборка выбирает из всей базы необходимый перечень

документов.

В данном примере при создании нового документа указывается текущая дата с помощью встроенной функции ТекущаяДата. Далее указывается номер документа. Номер отличается типом. Он может быть числовым либо строковым. В первом типе используются для нумерации только числа, во втором типе участвуют кроме чисел, еще и буквы. Такой тип чаще всего используется на предприятии. Кроме этого указывается комментарий и вид операции. Значение вида операции вводится с помощью встроенного в конфигурацию перечисления ВидыОперацийЗаказПокупателя.

После создания нового документа можно вывести форму документа на экран монитора, чтобы пользователь мог самостоятельно ее заполнить.

После создания и вывода на экран монитора нового документа «ВнутреннийЗаказ» можно ввести необходимые сведения и сохранить документ в базе данных. При этом для выбора номера используется метод «УстановитьНовыйНомер», а для получения формы используется метод «ПолучитьФорму», который кроме обязательных параметров имеет необязательные: ссылку на первоначальную родительскую форму и ключ идентификации, чтобы ускорить процесс обращения к форме.

Документ имеет одну основную и несколько дополнительных форм, причем каждая форма строго имеет свое название в типовых конфигурациях. Если же разработчик, выполняя заказ, создает новую конфигурацию, то он присваивает создаваемым формам имена по желанию заказчика. Поэтому прежде, чем работать с документом, надо изучить его устройство. После создания нового документа, его сохраняют в базе данных и при необходимости делают его проводку. В каждом документе хранится различная информация, которая используется для анализа данных и составления отчетов. Накопление, отыскание, использование информации осуществляется с помощью регистров, разработанные фирмой 1С. Сведения о документе и его проведение записываются в определенные регистры, связанные между собой, которые создаются при построении конфигурации

на первоначальном этапе.

Сохранение документа в базе данных осуществляется методом Записать. Пример записи документа представлен в следующем листинге. Запись документа (Листинг 1).

```
НовыйАвансовыйОтчет =  
Документы.АвансовыйОтчет.СоздатьДокумент( ); // Установить текущую  
дату НовыйАвансовыйОтчет.Дата=ТекущаяДата( ); //Установить номер для  
документа НовыйАвансовыйОтчет.Номер=( ); //Записать документ  
НовыйАвансовыйОтчет.Записать( );
```

Листинг 1. – Сохранение документа.

После создания нового документа «АвансовыйОтчет», после введения его даты и номера, надо будет его сохранить в базе данных с помощью метода «Записать», который имеет необязательные параметры. Один из них устанавливает режим записи: запись и проведение документа или только запись. Проведение документа подразумевает движение документа в системе и объединение его с другими документами в конфигурации.

Если создается новый документ, первым шагом его сохраняют или записывают в базу данных, вторым пунктом его проводят. При уже существующем документе его препроводят. Управление блокировкой документа. Сначала создается новый документ, который заполняется данными и сохраняется в базе данных. Потом по номеру искомого документа получают ссылку на существующий документ с помощью метода НайтиПоНомеру. Внесение изменений в документ производится после выведения документа на экран монитора с помощью метода ПолучитьОбъект. Далее при возникновении ошибки выполняется попытка блокировки документа. Если блокировка не получается, приходит сообщение об ошибке, если блокировка получается, то меняется номер документа, а измененный документ записывается в базу и его надо разблокировать, используя метод Разблокировать. Наличие блокировки проверяется методом Зabloкирован, возвращающий истинное или ложное значение, зависящее от

того, в каком состоянии находится блокировка. Применение данного метода не всегда оправдано, для этих целей используется обработчик ошибок.

2.3 Разработка конфигурации системы

Подсистемы – это общие объекты конфигурации. На их основе платформа формирует командный интерфейс прикладного решения и визуально разделяет всю функциональность программы на крупные и мелкие блоки. Общая структура конфигурации представлена на рисунке 1.

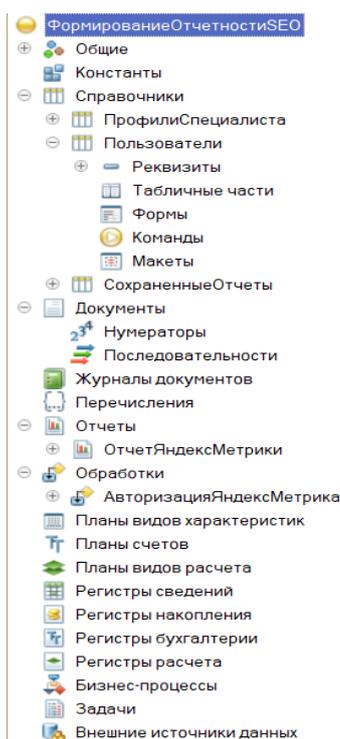


Рисунок 1 – Структура конфигурации «Формирование отчетности SEO-специалиста»

Для реализации программного продукта, созданы следующие объекты конфигурации:

1. Общие.

включает в себя элемент «Подсистемы», состоящий из элементов:

- Справочники;
- Обработки;

На рисунке 2 представлен скриншот с перечисленными объектами конфигурации.

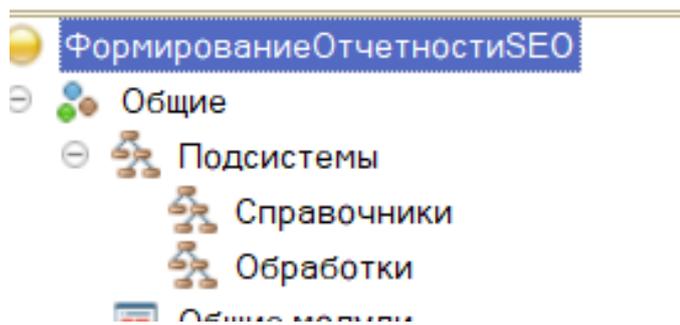


Рисунок 2 – Подсистемы конфигурации

2. Справочники.

Перечень справочников данного объекта конфигурации выглядит следующим образом:

- Пользователи;
- Профили специалиста;
- Сохраненные отчеты;

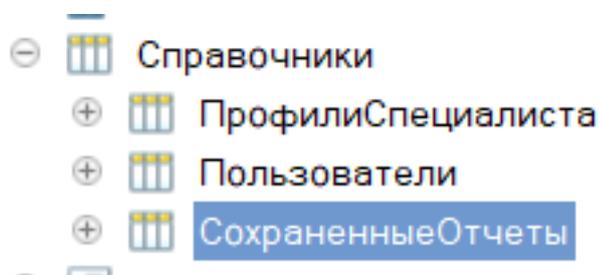


Рисунок 3 – Список справочников

1. Отчеты

- Отчет Яндекс Метрики

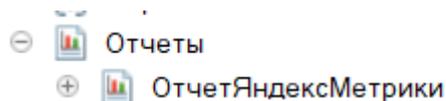


Рисунок 4 – Список отчетов конфигурации

2. Обработки

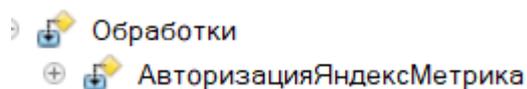


Рисунок 5 – Список обработок конфигурации



Рисунок 6 – Дерево объектов конфигурации.

2.4 Разработка логической модели

Обработка (преобразование) информации — это процесс изменения представления информации или его содержания. Обрабатывать можно информацию любого вида, и правила обработки могут быть самыми разнообразными. В результате обработки имеющейся информации мы получаем новую информацию.

Во многих задачах бывает заранее известно правило, по которому следует осуществлять преобразование входной информации в выходную. Это правило может быть представлено в виде формулы или подробного плана действий.

Обработка информации — это решение информационной задачи, или процесс перехода от исходных данных к результату.

Процесс обработки информации не всегда связан с получением каких-то новых сведений. Например, при переводе текста с одного языка на другой. Обработка информации, связанная с изменением её формы, но не изменяющая содержания, происходит при систематизации информации, поиске информации, кодировании информации.

Обработка информации – это:

- Представление и преобразование информации из одного вида в другой в соответствии с формальными правилами;
- Процесс интерпретации (осмысления) данных;
- Процесс преобразования к виду, удобному для передачи или восприятия (кодирование, декодирование и т.д.);
- Процесс преднамеренного искажения или изменения структуры данных, изменение числовых значений данных и т.д.

Для работы с API Яндекс метрики используется OAuth 2.0.

OAuth 2.0 — протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе.

После создания и настройки профиля OAuth, полученный токен вставляется в заголовки запроса к API Яндекс метрики.

В качестве логической модели разработана схема преобразования данных.

