ДЕПАРТАМЕНТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ТОМСКОЙ ОБЛАСТИ

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

Специальность 09.02.07 Информационные системы и программирование

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ УЧЕТА НАСЕЛЕНИЯ СТРАНЫ

Пояснительная записка

к курсовому проекту

КП.23.09.02.07.602.11. ПЗ

|  |  |  |
| --- | --- | --- |
| Студент  «\_\_»\_\_\_\_\_\_\_\_ 2023 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Н.В. Коровин |
| Руководитель  «\_\_»\_\_\_\_\_\_\_\_ 2023 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.Ю. Маюнова |
|  |  |  |
|  |  |  |
|  |  |  |

Томск 2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc138241849)

[1.ОБЩАЯ ЧАСТЬ 4](#_Toc138241850)

[1.1.Анализ предметной области 4](#_Toc138241851)

[1.2.Выбор средств и сред разработки 5](#_Toc138241852)

[2.СПЕЦИАЛЬНАЯ ЧАСТЬ 10](#_Toc138241853)

[2.1.Описание требований к информационной системе 10](#_Toc138241854)

[2.2.Диаграмма вариантов использования 12](#_Toc138241855)

[2.3.Диаграмма состояний 13](#_Toc138241856)

[2.4.Схема данных 28](#_Toc138241857)

[2.5.Пользовательские сценарии 33](#_Toc138241858)

[2.6.Прототип основных интерфейсов 36](#_Toc138241859)

[ЗАКЛЮЧЕНИЕ 43](#_Toc138241860)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 45](#_Toc138241861)

[ПРИЛОЖЕНИЕ А – Листинг программы 46](#_Toc138241862)

[ПРИЛОЖЕНИЯ Б – Инструкция пользователя 133](#_Toc138241863)

[ПРИЛОЖЕНИЕ В - Тесты 151](#_Toc138241864)

ВВЕДЕНИЕ

Во всех странах ведут учет граждан, которые проживают на территории данной страны. Знание о количестве жителей страны - важная информация, так как может показать крупные города с большим количеством населения, регионы страны, где малое количество граждан, или возможность перенаселения городов. Переписью занимаются с 1897 года это огромные количество информации о гражданах в стране. Раньше перепись населения производилась в виде записи данных граждан на бумагу, это дело было долгое и небезопасное, так как бумага может деформироваться различными способами и утратить информацию, которая была записана на ней. Придумывали разные способы хранения данных граждан, которые были записаны в общее хранилище данных, но сейчас одним из простых и удобных способов хранить данные – это хранение в цифровых источниках в ресурсах интернета или с помощью компьютера. Решили с источником данных, но удобство в переписи населения осталось прежним – неудобным. Так как ходить с большим количеством бумаги не удобно, поэтому можно перепись автоматизировать системой, в которой удобно будет работать с данными граждан. Поэтому разработанная автоматизированная информационная система для учета населения страны считается актуальной, так как это облегчает работу по переписи данных населения, а также хранит в себе данные, которые были записаны.

1. ОБЩАЯ ЧАСТЬ
   1. Анализ предметной области

Требуется разработать информационную систему для государственных нужд, в которых требуется переписать и собрать данные граждан. Каждый год государственные организации, которые занимаются аналитической деятельностью, связанные с переписью населения страны, проводят мероприятие, в течении которого проводится сбор данных у каждого гражданина. Во время данного мероприятия во всех субъектах страны, местные органы власти назначают ответственную организацию, чтобы собирать информацию о всех проживающих в данном субъекте, чтобы зафиксировать всю полученную информацию в общую базу данных. Если есть система, которая занимается хранением данных и работает на компьютере, то перенос информации с большого количества анкет в общую базу данных может занять несколько полных дней. Было бы удобно, если можно было бы составить электронный документ или таблицу, которая заполняется при опросе граждан, и после окончания опросов граждан можно было бы загрузить в общую базу данных.

Для решения данной проблемы понадобится система, которая должна перечень таких функций как:

1. внесение данных гражданином;
2. авторизация в учетную запись сотрудника;
3. выход из учетной записи сотрудника;
4. добавление данных гражданина сотрудником;
5. удаление данных гражданина сотрудником;
6. изменение данных граждан сотрудником;
7. сохранение данных;
8. нахождение идентичных данных гражданина в разных населённых пунктах;
9. поиск данных гражданина;
10. подсчет граждан по заданным критериям сотрудника;
11. создание новых учетных записей сотрудников;
12. вывод остальной информации (населенные пункты, образовательные учреждения и т.п.);
13. выгрузка данных граждан по субъектам в файл;
14. загрузка данных граждан из файла;
15. сохранение истории выполненных операций.
    1. Выбор средств и сред разработки

Во время проектирования разрабатывается диаграмма вариантов использования, диаграмма состояний, логическая модель и макеты пользовательского интерфейса.

Для разработки диаграмм, логической модели и технический проекта могут использоваться такое программное обеспечение, как:

1. Draw.io - бесплатное кроссплатформенное программное обеспечение для рисования графиков с открытым исходным кодом, разработанное на HTML5 и JavaScript. Его интерфейс можно использовать для создания диаграмм, таких как блок-схемы, каркасы, диаграммы UML, организационные диаграммы и сетевые диаграммы.
2. Microsoft Visio — векторный графический редактор, редактор диаграмм и блок-схем для Windows.Выпускается в трёх редакциях: Standard, Professional и Pro for Office 365.Аналогично с Adobe Reader, в стандартный набор программ MS Office входит только средство для просмотра и печати диаграмм Microsoft Visio Viewer. Полнофункциональная версия Microsoft Visio Professional для создания и редактирования монограмм и диаграмм.
3. Sybase PowerDesigner - инструментарий для создания бизнес-приложений, включающий в себя средства моделирования бизнес-процессов, возможности концептуального и физического проектирования баз данных и моделирования с использованием UML, и предоставляющий централизованный репозиторий для хранения моделей и объектов.
4. Erwin Data Modeler — это компьютерная программа для проектирования и документирования баз данных. Модели данных помогают визуализировать структуру данных, обеспечивая эффективный процесс организации, управления и администрирования таких аспектов деятельности предприятия, как уровень сложности данных, технологий баз данных и среды развертывания.

Будет использоваться Draw.io, так как в данной программе много шаблонов для создания диаграмм, блок-схем и тп. А также это программа имеет браузерную и клиентскую версию, а другие предложенные программы имеют только клиенскую версию. И она является бесплатной, в отличие от остальных.