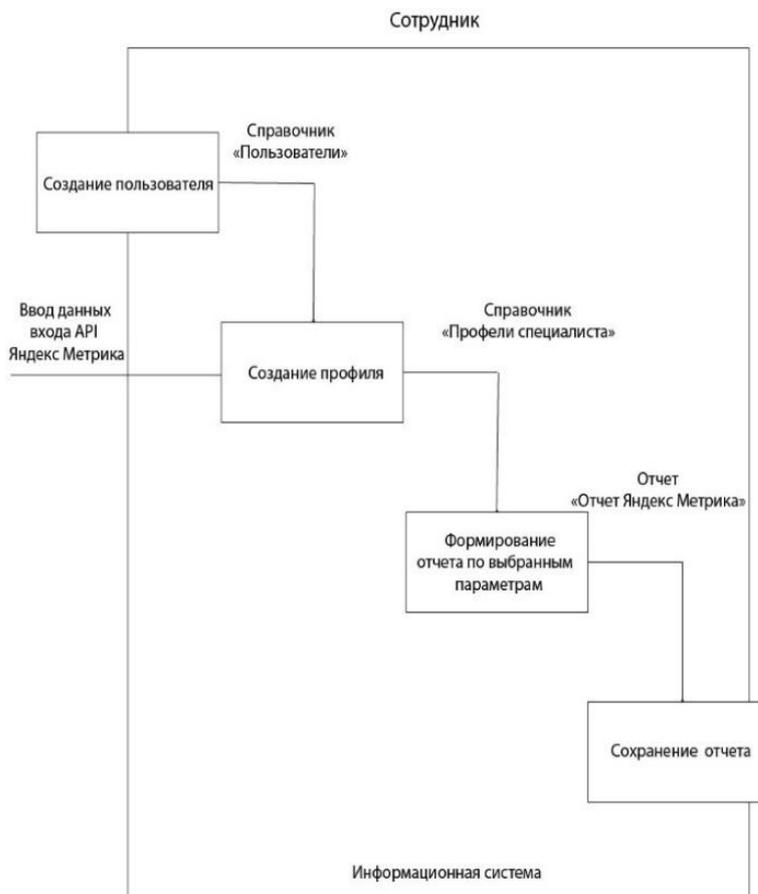


Рисунок 7 – Схема преобразования данных

Так же составлена контекстная диаграмма

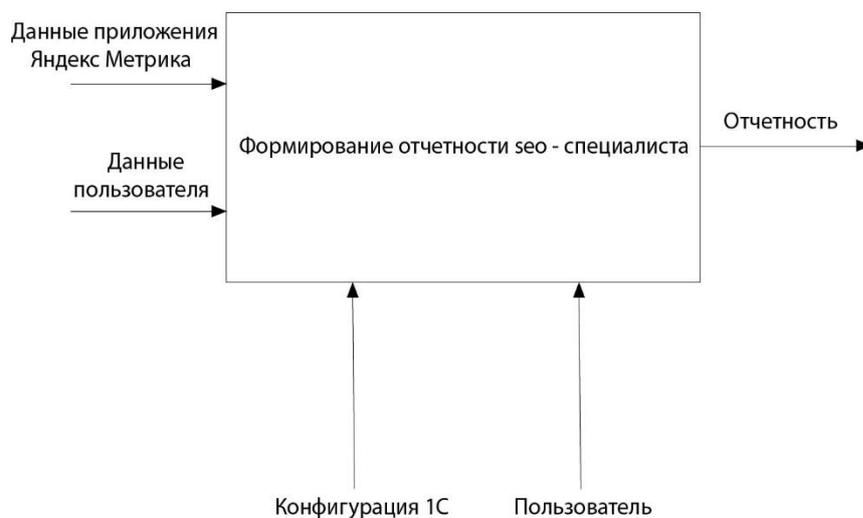


Рисунок 8 – Контекстная диаграмма.

Выводы к главе 2

В данной главе рассматривалась разработка и описание архитектуры и принципы функционирования проектируемой системы. Создание нового документа в проекте, структура конфигурации отчетности. Была разработана логическая модель преобразования данных и составлена контекстная диаграмма.

Глава 3 Разработка компонентов информационной системы

3.1 Разработка компонента «Справочники»

Общий вид данного компонента представлен на рисунке 8.

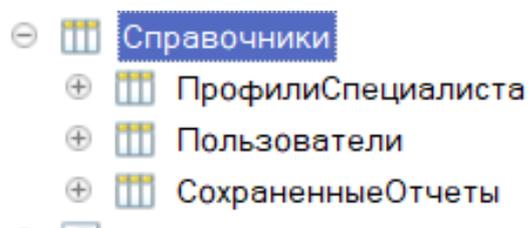


Рисунок 9 – Структура элемента «Справочники»

1. Справочник «Профили специалиста»

Данный справочник содержит информацию о данных метрики для работы с API. Реквизиты справочника представлены в таблице 1.

Таблица 1 - Описание элементов справочника «Профили специалиста»

Имя поля	Описание	Тип данных
1	2	3
Код	Обязательный реквизит, является уникальным	
Пользователь	Пользователь	Справочник.Пользователи
Токен метрики	Токен метрики	Строка
ID Приложения	ID Приложения	Строка
Счетчик	Строка счетчика	Строка

2. Справочник «Пользователи»

Данный справочник включает в себя список всех сотрудников работающих в компании.

Таблица 2 - Описание элементов справочника «Сотрудники»

Имя поля	Описание	Тип данных
1	2	3
Код	Обязательный реквизит, является уникальным	
Наименование	ФИО	Строка

3. Справочник «Сохраненные отчеты»

Данный справочник включает в себя все сохранённые отчеты.

Структура данного справочника представлена в таблице 3.

Таблица 3 - Описание элементов справочника «Специальность»

Имя поля	Описание	Тип данных
1	2	3
Код	Обязательный реквизит, является уникальным	
Наименование	Наименование	Строка
Хранилище отчета	Хранилище отчета	Хранилище значений

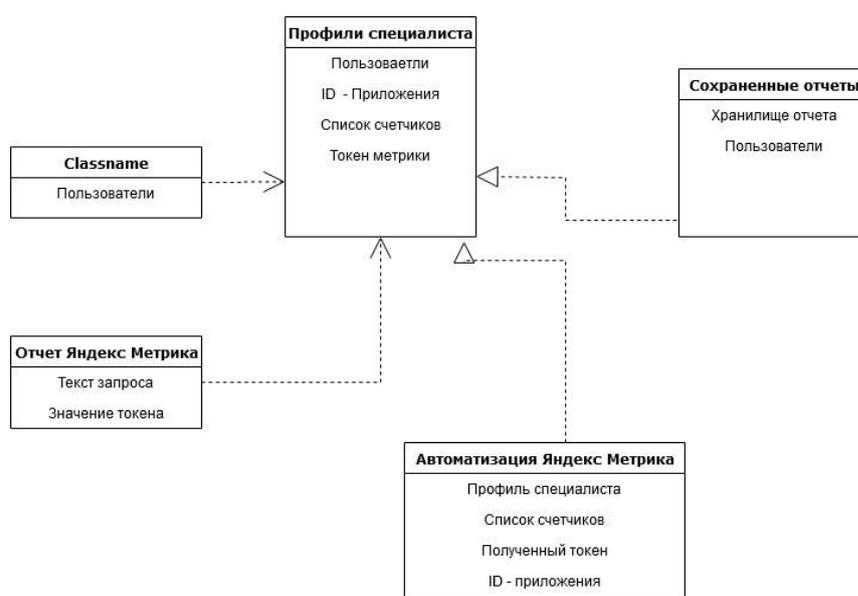


Рисунок 10 - Описание объектов конфигурации

3.4 Разработка пользовательского интерфейса системы

Подсистемы – это общие объекты конфигурации. Подсистемы состоят из объектов конфигурации и объединяют их в блоки.

Пользовательский интерфейс главного окна автоматизированной системы представлен на рисунке 11.

В разработке основной модуль системы состоит из двух подсистем:

- Справочники;
- Обработки;

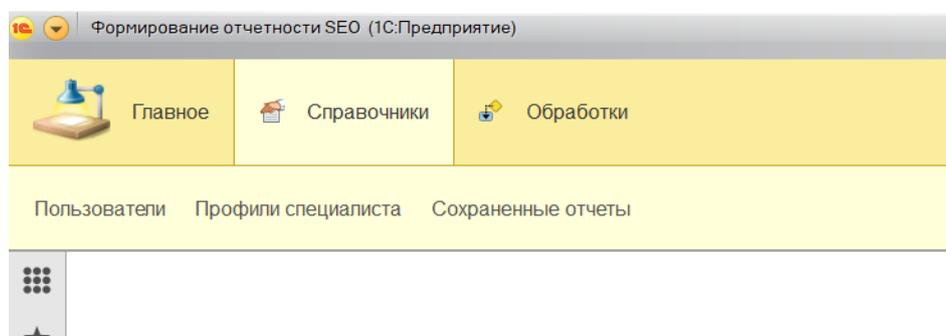


Рисунок 11 – Подсистемы конфигурации

В процессе работы пользователем выбирается одна из систем. В подсистеме «Справочники» находятся справочники, в которых находятся справочники для ввода информации о сотрудниках, данных метрики, счетчиков, токены и хранятся сохраненные отчеты. В подсистеме «Обработки» находится обработка, которая занимается заполнением данных элемента справочника «Профили специалиста» и отчет «Отчет Яндекс Метрики» обращается к сервису данных и формирует отчет на их основе.

Для добавления справочника отрываем конфигуратор и в вкладке «Справочники» добавляем новый элемент. Указываем имя «Пользователи». На вкладке «ФИО» добавляем новый реквизит с наименованием «Описание» с типом «Строка». Длин строки неограниченная. Состав данных изображен на рисунке 12.

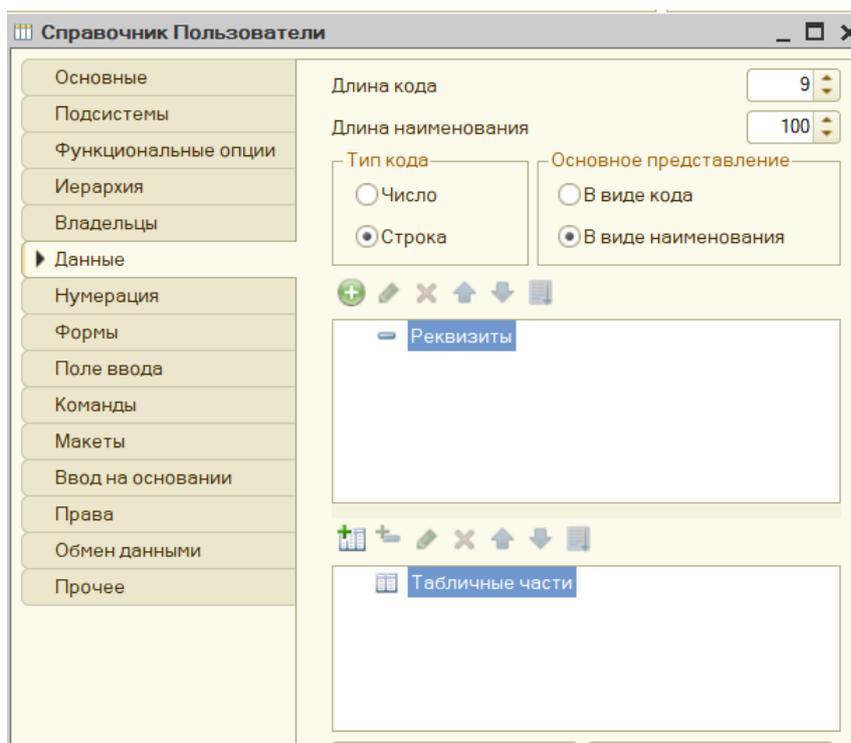


Рисунок 12 – Добавления элемента в справочник

Реквизит «Наименование» переименован в «Ф.И.О». Для переименование стандартного реквизита в конфигураторе необходимо открыть стандартные реквизиты справочника (рисунок 13).

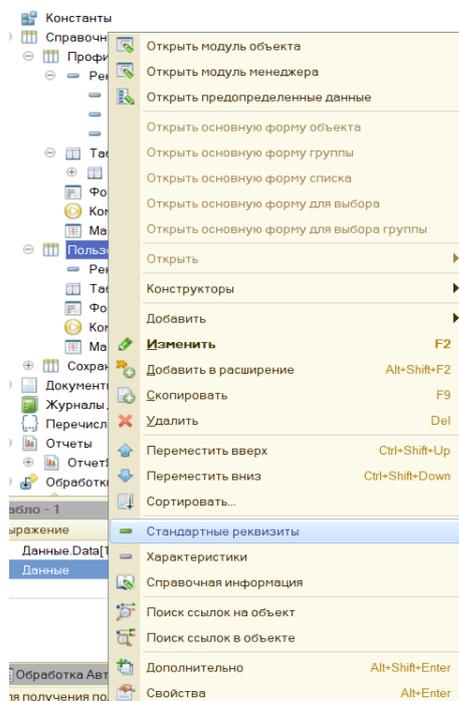


Рисунок 13 – Стандартные реквизиты справочника

И в свойствах реквизита «Наименование» изменить синоним(рисунок 14). Длина реквизита увеличена до 100.

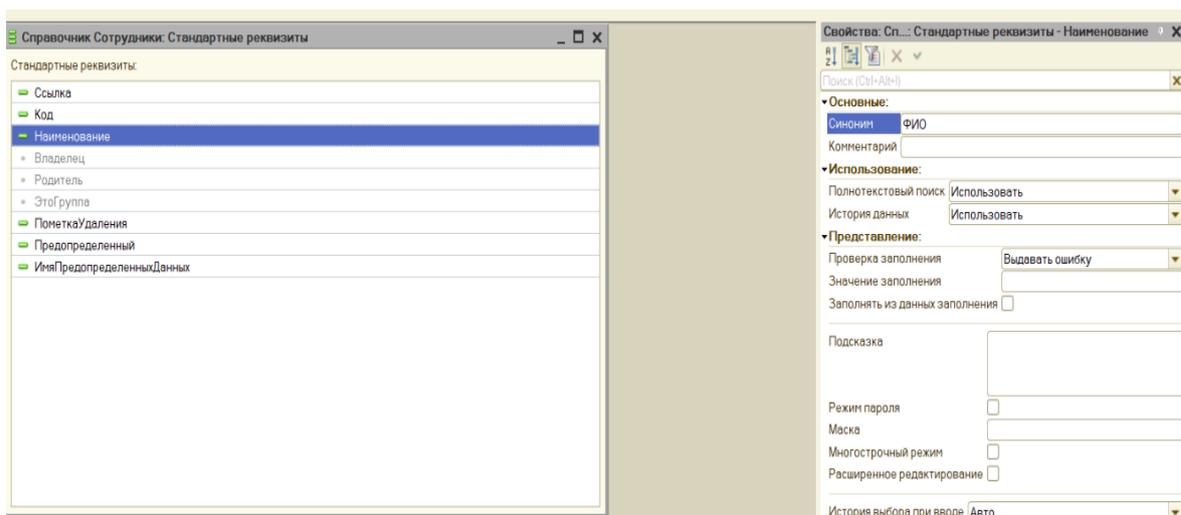


Рисунок 14 – Наименование реквизитов.

Справочник будет иметь стандартную форму, так как у него два реквизита. Справочник «Профили специалиста» используется для ввода данных авторизации в API. Справочник будет иметь реквизит «Пользователь» с типом «СправочникСсылка.Пользователи», «ТокенМетрики» с типом «Строка», «IDПриложения» с типом «Строка», «Счетчик» работы с типом «Строка». Форма справочника изображена на рисунке 15;

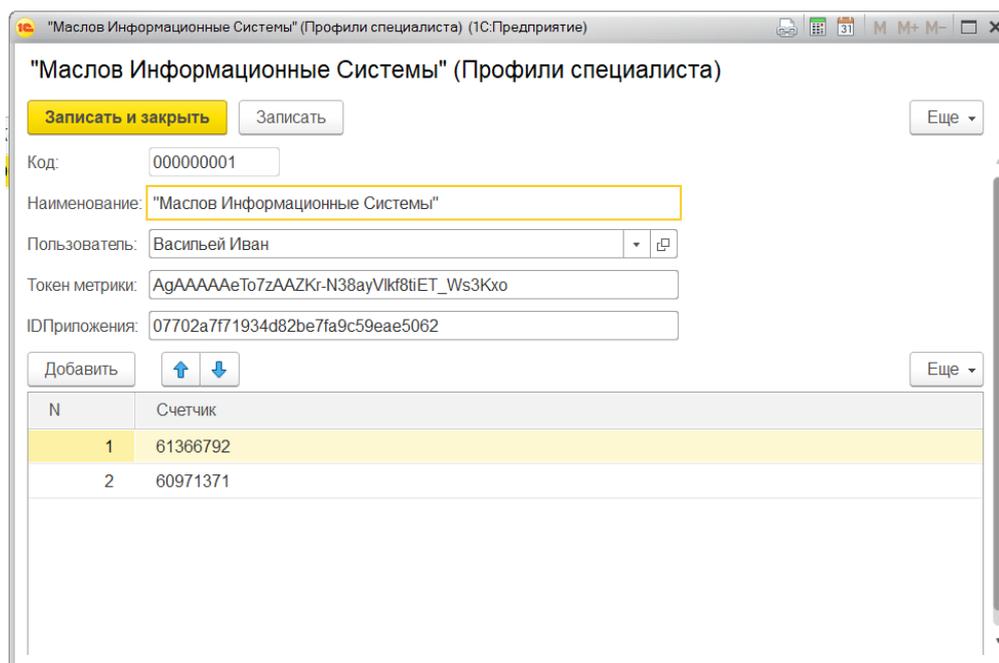


Рисунок 15 – Профиль специалиста

Справочник «Сохраненные отчеты» используется для сохранения и отображения сформированных ранее отчетов. Справочник будет иметь реквизит «Пользователь» с типом «СправочникСсылка.Пользователи», «Хранилище отчета» с типом «хранилище значений». Реквизит типа «хранилище значений» позволяет хранить в них данные. Он может быть записан в поля базы данных, имеющие соответствующий тип. Это позволяет сохранять в базе данных значения, тип которых не может быть выбран в качестве типа поля, например. Форма справочника - на рисунке 16.

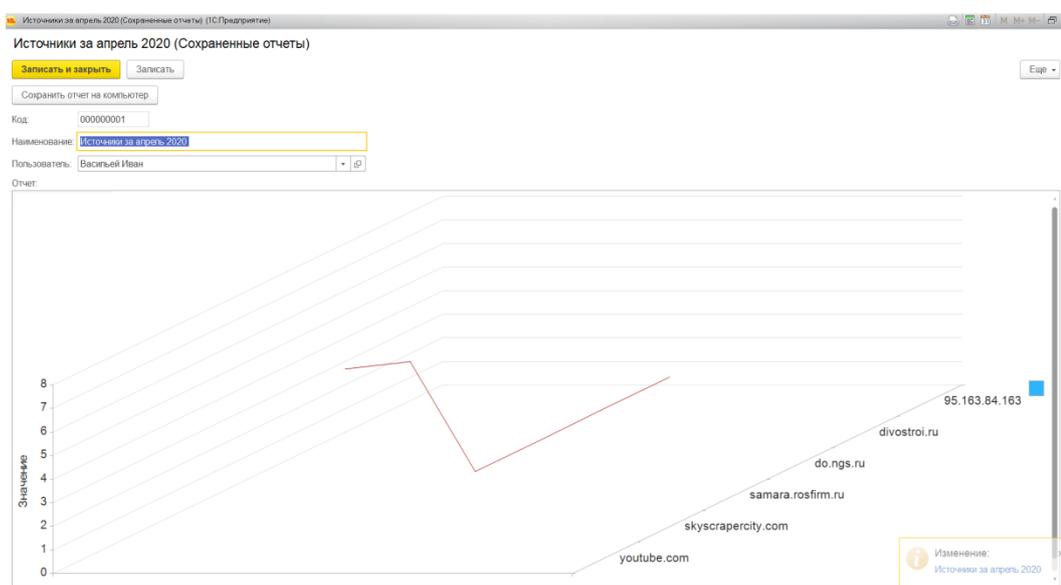


Рисунок 16 – Диаграмма обратных ссылок

Для отображения отчетов код записывался в обработчик (рисунок 17.) «ПриСозданииНаСервере», который вызывается при открытии элемента. В этом же участке есть код, который используется при сохранении отчетов. В параметрах создания передается сформированный отчет и процедура «Свойство()» выполняет проверку существования значения и заполняет его. Для записи значения в хранилище значений необходимо преобразовать данные формы в данные объекта, заполнить его и записать. Далее получается значение хранилища и если оно заполнено – выводится на табличное поле формы.

&НаСервере

```
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    ДанныеОтчета = Неопределено;
    Если Параметры.Свойство("ДанныеОтчета", ДанныеОтчета) Тогда
        ЭтотОбъектДанные = РеквизитФормыВЗначение("Объект");
        ЭтотОбъектДанные.ХранилищеОтчета = ДанныеОтчета;
        ЭтотОбъектДанные.Записать();
        ЗначениеВРеквизитФормы(ЭтотОбъектДанные, "Объект");
    КонецЕсли;

    ЭтотОбъектДанные = РеквизитФормыВЗначение("Объект");
    ТабДок=ЭтотОбъектДанные.ХранилищеОтчета.Получить();
    Если ТабДок<>Неопределено Тогда
        ЭтаФорма.Отчет.Вывести(ТабДок);
    КонецЕсли;
КонецПроцедуры
```

Рисунок 17 – Отображения отчетов.