Для разработки описания требований к информационной системе могут использоваться текстовые редакторы такие, как:

1. Microsoft Word — текстовый процессор, предназначенный для создания, просмотра, редактирования и форматирования текстов статей, деловых бумаг, а также иных документов, с локальным применением простейших форм таблично-матричных алгоритмов. Выпускается корпорацией Microsoft в составе пакета Microsoft Office.
2. ONLYOFFICE (в прошлом Teamlab) — офисный пакет с открытым исходным кодом, презентуется как разработанный компанией Ascensio System SIA с головным офисом в Риге (Латвия). Доступны облачная версия и версия для развертывания в локальной сети. Кроме того, редакторы ONLYOFFICE интегрируются с популярными платформами для совместной работы, включая ownCloud, Nextcloud, SharePoint и другими. ONLYOFFICE является официальным технологическим партнером Nextcloud, ownCloud, SeaFile, Pydio, eXo Platform и XWiki.
3. LibreOffice — кроссплатформенный, свободно распространяемый офисный пакет с открытым исходным кодом, созданный как ответвление OpenOffice.org в 2010 году. Разрабатывается сообществом из более чем 480 программистов под эгидой некоммерческого фонда The Document Foundation за счёт пожертвований отдельных лиц и организаций.
4. Apache OpenOffice (AOO, также известен как OpenOffice) — свободный пакет офисных приложений, является одним из преемников OpenOffice.org и наследник IBM Lotus Symphony.Официально поддерживается на платформах Linux, Windows, macOS Intel/PowerPC (поддержка оболочки Aqua находится в стадии альфа-тестирования). Существуют неофициальные портированные версии для FreeBSD, Linux PowerPC, OS/2 и Android.

Среди текстовых редакторов будет использоваться Onlyoffice из-за кроссплотформенности, бесплатной версии и удобной в пользовании.

Для разработки макетов пользовательского интерфейса могут использоваться такие приложения, как:

1. Adobe Photoshop - многофункциональный растровый графический редактор, разрабатываемый и распространяемый компанией Adobe Systems. В основном работает с растровыми изображениями, однако имеет некоторые векторные инструменты. Продукт является лидером рынка в области коммерческих средств редактирования растровых изображений и наиболее известной программой разработчика.
2. Fluid UI — это инструмент прототипирования мобильного интерфейса HTML5, который помогает разработчику пользовательского интерфейса быстро создавать прототипы, помещая предварительно встроенные виджеты в редактор drag-and-drop WYSIWYG на этапе разработки приложений. Он обеспечивает итерацию и совместную работу между пользователем и клиентом
3. Microsoft Paint - многофункциональный, но в то же время довольно простой в использовании растровый графический редактор компании Microsoft, входящий в состав всех операционных систем Windows, начиная с первых версий.
4. Pencil – бесплатный инструмент для прототипирования графического интерфейса с открытым исходным кодом, который люди могут легко установить и использовать для создания макетов на популярных настольных платформах.

Среди этих данных приложений будет использоваться приложение Pencil, так как удобна в управлении, расширяема шаблонами и есть на системах Windows, Linux, macOS.

Для реализации программного кода будет использоваться:

1. Язык программирования

Встроенный язык программирования 1С: Предприятие — язык программирования, который используется в семействе программ «1С: Предприятие». В версии 7.x являлся интерпретируемым языком сверхвысокого уровня. Интерпретация текста программного модуля в байт-код выполнялась в момент обращения к этому модулю в процессе работы, таким образом обычно интерпретировалась только часть текстов программных модулей (в версиях 7.7 и старше). Начиная с версии 8.х модули компилируются.

1. Средство создания пользовательского интерфейса

Визуальная среда разработки («Конфигуратор») является неотъемлемой частью пакета программ «1С: Предприятие».

1. Средство управления версиями программного кода

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая SSL), а для крупных корпоративных проектов предлагаются различные платные тарифные планы.

1. Средство получения исполняемого кода

Средой исполнения языка является программная платформа «1С: Предприятие».

1. Средство управления базой данных

«1С: Предприятие 8» выгодно отличается от универсальных инструментальных средств. Обычно, при создании бизнес-приложений с использованием универсальных сред разработки, используются отдельно поставляемые системы управления базами данных. А это значит, что разработчику приходится постоянно заботиться о преобразованиях между типами данных, поддерживаемыми той или иной системы управления базами данных, и типами, поддерживаемыми языком программирования.

1. Отладчик

Отладчик является встроенным в конфигуратор инструментом. Он помогает отлаживать программные модули, создаваемые в процессе разработки прикладного решения. Отладчик позволяет отслеживать последовательность выполнения операторов встроенного языка и просматривать значения переменных.

1. Средства тестирования приложения

Средств для тестирования не будет, так как средств мало или они используются в частных компаниях, либо за платную версию «1С: Предприятие 8». Будет проводиться ручное тестирование с методом «Белого ящика». Метод «Белого ящика» означает, что тестировщик имеет полный доступ к коду.

1. СПЕЦИАЛЬНАЯ ЧАСТЬ
   1. Описание требований к информационной системе

Пользователь — лицо, которое использует действующую систему для выполнения конкретной функции.

Администратор – пользователь, который имеет полный доступ ко всем функциям системы, которую он использует.

Гражданин – человек, который проживает в Российской Федерации и имеет гражданство страны.

Сотрудник – человек, который является работником компании и работает с системой.

Учетная запись – связанные данные пользователя в один профиль, которые заполняются и сохраняются в базу данных.

База данных — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

Населенные пункты - общее название мест, где живут люди (город, деревня, посёлок и т. п.).

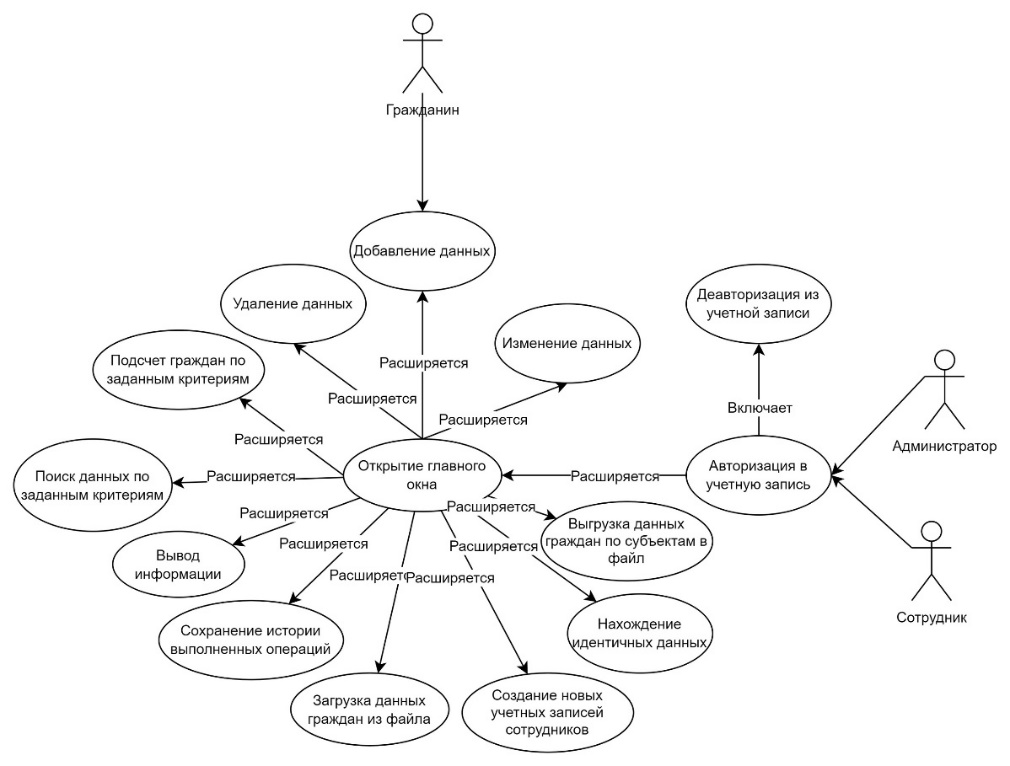
Образовательные учреждения - учреждение, осуществляющее образовательный процесс, то есть реализующее одну или несколько образовательных программ и (или) обеспечивающее содержание и воспитание обучающихся, воспитанников.

Субъект - название территориальных единиц верхнего уровня - регионов в Российской Федерации.

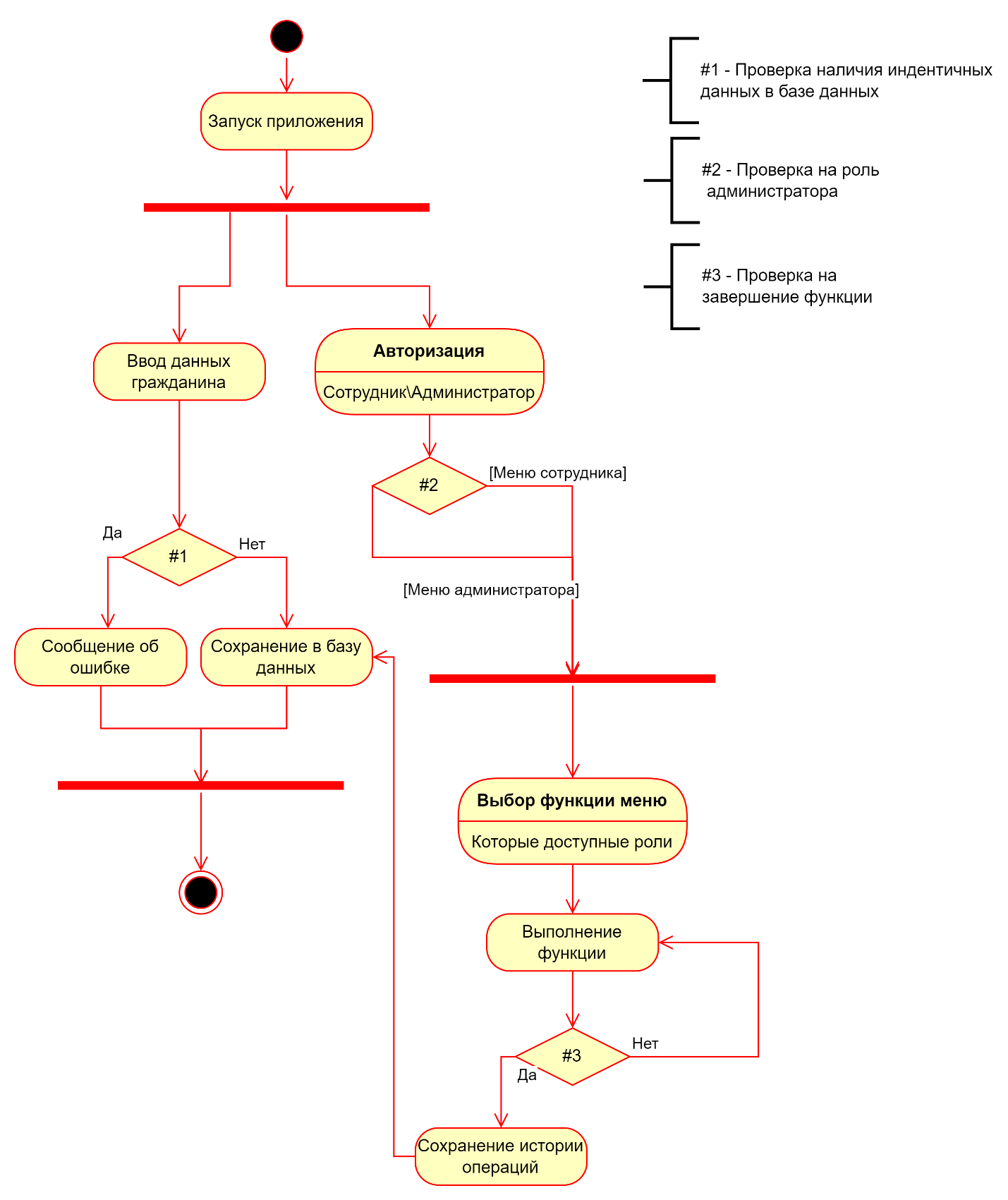
Файл – именованная область данных на носителе информации, используемая как базовый объект взаимодействия с данными в операционных системах.