Опишем процедуру сохранения отчета на память компьютера (рисунок 18). Чтобы вызвать окно выбора пути в 1С имеется встроенный конструктор «Диалог выбора файла» в которой нужно передать параметр «Выбор каталога». В параметрах диалога передаётся описание оповещения, которое вызывает процедуру указанную в параметрах поле закрытия формы указания пути. После выбора выполняется проверка на указание пути и отчет записывается с помощью процедуры «Записать()».

```
«НаКлиенте
| Процедура СохранитьОтчетНаКомпьютер(Команда)
    СтандартнаяОбработка = Ложь;
    ДиалогВыбораФайла = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.ВыборКаталога);
    ДиалогВыбораФайла.Заголовок = "Укажите место хранения файла";

    ДиалогВыбораФайла.Показать(Новый ОписаниеОповещения("ПутьКФайлуСохранениеЗавершение", ЭтотОбъект, Новый Структура("ДиалогВыбораФайла", ДиалогВыбораФайла)));
. КонецПроцедуры

«НаКлиенте
| Процедура ПутьКФайлуСохранениеЗавершение(ВыбранныеФайлы, ДополнительныеПараметры) Экспорт
    Если ВыбранныеФайлы <> Неопределено Тогда
        Адрес = ВыбранныеФайлы[0] + "\" + Объект.Наименование+ ".xls";
        ЗаписатьНаСервере(Адрес);
    КонецЕсли;
. КонецПроцедуры

«НаСервере
| Процедура ЗаписатьНаСервере(Адрес)
    Отчет.Записать(Адрес, ТипФайлаТабличногоДокумента.XLS);
. КонецПроцедуры
```

Рисунок 18 – Сохранение отчета на память компьютера

3.5 Разработка обработки «Авторизация Яндекс Метрика»

Данная обработка предназначена для быстрого и удобного заполнения профиля пользователя.

В основные функции входит:

- Заполнение списка счетчиков
- Заполнение ID Приложения
- Заполнения токена

Чтобы добавить на форму поле HTML-страницы необходимо создать реквизит с типом «Строка», добавить его на форме и изменить его вид на «Поле HTML документа». (рисунок 19).

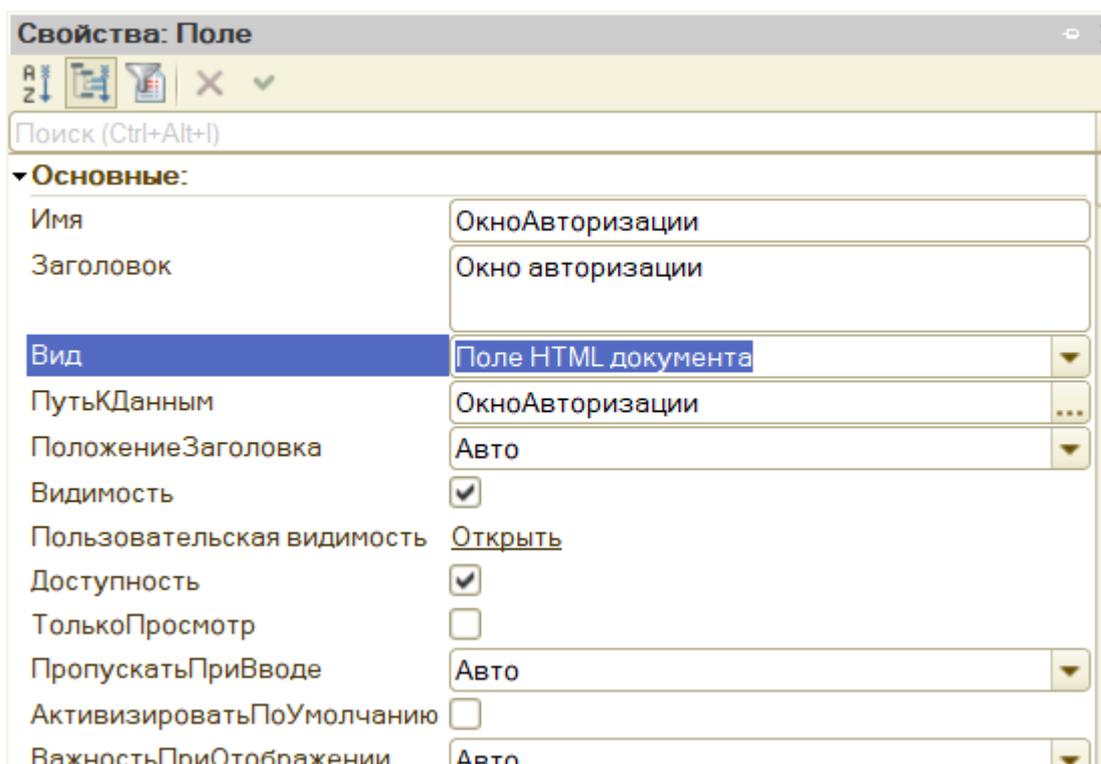


Рисунок 19 - Поле HTML документа.

Для того, чтобы перейти на какую-либо страницы необходимо значению реквизиты присвоить необходимой адрес. Например: «ОкноАвторизации = "oauth.yandex.ru";».

Добавлены команды перехода на страницы:

1. Открыть окно списка приложений, адрес - `oauth.yandex.ru`. Для перехода нужно заполнить Профиль Специалиста, сделана проверка и вывод сообщения. (рисунок 20).

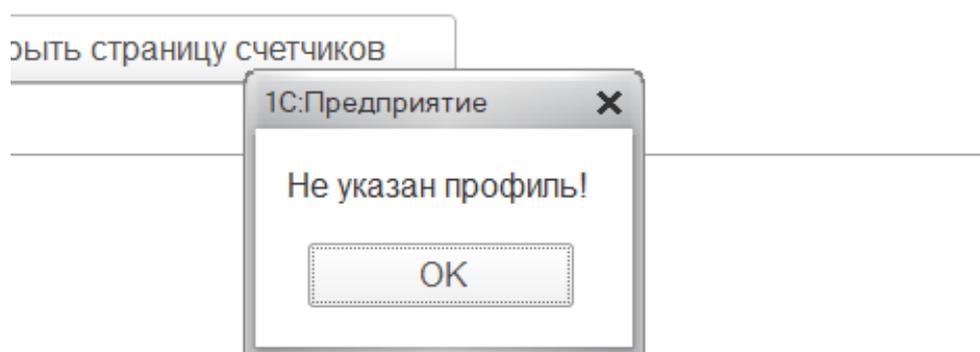


Рисунок 20 – Окно предупреждения

2. Открыть окно получения токена. Данные адрес собирается из адреса токена с указанием id приложения, которое вводится прежде. И имеются проверка на заполнения профиля и приложения.

3. Открыть страницу счетчиков, адрес - `oauth.yandex.ru` (рисунок 21).

```
&НаКлиенте
□ Процедура ОткрытьОкноСпискаПриложений (Команда)
  Если ЗначениеЗаполнено (ПрофильСпециалиста) Тогда
    ОкноАвторизации = "oauth.yandex.ru"
  Иначе
    ПоказатьПредупреждение (, "Не указан профиль!");
  КонецЕсли;
КонецПроцедуры
```

Рисунок 21 – Авторизация через службу Oauth

Поле окна авторизации сделано свертываемым для упрощения

отображения. Чтобы сделать окно свертываемым необходимо добавить его в группу и сменить поведение группы на «свертываемая». (рисунок 22).

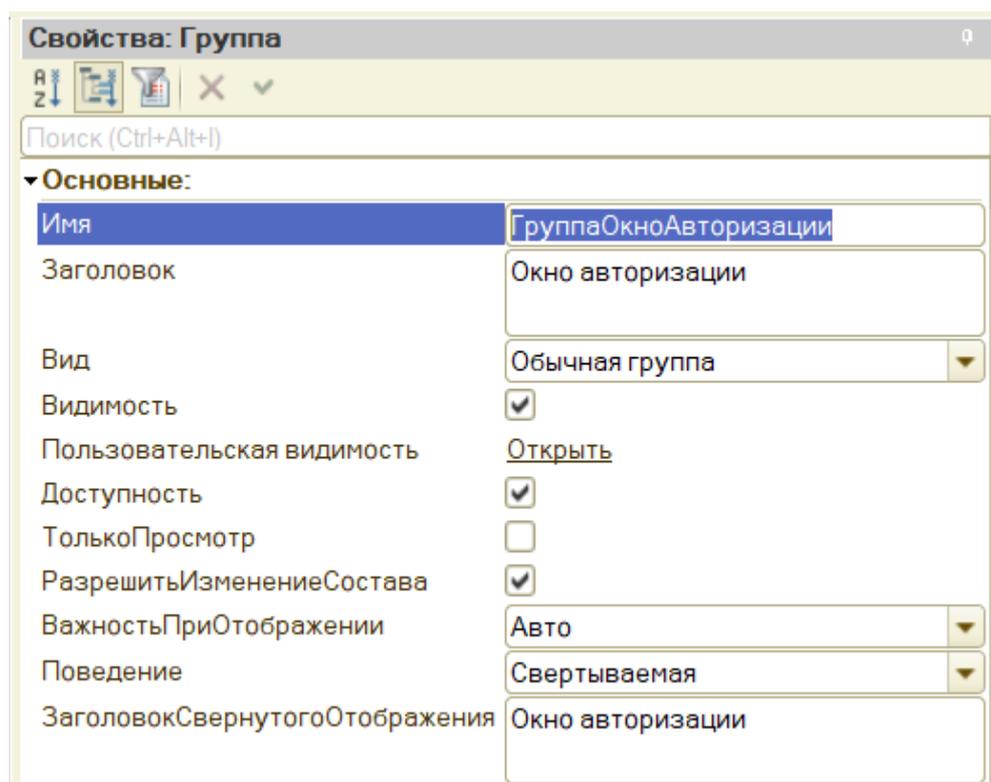


Рисунок 22 – Настройка окна авторизации.

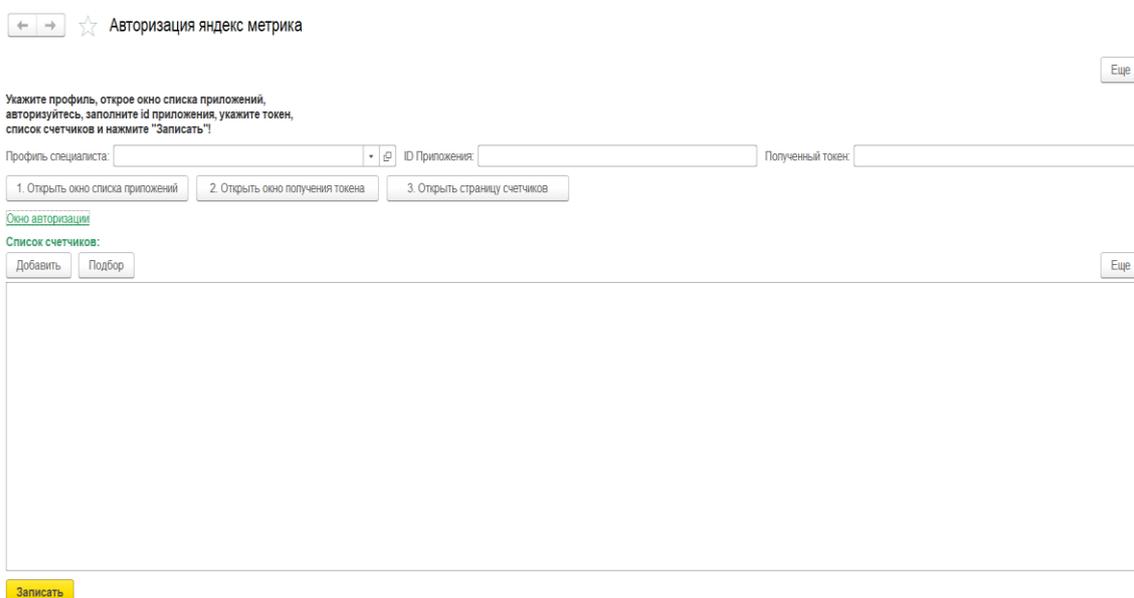


Рисунок 23 - Форма обработки на изображении

3.6 Разработка компонента «Отчет Яндекс метрики»

Далее разработаем отчет, который будет заниматься отображением и получением данных API Яндекс метрики. Для начала разберём работу API.