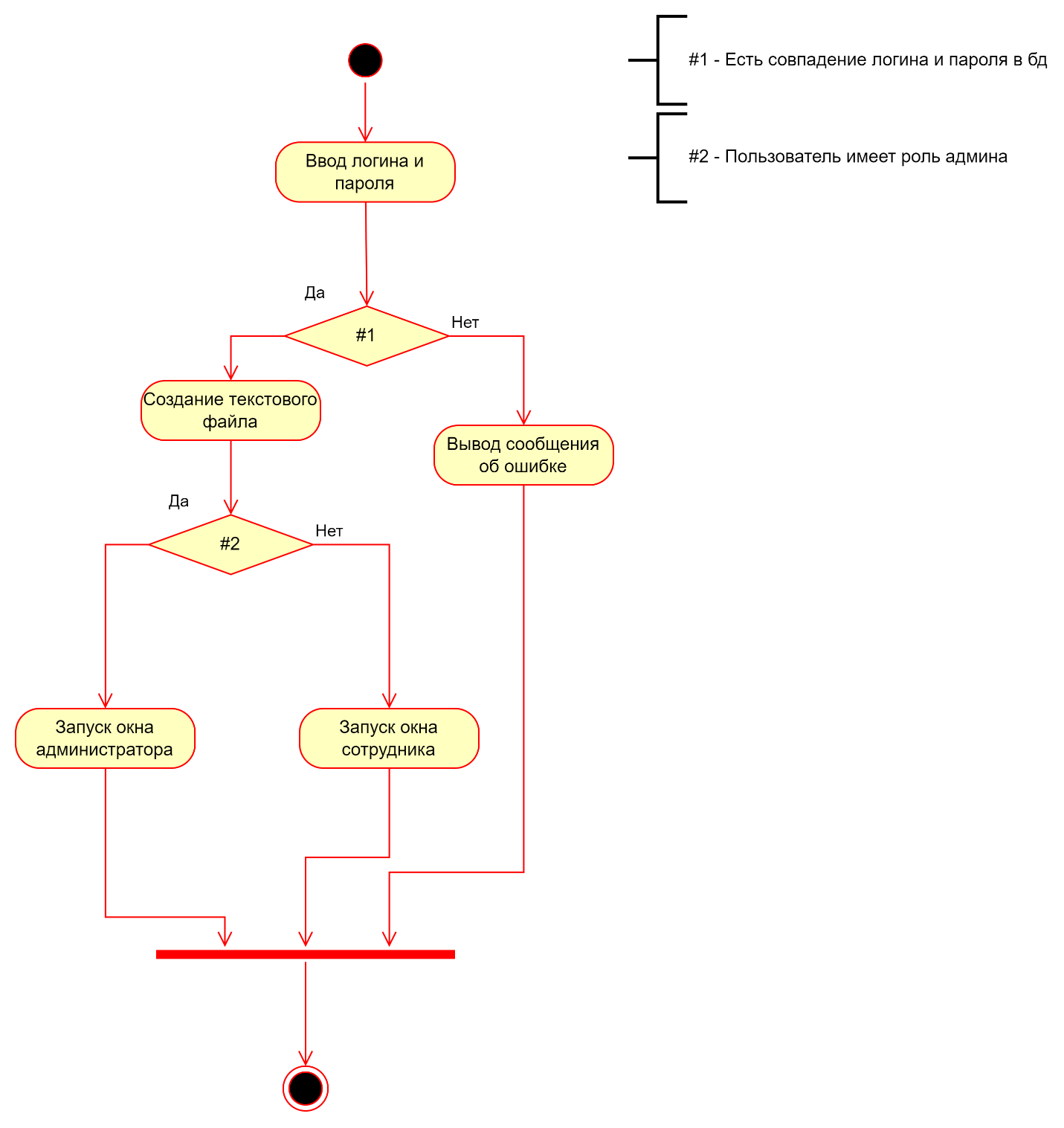
Для выполнения цели необходимо, чтобы разрабатываемая система выполняла ряд требований:

1. давать возможность внесение данных гражданина в неавторизованном режиме;
2. возможность авторизации сотрудника;
3. возможность выход из учетной записи сотрудника;
4. предоставлять возможность внесения данных граждан сотрудником;
5. предоставлять возможность удаление данных граждан сотрудником;
6. предоставлять возможность изменение данных граждан сотрудником;
7. обеспечивать сохранение данных в базу данных;
8. находить идентичные данные гражданина в разных населенных пунктах;
9. возможность поиска данных по заданным критерия сотрудника;
10. обеспечивать возможность подсчет граждан по заданным критериям сотрудника;
11. возможность регистрации новых учетных записей администратором;
12. обеспечивать вывод информации (населенные пункты, образовательные учреждения и т.п.);
13. возможность выгрузки данных граждан по субъектам в файл;
14. возможность загрузки данных граждан из файла;
15. обеспечивать сохранение истории выполненных операций.
    1. Диаграмма вариантов использования

  
Рисунок 1 – Диаграмма вариантов использования

* 1. Диаграмма состояний

  
Рисунок 2 – Общая диаграмма состояний

  
Рисунок 3 – Диаграмма состояний функции «Авторизации»