API отчетов позволяет получать информацию о статистике посещений сайта и другие данные, не используя интерфейс Яндекс.Метрики.

При формировании запроса к API используются группировки (dimensions) и метрики (metrics).

Группировка (dimension) — это атрибут визита или хита, по которому можно сгруппировать данные.

В запросах к API группировки задаются в параметре `dimensions`. Если необходимо указать несколько группировок, перечислите их через запятую.

Также возможен отчет без группировок, в этом случае будет рассчитан суммарный результат.

Метрика (metric) — числовая величина, которая рассчитывается на основе атрибута хита или визита. В запросах к API метрики задаются в параметре `metrics`. Метрики и группировки, указанные в запросе к API, позволяют формировать необходимую нам структуру отчета. Для указания периода используются параметры «date1» и «date2» в формате "гггг-мм-дд".

Группировки и метрики имеют два типа:

- `ym:s:`
- `ym:pv:`

Одновременное использование обоих типов запрещено. Создадим отчет, для него форму и на форме расположим команду открытия формы в которой будет составляться текст запроса. Для открытия формы используем процедуру «ОткрытьФорму», в параметры передаем описание оповещения, которое будет вызывать процедуру после закрытия формы с настройками. Процедура после закрытия формы настроек заполняет

значениями реквизиты объекта.

Добавим команду сохранения отчета на командную панель отчета (рисунок 24).

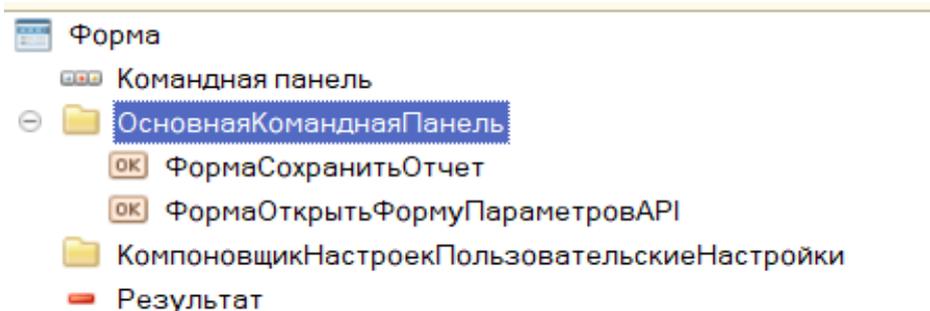


Рисунок 24 – Командная панель отчета

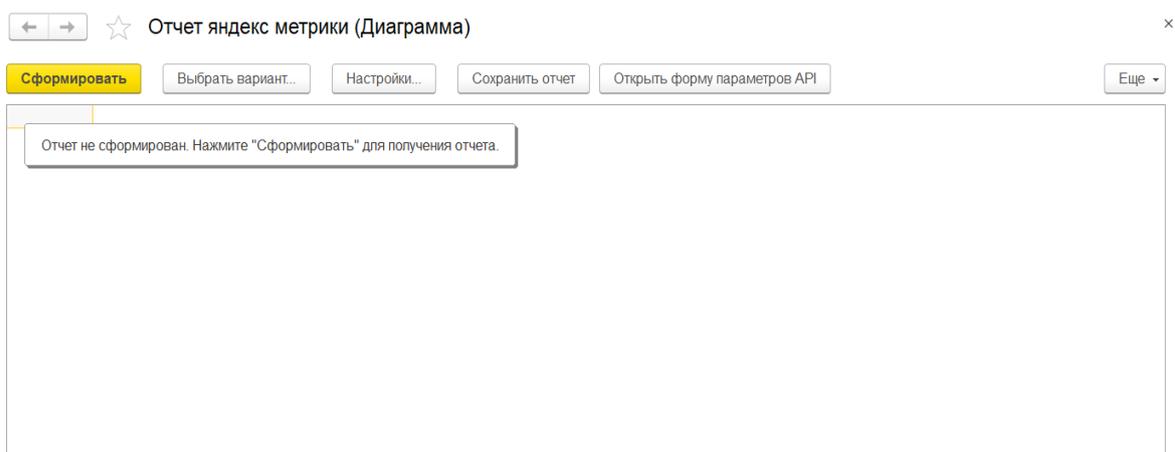


Рисунок 25 – Форма отчета на изображении

Форма настроек разделена на выбор метрик и группировок. При выборе одного типа метрики или группировки элементы одной группы автоматически блокируются. Для этого элементы разнесены по группам, ставя параметр группы «Доступность» в значение Ложь – все элементы входящие в эту группу наследуют это значение. На форме расположены поля выбора профиля типом «СправочникСсылка.ПрофилиСпециалиста» и поле выбора счетчика, которое будет заполняться при выборе профиля. Чтобы создать поле выбора создается реквизит с типом «Строка» и в

свойствах реквизита устанавливается значение

«РежимВыбораИзСписка»(рисунок26).



Рисунок 26 – Режим выбора из списка

Для заполнения списка выбора у элемента поля имеется свойство «список выбора». При изменении профиля в цикле это свойство будет заполняться или очищаться, если значение профиля будет пустое. Заполнение списка выбора выполняется на сервере, потому что данные счетчика хранятся в табличной части, а данные табличных частей доступны только там.

При нажатии «Принять» выполняется проверка на заполнение поля профиль, счетчика и метрик и выбор хоть одной метрики. Если не указывать группировки в таком случае мы получим суммарное значение за период. Для составления строки выбор метрик выполняется проверка на выбор значения и к пустому текстовому полю прибавляется значение. Для указания метрик они указываются через в параметр «&metrics=». Чтобы в родительскую форму передать значение в параметр закрытия передается массив с текстом запроса и токеном. Токен передается из профиля специалиста. Так же текст запроса дополняется выбранным счетчиком метрики в параметр «&id». Добавим функцию сохранения и восстановления значений профиля, метрик и периода. Для этого в параметрах формах установлено значение «АвтоматическоеСохранениеДанныхВНастройках». После этого в списке реквизитов напротив нужных реквизитов устанавливается значение в колонке «Сохранение» (рисунок 27).



Рисунок 27 – Колонка сохранения

После этого на форме автоматически появятся команды сохранения и восстановления настроек. Полученная форма настроек на рисунке 28.

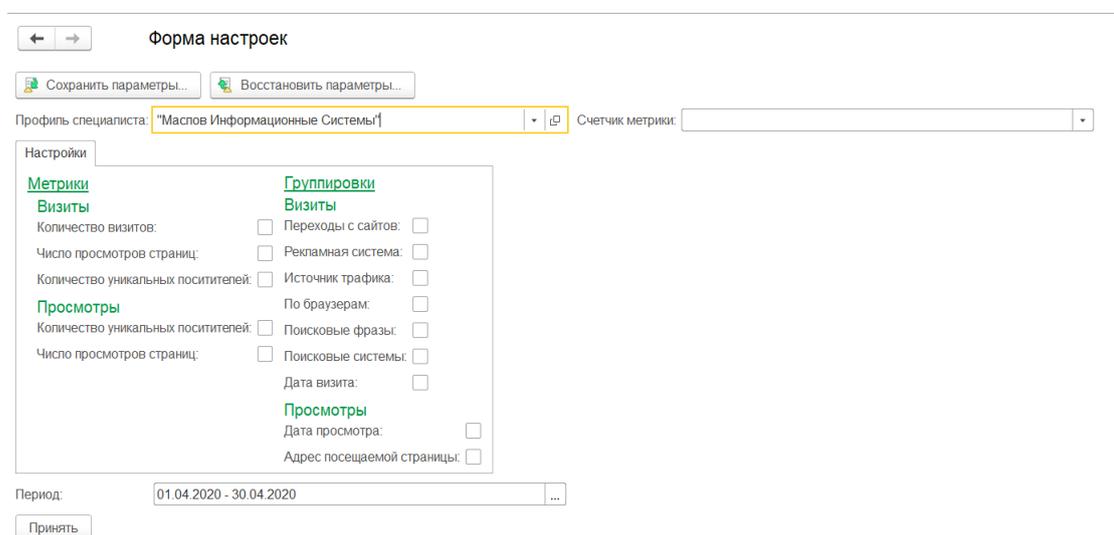


Рисунок 28 – Форма настроек

После заполнения настроек полученный результат записывается в реквизиты объекта. Отчеты в 1с строятся на основе схемы компоновки данных. Она содержит в себе набор параметров и структуру данных их источник. Так как у нас данные будут получаться через api, а не с базы 1с создаётся набор данных "Объект". В него добавляются поля «значение», «метрика». Поле «метрика» будет отображать значение группировки, «значение» - значение группировки (рисунок 29).