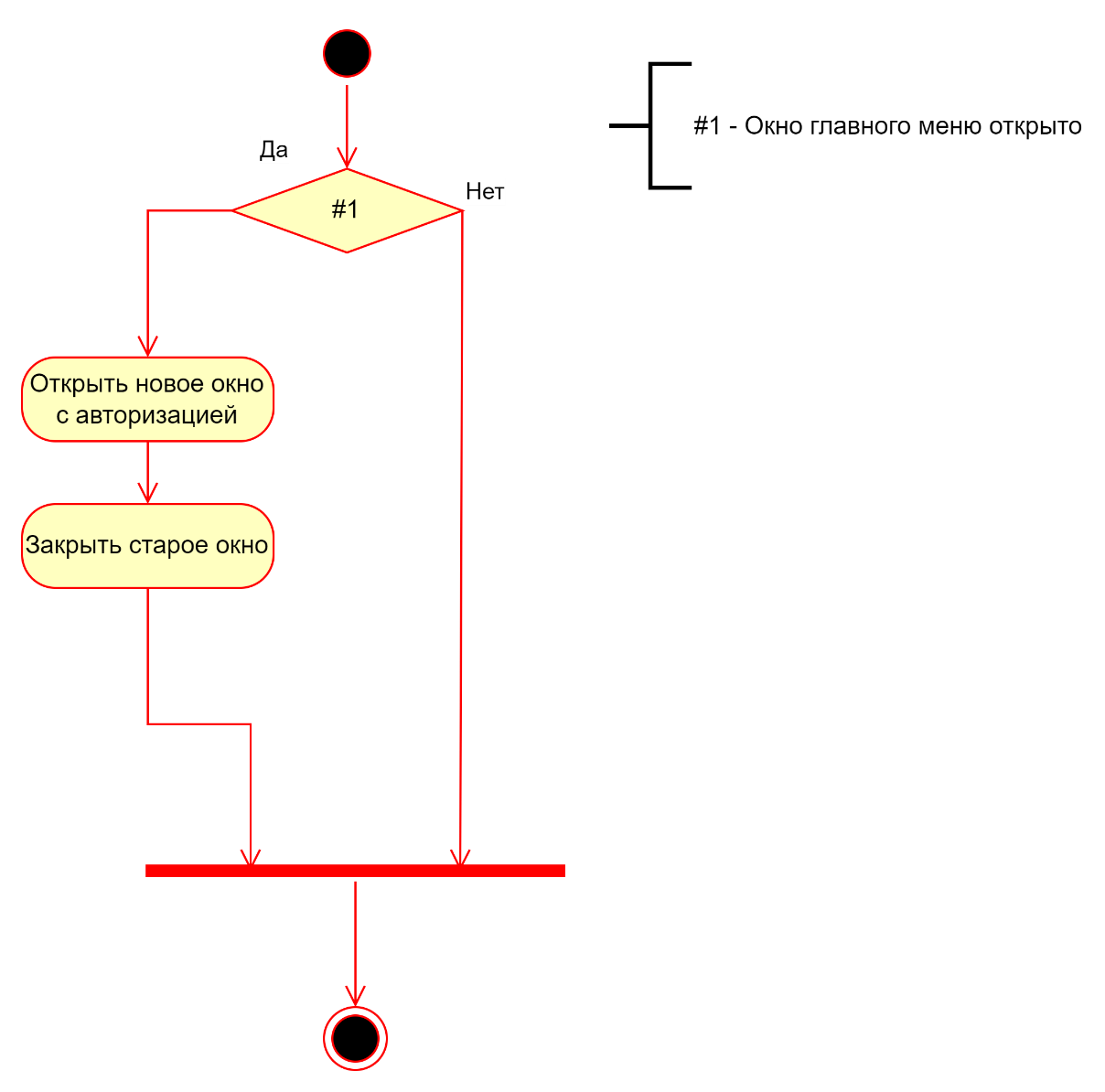
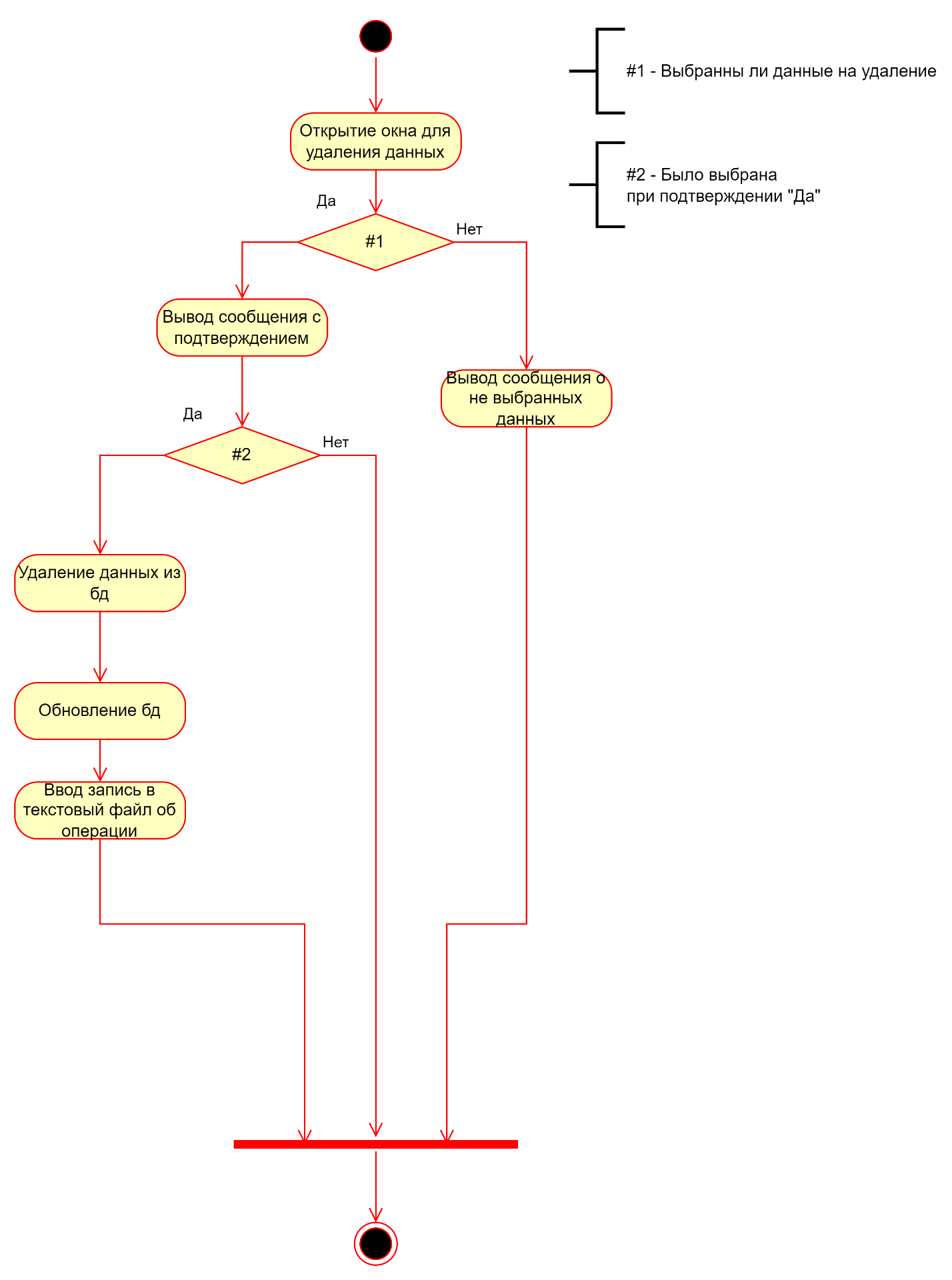
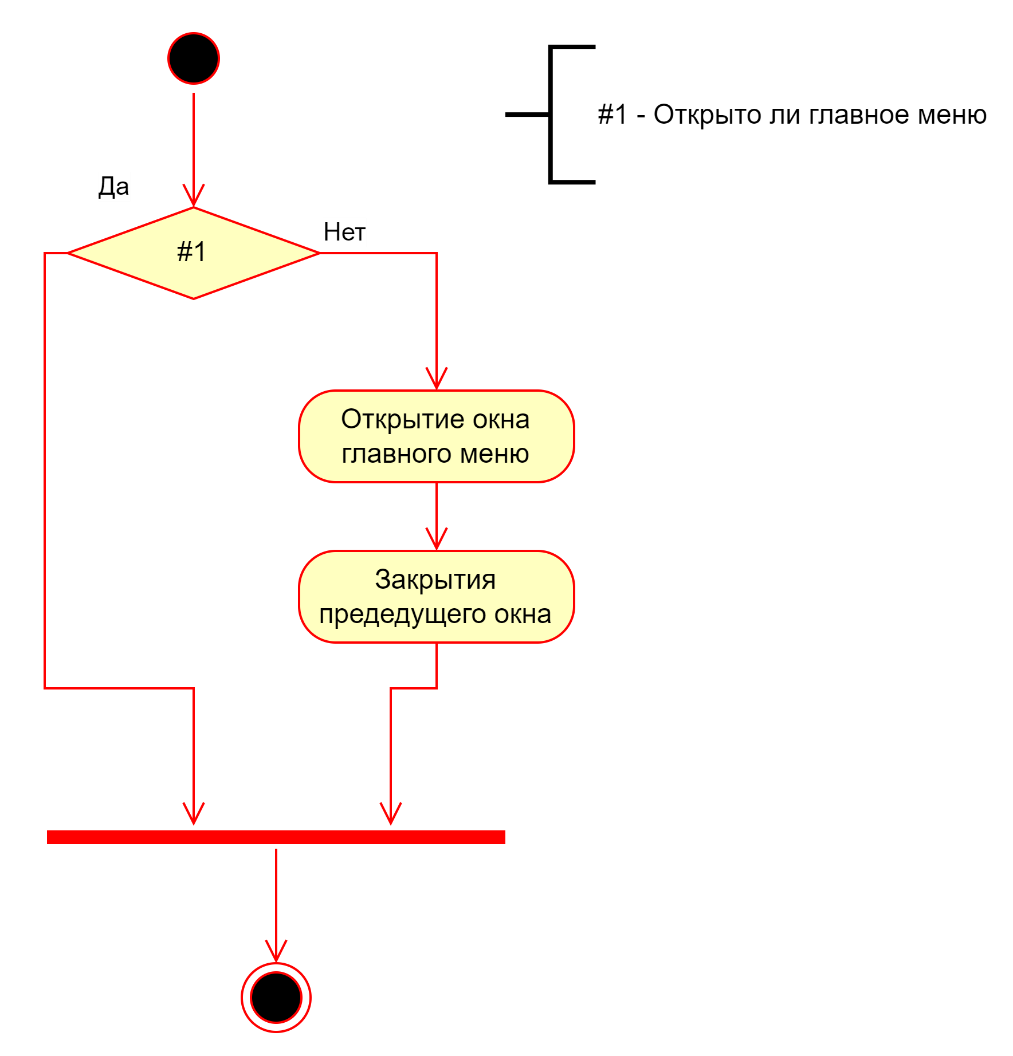
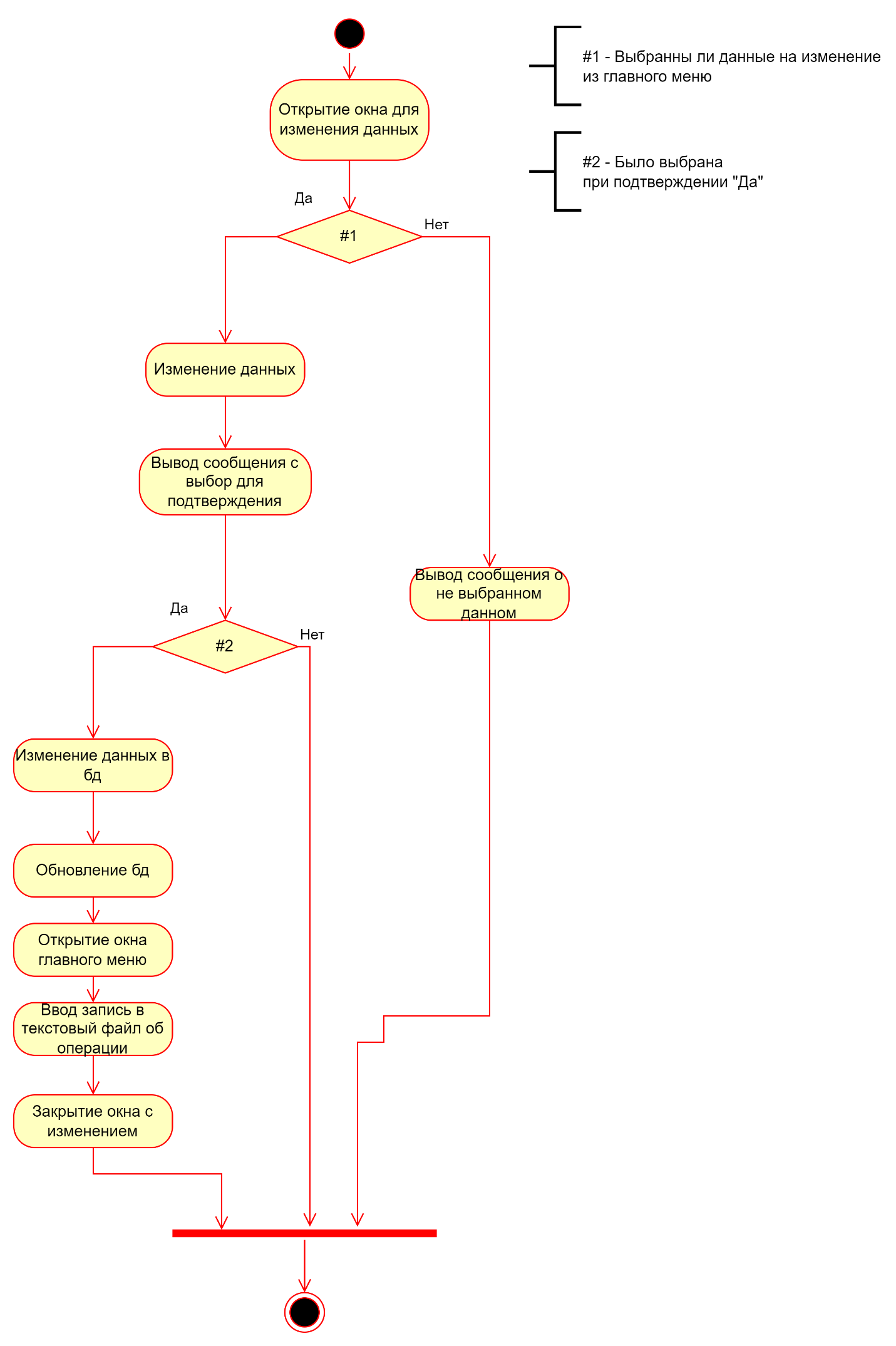
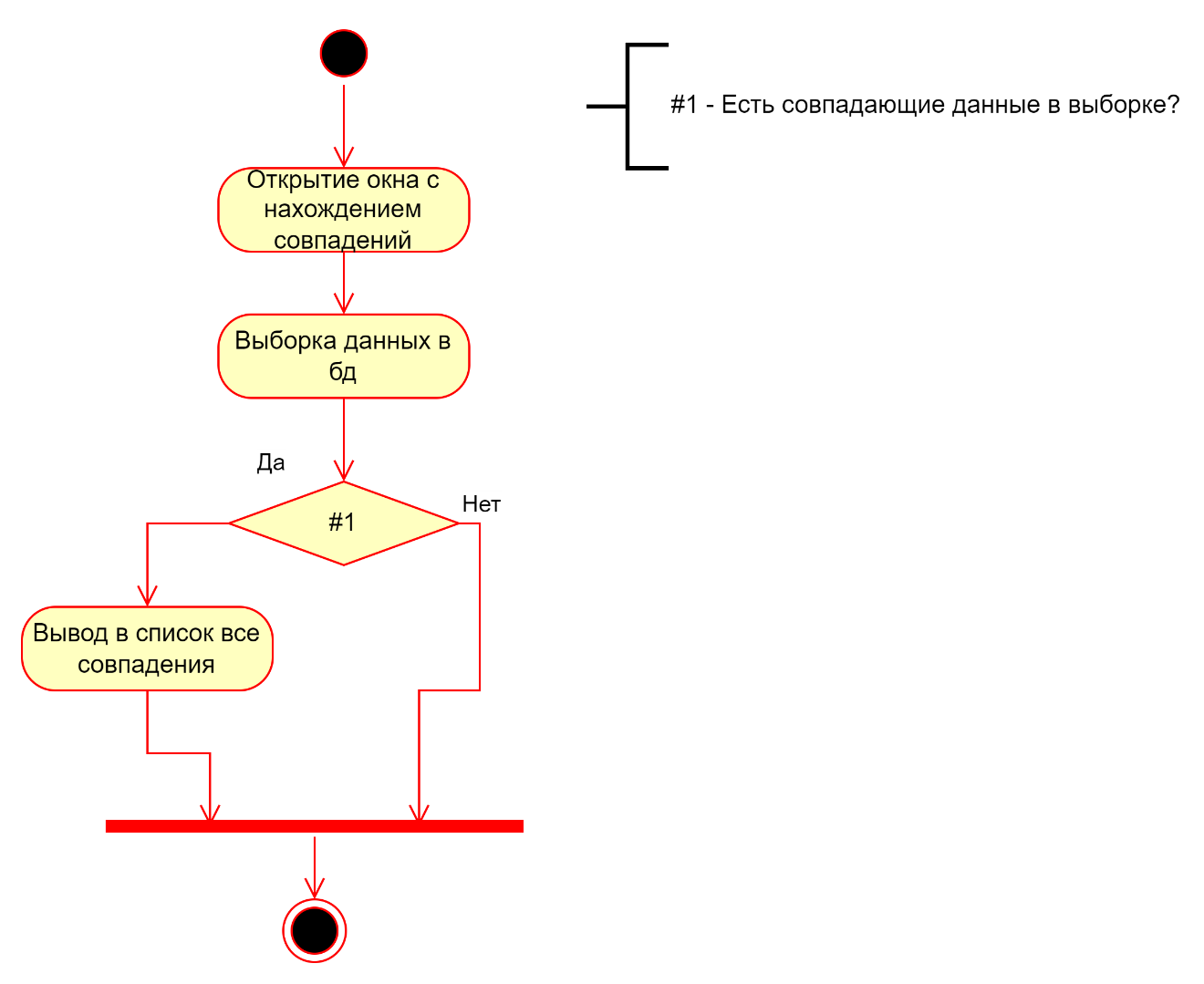
  