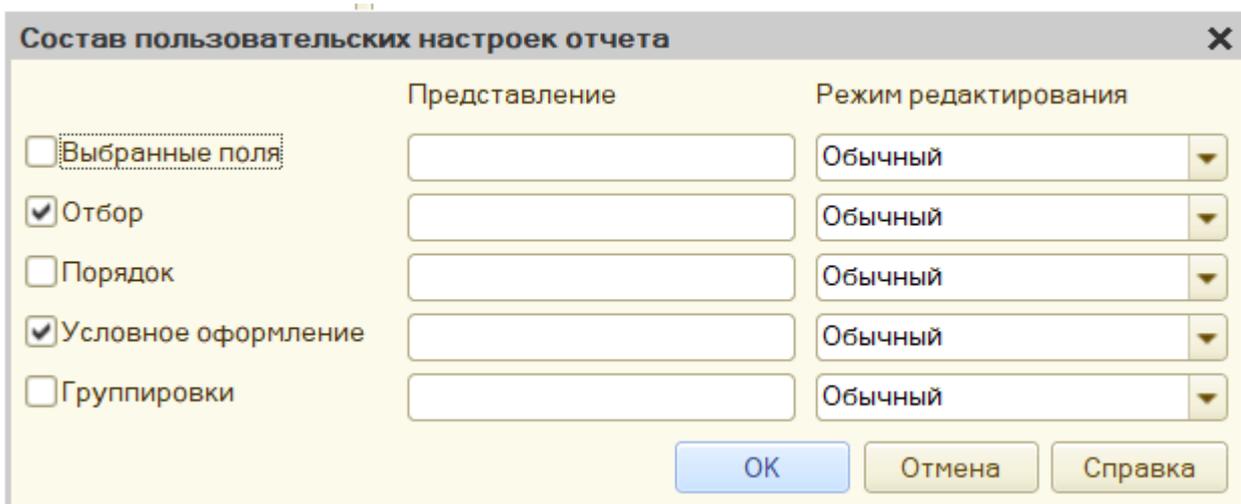


Рисунок 32 – Фильтр свойств

Опишем процедуру формирования отчета. Для этого в модуле объекта в процедуре «ПриКомпоновкеРезультата» пишется код. Отключается стандартная обработка, чтобы описать свою процедуру вывода отчета. Выполняются проверки на заполнение токена и текста запроса (рисунок 33).

```

Процедура ПриКомпоновкеРезультата (ДокументРезультат, ДанныеРасшифровки, СтандартнаяОбработка)
СтандартнаяОбработка = Ложь;
Если Не ЗначениеЗаполнено (ЭтотОбъект.ТекстЗапроса) Тогда
    Сообщить ("Текст запроса пустой!");
    Возврат
КонецЕсли;

Если Не ЗначениеЗаполнено (ЭтотОбъект.ЗначениеТокена) Тогда
    Сообщить ("Не указан токен OAuth!");
    Возврат
КонецЕсли;

```

Рисунок 33 – Процедура формирование отчета

Сервис API Яндекс метрики поддерживает только защищенные запросы https. SSL Сертификаты берутся из операционной системы. Создаём соединение с сервером "api-metrika.yandex.net" по защищенному соединению и отправляем созданный ранее запрос. В заголовки запроса вставляются значения «Content-Type», "application/x-yametrika+json" и "Authorization", "OAuth" с значением токена.

На сайте «https://yandex.ru/dev/metrika/doc/api2/api_v1/concept/errors-dospage/» есть список ошибок в получаемом ответе. Код состояния = «200»

означает, что запрос выполнен корректно и данные получены, в ином случае выводится сообщение с полученным кодом состояния. Ответ приходит в типе «json», после чтения результатом будет массив структур. Значение данных хранится в массиве «data». Если он будет пуст – значит данных за этот период нет. Если строка одна – значит это сумма без группировок, оно будет выводиться в сообщении.

В ином случае преобразуем массив структур в таблицу значений т.к. из массива отчет не построить. Создаём таблицу значений с колонками «Значение», «Метрика» и в цикле заполняем её (рисунок 34).

```
функция ПреобразоватьМассивСтруктурВТаблицуЗначений(Массив, Метрики)
ТЗ = Новый ТаблицаЗначений;
ТЗ.Колонки.Добавить("Метрика");
ТЗ.Колонки.Добавить("Значение");
Счетчик = 0;
КоличествоМетрик = Метрики.Количество();
Для Каждого Элемент Из Массив Цикл
    Пока Счетчик < КоличествоМетрик Цикл
        НоваяСтрока = ТЗ.Добавить();
        Если КоличествоМетрик > 1 Тогда
            НоваяСтрока.Метрика = Элемент.dimensions[0].Name + " " + Прав(Метрики[Счетчик], СтрДлина(Метрики[Счетчик]) - СтрНайти(Метрики[Счетчик],
        Иначе
            НоваяСтрока.Метрика = Элемент.dimensions[0].Name;
        КонечЕсли;
        НоваяСтрока.Значение = Элемент.metrics[Счетчик];
        Счетчик = Счетчик+1;
    КонечЦикла;
Счетчик = 0;
КонечЦикла;
Возврат ТЗ
КонечФункции
```

Рисунок 34 – Таблица значений

Полученную таблицу значений передаётся в компоновщик формирования отчета. Чтобы передать таблицу значений в компоновщик, она помещается в структуру и передаётся в параметрах инициализации процессора компоновки данных.

3.7 Тестирование приложения

3.7.1 Тестирование суммы метрики за период

В настройках выбран профиль специалиста, счетчик и метрику «Количество визитов». В результате получаем сообщение с суммы значения. (рисунок 35).

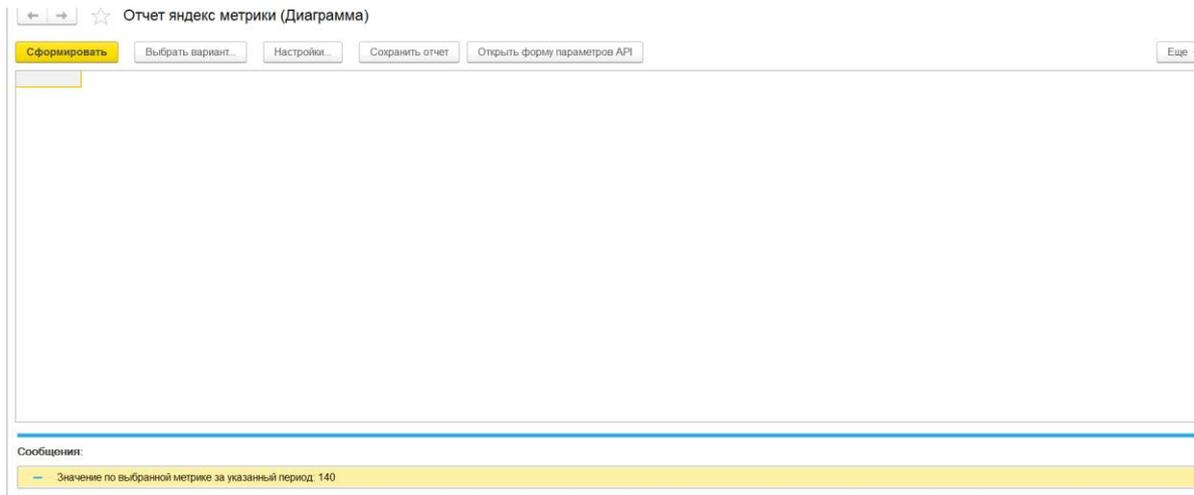


Рисунок 35 – Количество визитов

3.7.2 Тестирование метрики по группировке

В настройках выбран профиль специалиста, счетчик, метрику «Число просмотров страниц» и группировку по дате просмотра. В результате получаем отчет. (рисунок 36).

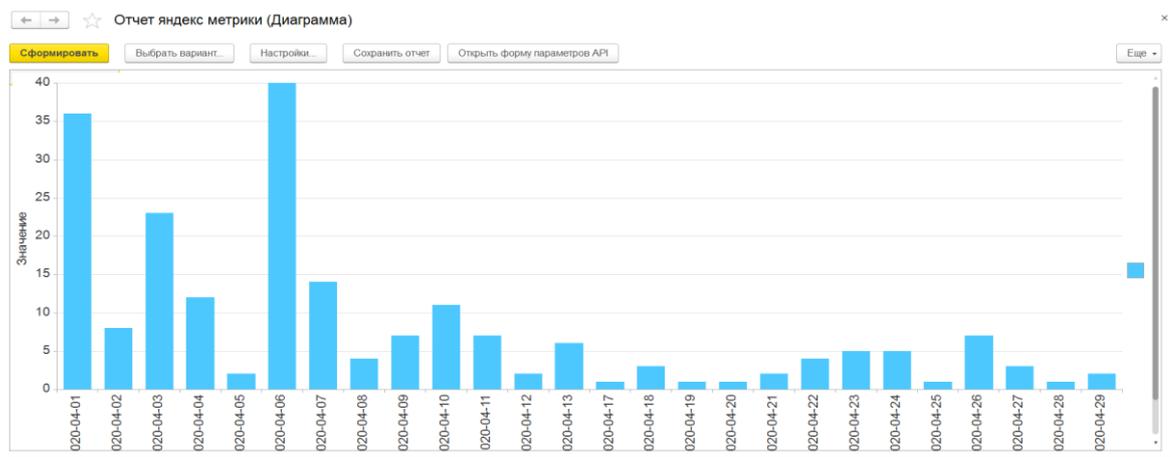


Рисунок 36 – Диаграмма количество просмотров

Так же выбираем вариант «Таблица» (рисунок 37).

← → ☆ Отчет яндекс метрики (Таблица) x

Сформировать | Выбрать вариант... | Настройки... | Сохранить отчет | Открыть форму параметров API | Еще ▾

Значение	2020-04-01	2020-04-02	2020-04-03	2020-04-04	2020-04-05	2020-04-06	2020-04-07	2020-04-08	2020-04-09	2020-04-10	2020-04-11	2020-04-12	2020-04-13	2020-04-14
Значение	36	8	23	12	2	40	14	4	7	11	7	2	6	6
Итого	36	8	23	12	2	40	14	4	7	11	7	2	6	6

Рисунок 38 – Выбор таблицы

3.7.3 Тестирование сохранения

Нажимаем команду «Сохранить отчет», в открывшемся окне заполняем имя и пользователя. (рисунок 39).