Рисунок 4 – Диаграмма состояний функции «Выход из учетной записи»

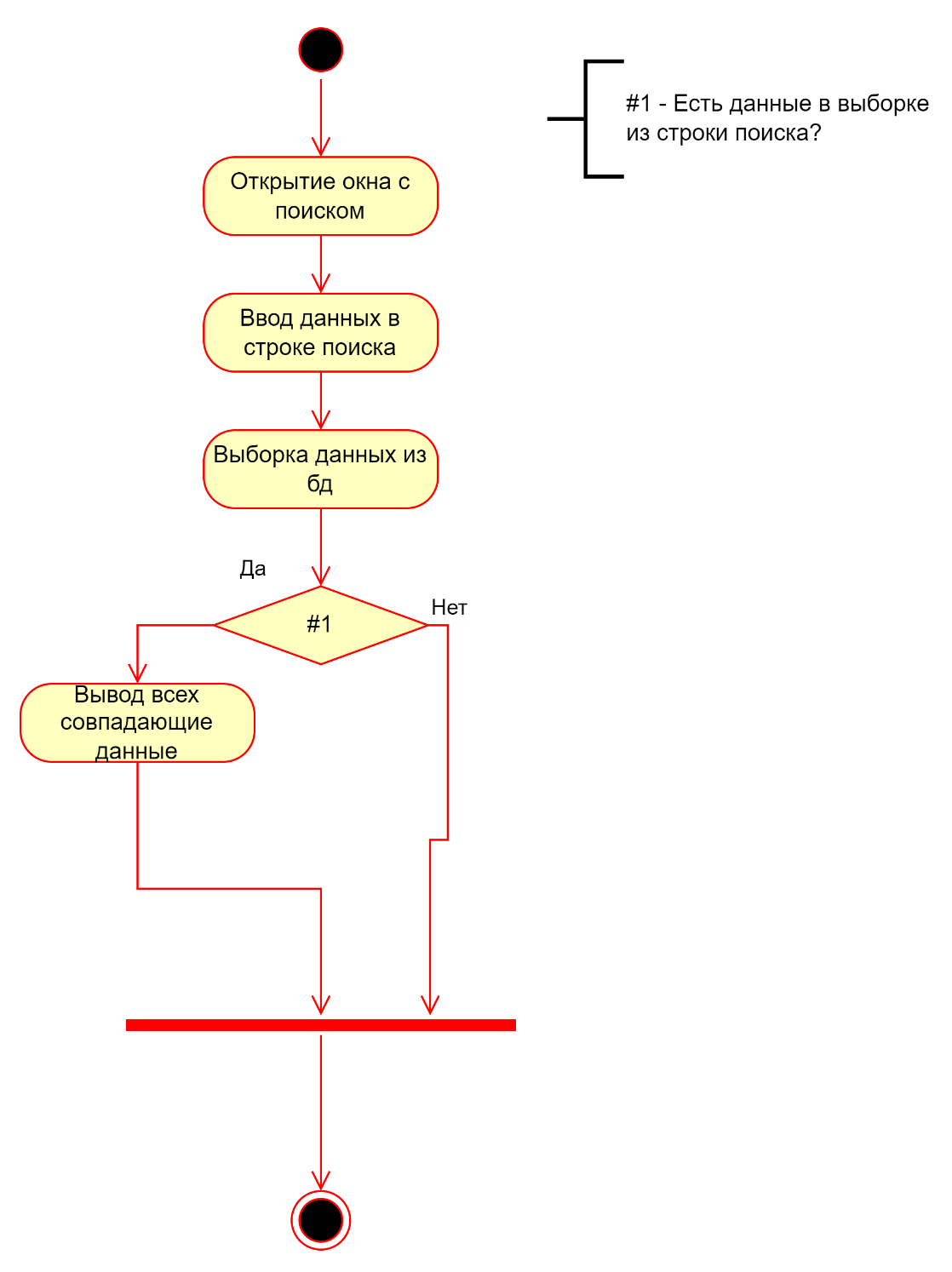
  
Рисунок 4 – Диаграмма состояний функции «Добавления данных»