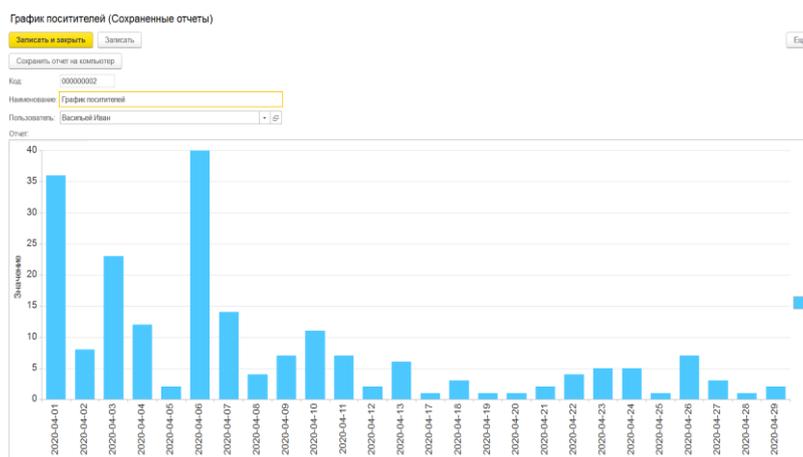


Рисунок 39 – Сохранение отчета

Выводы к главе 3

В третьей главе показана разработка компонентов «Справочники» и его структура, разработка пользовательского интерфейса системы, сохранения отчетов, диаграмма обратных ссылок. Разработка обработки «Авторизации Яндекс Метрика» настройка окна авторизации через службу OAuth. Разработка компонента «Отчет Яндекс метрика». Было произведено тестирование приложения, тестирование суммы метрики за период, тестирование метрики по группировке, тестирование сохранения.

Глава 4 Экономическая часть

4.1 Характеристики результатов разработки

Данная выпускная квалификационная работа посвящена разработке программного решения для автоматизации формирования отчета seo-специалиста.

Реализация данного проекта позволяет:

- Получать данные аналитик Яндекс метрик.
- Сохранение готовых отчетов в базу 1с.

Для оценки эффективности полученного в результате реализации разработки программного продукта используем два критерия: удобство и реализация полного цикла бизнес процесса формирования отчета.

Для расчета экономического эффекта от разработки необходимо определить прогнозируемую прибыль от внедрения разработки в работу seo-специалистов. Разработка данного пособия, так же как и разработка любого другого программного продукта, требует определенных материальных, временных и трудовых затрат, а следовательно должна соответственно окупаться. С экономической точки зрения затраты связанные с выполнением проекта, должны быть покрыты доходами от реализации конечного продукта.

Трудоемкость выполнения программного продукта — характеризуется перечнем основных этапов и видов работ, которые должны быть выполнены. Упорядочен данный перечень в соответствии со смысловым содержанием каждого вида работ и взаимосвязями между всеми видами работ.

Форма расположения работ по этапам:

1.Разработка технического задания (ТЗ):

- получение ТЗ.

2.Подготовительный этап:

- сбор информации;
- выбор объектного построения программы;
- разработка общей методики создания продукта.

3. Основной этап:

- разработка основного алгоритма;
- создание интерфейса;
- отладка.

4. Завершающий этап:

- подготовка технической документации;
- сдача продукта.

Трудоемкость выполнения работы оценивается в человеко-часах и носит вероятностный характер.

Таблица 4 – Расчет трудоёмкости по проекту

Виды работ	Трудоёмкость
Получение ТЗ	2
Сбор информации	20
Выбор объектного построения программы	10
Разработка основного алгоритма	50
Создание интерфейса	24
Написание кода	20
Отладка	12
Подготовка технической документации	14
Сдача продукта	1
Итого	153

На основе данных о трудоемкости рассчитаем сумму потраченную на зарплату программисту. Предположим, зарплата среднего программиста 20000 тыс. руб., при 8 часовом рабочем дне 108,65 руб./час. Итого $217,3 \cdot 153 = 16623,45$ рублей. Отчисления с зп составляют около 30%, поэтому 21610, рублей сумма потраченная на зарплату программисту. Стоимость компьютера 30 тыс. руб. Используем его в течение 5 лет (62 мес.). Годовая амортизация составит в месяц 483,8 руб. В час: 0, 67 руб. Умножив на

трудоемкость, определим: $AO = 0,67 * 153 = 102,51$ руб. ПК в среднем потребляет 0,6 кВт.ч. $0,6 * 153 = 91,8$ кВт. Примерная стоимость кВт.ч. около 1,97 руб. Итого: $105 * 1,97 = 180,84$ руб.

Таблица 5–
Смета всех
затрат

Наименование статей затрат	Сумма, руб.
1. Расходные материалы	350
2. Основная заработная плата	21 610,4
3. Расходы на электроэнергию	91,8
4. Амортизационные отчисления	102,51
5. Отчисления на ЗП	9 974,07
Итого	32 128,78

Таким образом, суммарная стоимость затрат на создание данного программного продукта составляет 32 128,78 рублей, следовательно эту программу ниже этой цены продать нельзя так, как в этом случае программа не будет являться экономически эффективной.

В среднем seo-специалист тратит 2 часа на формирование и администрирование отчетов. Исходя из средней зарплаты 30 000 рублей, 140 рабочих часов и 20 дней, получаем $20 * 2 * (30\ 000 / 140) = 8\ 560$.

В результате внедрения конфигурации удалось сократить это время до $20 * (30\ 000 / 140) = 4\ 280$;

$8\ 560 - 4\ 280 = 4\ 280$ - месячная экономия одного сотрудника

$32\ 128 / 4\ 280 = 7.5$ – время окупаемости при одном сотруднике в компании.

Таким образом, внедрении конфигурации позволило сократить до часа время формирования отчетности и полную окупаемость в течении 7.5 месяцев при одном сотруднике.

Выводы к главе 4

Четвертая глава посвящена экономической целесообразности разработки данного ПО. Были рассмотрены характеристики результатов разработки, расчет трудоемкости проекта, сумма всех затрат

. Заключение

На основе нашего исследования, можно сделать выводы, что платформа 1С: Предприятие является многофункциональным продуктом и не требует больших познаний в области программирования. Имея данную программу, мы можем автоматизировать практически любой бизнес-процесс. В работе seo-специалиста важна актуальность информации, удобство получения различной информации и интеграции. В нашей разработке реализована работа с различными приложениями и рекламными компаниями. Возможна выборка данных по различным метрикам, группировкам и периоду или получение итогового значения. Реализовано сохранение настроек API, полученных отчетов в базу 1С или на компьютер. Так же есть выбор вариантов отображения и возможность пользовательского форматирования.

Для оценки эффективности полученного в результате реализации разработки программного продукта использованы два критерия: удобство и реализация полного цикла бизнес процесса формирования отчета.

Для расчета экономического эффекта от разработки определена прогнозируемую прибыль от внедрения разработки в работу seo-специалистов.

Внедрении конфигурации позволило сократить до часа время формирования отчетности и полную окупаемость в течении 7.5 месяцев при одном сотруднике.

Таким образом, применение нашей конфигурации упрощает взаимодействие seo-специалиста, что позволило сократить время получения отчетности с двух часов в день до одного и экономической окупаемости за семь с половиной месяцев при одном работающем в компании сотруднике.

Список используемой литературы

1. Администрирование 1С 8.3 с нуля для начинающих. [Электронный ресурс]. URL: <https://programmist1s.ru/administrirovanie-1s/> (дата обращения: 01.06.2020).
2. Ажеронок В.А. Разработка управляемого интерфейса. 2018. 36 с.
3. Бояркин В.Э. 1С:Предприятие 8. Конвертация данных: обмен данными между прикладными решениями. 2018. 117 с.
4. Выгрузка данных из 1С 8.3 в Эксель. [Электронный ресурс] URL: <https://programmist1s.ru/vyigruzka-dannyih-iz-1s/> (дата обращения: 01.06.2020).
5. Выгрузка данных из 1С 8.3 в Эксель. [Электронный ресурс]. URL: <https://programmist1s.ru/vyigruzka-dannyih-iz-1s/> (дата обращения: 01.06.2020).
6. Козырев, Д. В. Профессиональная разработка в системе 1С:Предприятие 8 / Д. В. Козырев, Москва: 1С-Учебный центр №3, 2009. - 103 с.
7. Митичкин, С. А. Разработка в системе 1С:Предприятие 8.0 / С. А. Митичкин. Москва: 1С-Паблишинг. 2003. 416 с.
8. Описание работы API Яндекс метрика [Электронный ресурс]. URL: <https://yandex.ru/dev/metrika> (дата обращения: 01.06.2020).
9. Парсер JSON (Штатные средства 1С, 8.3) [Электронный ресурс]. URL: <https://infostart.ru/public/551972/> (дата обращения: 01.06.2020).
10. Постовалов С. 1С: Предприятие 8. Зарплата и Управление Персоналом. Фирменные рецепты внедрения, БХВ-Петербург. 2018. 304 с.
11. Реализация прикладных задач в системе «1С: Предприятие 8.2» / А. П. Габец [и др.]. Москва: 1С-Паблишинг, 2010. 714 с.
12. Таблица значений в языке 1С 8.3 в примерах [Электронный ресурс]. URL: <https://helpme1c.ru/tablica-znachenij-v-yazyke-1s-8-v-primerax> (дата обращения: 01.06.2020).
13. Учебник по 1С, база знаний, форум [Электронный ресурс]. URL: <http://www.mista.ru> (дата обращения: 01.06.2020)

14. Учебные материалы по 1С, форум [Электронный ресурс]. URL: <http://1c-pro.ru/> (дата обращения: 01.06.2020).
15. Филатова В.А 1С:Предприятие 8.1. Бухгалтерия предприятия. Управление торговлей. Управление персоналом / Виолетта Филатова. М.: "БХВ-Петербург". 2018. 288 с.
16. Форум начинающих и профессиональных программистов [Электронный ресурс]. URL: <http://www.cyberforum.ru/1c/> (дата обращения: 01.06.2020).
17. Функция формат в языке 1С 8.3 [Электронный ресурс]. URL: <https://helpme1c.ru/funkciya-format-v-yazyke-1s-8-v-primerax> (дата обращения: 01.06.2020).
18. Хрусталева Е.Ю. Профессиональная разработка в системе «1С: Предприятие 8.3». 2019. 213 с.
19. Json in 1С [Электронный ресурс]. URL: <https://1c-programmer-blog.ru/programmirovanie/json-v-1s.html> (дата обращения: 01.06.2020).
20. Logical Architecture vs Physical Architecture [Электронный ресурс]. URL: <https://simplicable.com/new/logical-architecture-vs-physical-architecture#:~:text=Logical%20architecture%20is%20a%20structural,the%20relationship%20between%20software%20components>. (дата обращения: 10.02.2020).
21. Rational Unified Process [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Rational_Unified_Process (дата обращения: 01.06.2020).
22. Seidl, M., Scholz, M., Huemer, C., Kappel, G. UML @ Classroom, Undergraduate Topics in Computer Science, 2015.
23. Visual Modeling with Rational Rose 2002 and UML [Электронный ресурс]. URL: https://www.oreilly.com/library/view/visual-modeling-with/0201729326/0201729326_ch01lev1sec8.html (дата обращения: 01.06.2020).

Приложение А

Фрагменты программного кода

```
&НаСервере Процедура ПриСозданииНаСервере(Отказ,  
СтандартнаяОбработка)  
    ДанныеОтчета = Неопределено;  
    Если Параметры.Свойство("ДанныеОтчета", ДанныеОтчета) Тогда  
        ЭтотОбъектДанные = РеквизитФормыВЗначение("Объект");  
        ЭтотОбъектДанные.ХранилищеОтчета = ДанныеОтчета;  
        ЭтотОбъектДанные.Записать();  
        ЗначениеВРеквизитФормы(ЭтотОбъектДанные,"Объект");  
    КонецЕсли;  
    ЭтотОбъектДанные = РеквизитФормыВЗначение("Объект");  
    ТабДок=ЭтотОбъектДанные.ХранилищеОтчета.Получить();  
    Если ТабДок<>Неопределено Тогда  
        ЭтаФорма.Отчет.Вывести(ТабДок);  
    КонецЕсли;  
КонецПроцедуры  
&НаКлиенте  
Процедура СохранитьОтчетНаКомпьютер(Команда)  
    СтандартнаяОбработка = Ложь;  
    ДиалогВыбораФайла = Новый  
ДиалогВыбораФайла(РежимДиалогаВыбораФайла.ВыборКаталога);  
    ДиалогВыбораФайла.Заголовок = "Укажите место хранения  
файла";  
  
    ДиалогВыбораФайла.Показать(Новый  
ОписаниеОповещения("ПутьКФайлуСохранениеЗавершение", ЭтотОбъект,  
Новый Структура("ДиалогВыбораФайла", ДиалогВыбораФайла)));
```

КонецПроцедуры

Продолжение Приложения А

&НаКлиенте

Процедура ПутьКФайлуСохранениеЗавершение(ВыбранныеФайлы,
ДополнительныеПараметры) Экспорт

Если ВыбранныеФайлы <> Неопределено Тогда

Адрес = ВыбранныеФайлы[0] + "\" + Объект.Наименование+\".xls\";

ЗаписатьНаСервере(Адрес);

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура ЗаписатьНаСервере(Адрес)

Отчет.Записать(Адрес, ТипФайлаТабличногоДокумента.XLS);

КонецПроцедуры

//Обработка «Авторизация Яндекс метрика»

&НаКлиенте

Процедура Записать(Команда)

Если Не ЗначениеЗаполнено(ПрофильСпециалиста) Тогда

ПоказатьПредупреждение(,"Не указан профиль специалиста!");

Возврат

КонецЕсли;

Если Не ЗначениеЗаполнено(ПолученныйТокен) Тогда

ПоказатьПредупреждение(,"Не указан токен!");

Возврат

КонецЕсли;

Если СписокСчетчиков.Количество() = 0 Тогда

ПоказатьПредупреждение(,"Не указаны счетчики!");

Возврат

КонецЕсли;

Продолжение Приложения А

```
ЗаписатьНаСервере());
ПоказатьПредупреждение("Данные записаны!");
КонецПроцедуры
&НаСервере
Процедура ЗаписатьНаСервере()
    ПрофильОбъект = ПрофильСпециалиста.ПолучитьОбъект();
    ПрофильОбъект.ТокенМетрики = ПолученныйТокен;
    ПрофильОбъект.IDПриложения = IDПриложения;
    Для Каждого Элемент Из СписокСчетчиков Цикл
        НоваяСтрока =
ПрофильОбъект.СписокСчетчиков.Добавить();
        НоваяСтрока.Счетчик = Элемент.Значение;
    КонецЦикла;
    ПрофильОбъект.Записать();
КонецПроцедуры
&НаКлиенте
Процедура ПрофильСпециалистаПриИзменении(Элемент)
    ПрофильСпециалистаПриИзмененииНаСервере();
КонецПроцедуры
&НаСервере
Процедура ПрофильСпециалистаПриИзмененииНаСервере()
    IDПриложения = ПрофильСпециалиста.IDПриложения;
    Если Не ЗначениеЗаполнено(IDПриложения) Тогда
        ОкноАвторизации = "oauth.yandex.ru";
    КонецЕсли;
КонецПроцедуры
&НаКлиенте
Процедура ОткрытьОкноПолученияТокена(Команда)
```

Продолжение Приложения А

Если ЗначениеЗаполнено(ПрофильСпециалиста) Тогда

Если Не ЗначениеЗаполнено(IdПриложения) Тогда

ПоказатьПредупреждение("Не указано ID Приложения");

ОкноАвторизации = "oauth.yandex.ru";

Иначе

ОкноАвторизации =

"oauth.yandex.ru/authorize?response_type=token&client_id="+IdПриложения;

КонецЕсли;

Иначе

ПоказатьПредупреждение("Не указан профиль!");

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ОткрытьОкноПриложений(Команда)

Если ЗначениеЗаполнено(ПрофильСпециалиста) Тогда

Если Не ЗначениеЗаполнено(IDПриложения) Тогда

ПоказатьПредупреждение("Не указано ID Приложения");

ОкноАвторизации = "oauth.yandex.ru";

Иначе

ОкноАвторизации = "metrika.yandex.ru/list";

КонецЕсли;

Иначе

ПоказатьПредупреждение("Не указан профиль!");

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ОткрытьОкноСпискаПриложений(Команда)

Если ЗначениеЗаполнено(ПрофильСпециалиста) Тогда

Продолжение Приложения А

ОкноАвторизации = "oauth.yandex.ru"

Иначе

ПоказатьПредупреждение("Не указан профиль!");

КонецЕсли;

КонецПроцедуры

//Отчет «Отчет Яндекс Метрики»

Процедура ПриКомпоновкеРезультата(ДокументРезультат,
ДанныеРасшифровки, СтандартнаяОбработка)

СтандартнаяОбработка = Ложь;

Если Не ЗначениеЗаполнено(ЭтотОбъект.ТекстЗапроса) Тогда

Сообщить("Текст запроса пустой!");

Возврат

КонецЕсли;

Если Не ЗначениеЗаполнено(ЭтотОбъект.ЗначениеТокена) Тогда

Сообщить("Не указан токен OAuth!");

Возврат

КонецЕсли;

ssl = Новый ЗащищенноеСоединениеOpenSSL(

Новый СертификатКлиентаWindows(),

Новый СертификатыУдостоверяющихЦентровWindows());

Соединение = Новый HTTPСоединение("api-
metrika.yandex.net",,,,,,ssl);

Запрос = Новый

HTTPЗапрос("/stat/v1/data?" + ЭтотОбъект.ТекстЗапроса);

Запрос.Заголовки.Вставить("Content-Type", "application/x-
yametrika+json");

Продолжение Приложения А

```
Запрос.Заголовки.Вставить("Authorization", "OAuth "+
ЭтотОбъект.ЗначениеТокена);

Ответ = Соединение.Получить(Запрос);
Если Ответ.КодСостояния = 200 Тогда
    ЧтениеJSON = Новый ЧтениеJSON;
    Строка = Ответ.ПолучитьТелоКакСтроку();
    ЧтениеJSON.УстановитьСтроку(Строка);
    Данные = ПрочитатьJSON(ЧтениеJSON,);
    Если Данные.data.Количество() = 0 Тогда
        Сообщить("По указанным параметрам нет данных!");
    Возврат
    КонецЕсли;
    Если Данные.data.Количество() = 1 Тогда
        Сообщить("Значение по выбранной метрике за
указанный период: " +Данные.data[0].metrics[0])
    Иначе
        ТаблицаДанных =
        ПреобразоватьМассивСтруктурВТаблицуЗначений(Данные.Data,
Данные.query.metrics);
        СхемаКомпоновкиДанных = ПолучитьМакет("Макет");;

        Настройки = КомпоновщикНастроек.ПолучитьНастройки();
        ДанныеРасшифровки = Новый ДанныеРасшифровкиКомпоновкиДанных;
        КомпоновщикМакета = Новый
        КомпоновщикМакетаКомпоновкиДанных;МакетКомпоновки =
        КомпоновщикМакета.Выполнить(СхемаКомпоновкиДанных, Настройки,
ДанныеРасшифровки);
        ВнешнийНаборДанных = Новый Структура("Таблица",
```

Продолжение Приложения А

```
ТаблицаДанных);
ПроцессорКомпоновкиДанных = Новый ПроцессорКомпоновкиДанных;
ПроцессорКомпоновкиДанных.Инициализировать(МакетКомпоновки,
ВнешнийНаборДанных, ДанныеРасшифровки);
        ДокументРезультат.Очистить();
ПроцессорВывода = Новый
ПроцессорВыводаРезультатаКомпоновкиДанныхВТабличныйДокумент;
ПроцессорВывода.УстановитьДокумент(ДокументРезультат);
ПроцессорВывода.Вывести(ПроцессорКомпоновкиДанных);
КонецЕсли;
Иначе
Сообщить("Код ошибки:" + Ответ.КодСостояния)
КонецЕсли;
КонецПроцедуры
Функция ПреобразоватьМассивСтруктурВТаблицуЗначений(Массив,
Метрики)
    ТЗ = Новый ТаблицаЗначений;
    ТЗ.Колонки.Добавить("Метрика");
    ТЗ.Колонки.Добавить("Значение");
    Счетчик = 0;
    КоличествоМетрик = Метрики.Количество();
    Для Каждого Элемент Из Массив Цикл
        Пока Счетчик < КоличествоМетрик Цикл
            НоваяСтрока = ТЗ.Добавить();
            Если КоличествоМетрик > 1 Тогда
                НоваяСтрока.Метрика = Элемент.dimensions[0].Name + " " +
Прав(Метрики[Счетчик],
```

Продолжение Приложения А

```
СтрДлина(Метрики[Счетчик]) - СтрНайти(Метрики[Счетчик], ":" ,  
НаправлениеПоиска.СКонца));  
Иначе  
    НоваяСтрока.Метрика =  
Элемент.dimensions[0].Name;  
    КонецЕсли;  
    НоваяСтрока.Значение = Элемент.metrics[Счетчик];  
    Счетчик = Счетчик+1;  
КонецЦикла;  
Счетчик = 0;  
КонецЦикла;  
Возврат ТЗ  
КонецФункции
```