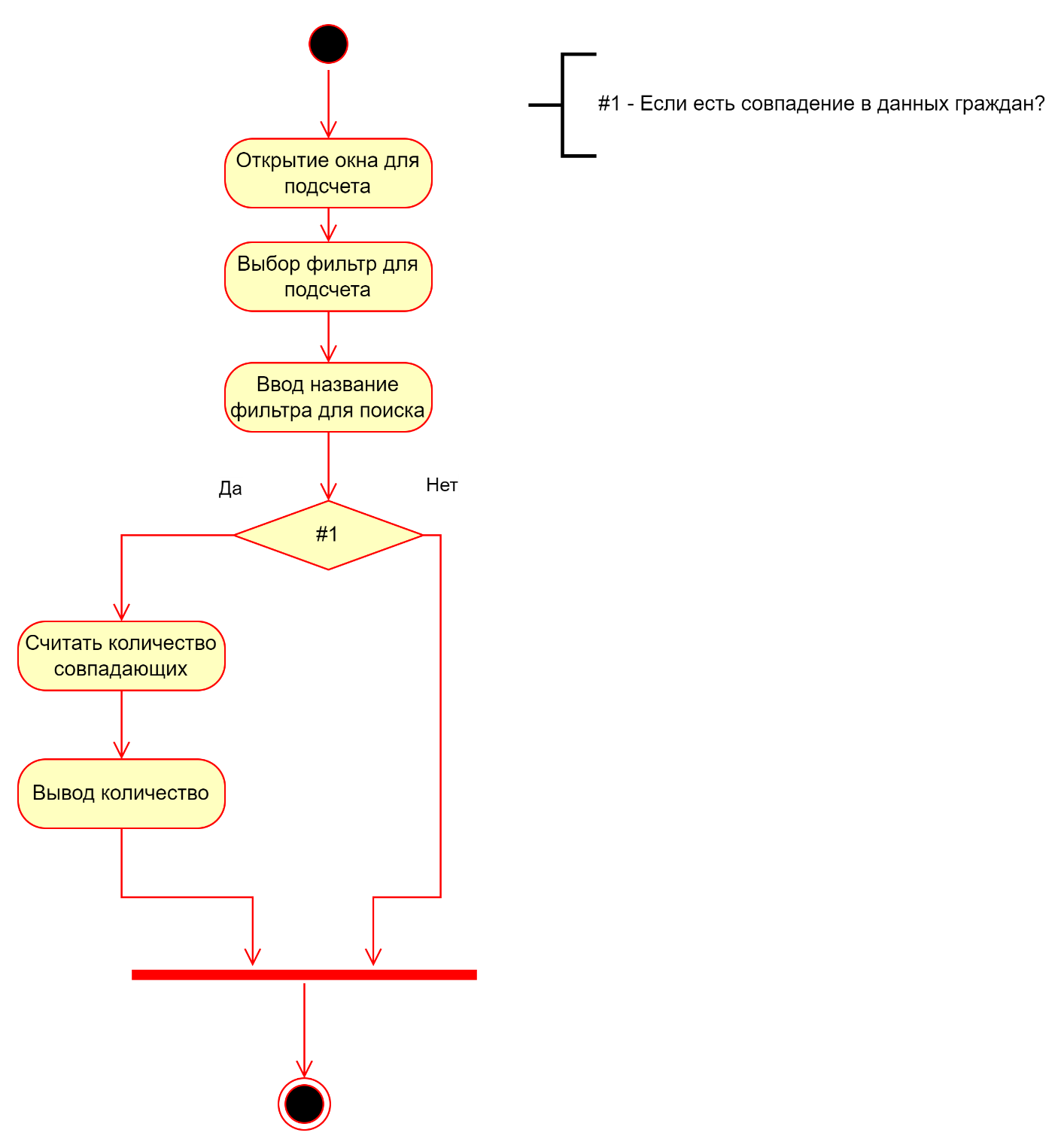
  
Рисунок 4 – Диаграмма состояний функции «Удаление данных гражданина»

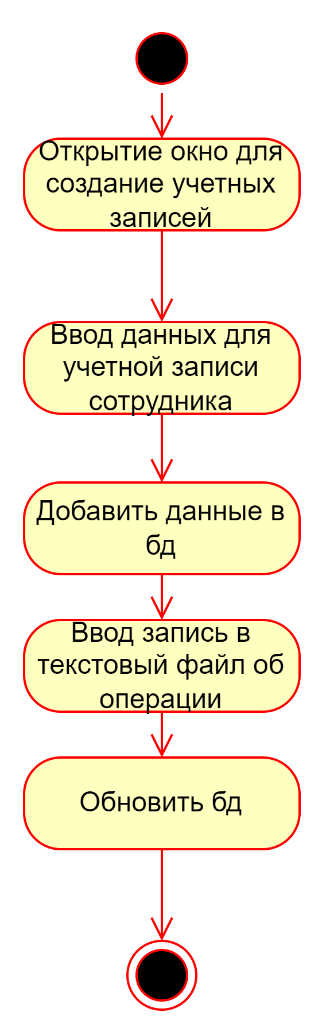
  
Рисунок 5 – Диаграмма состояний функции «Возврат на главное меню»

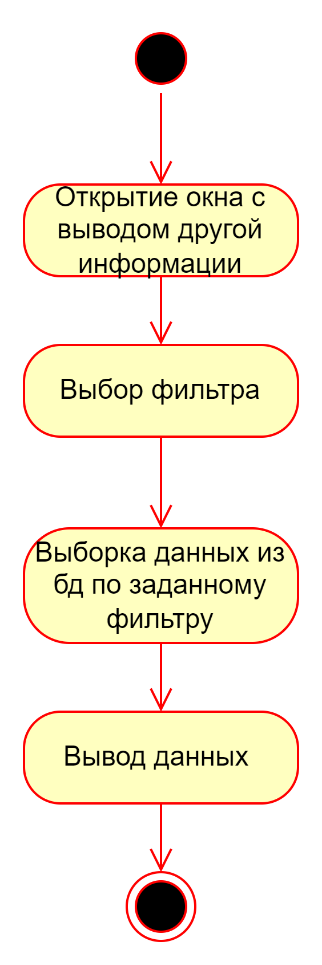
  
Рисунок 5 – Диаграмма состояний функции «Изменение данных гражданина»

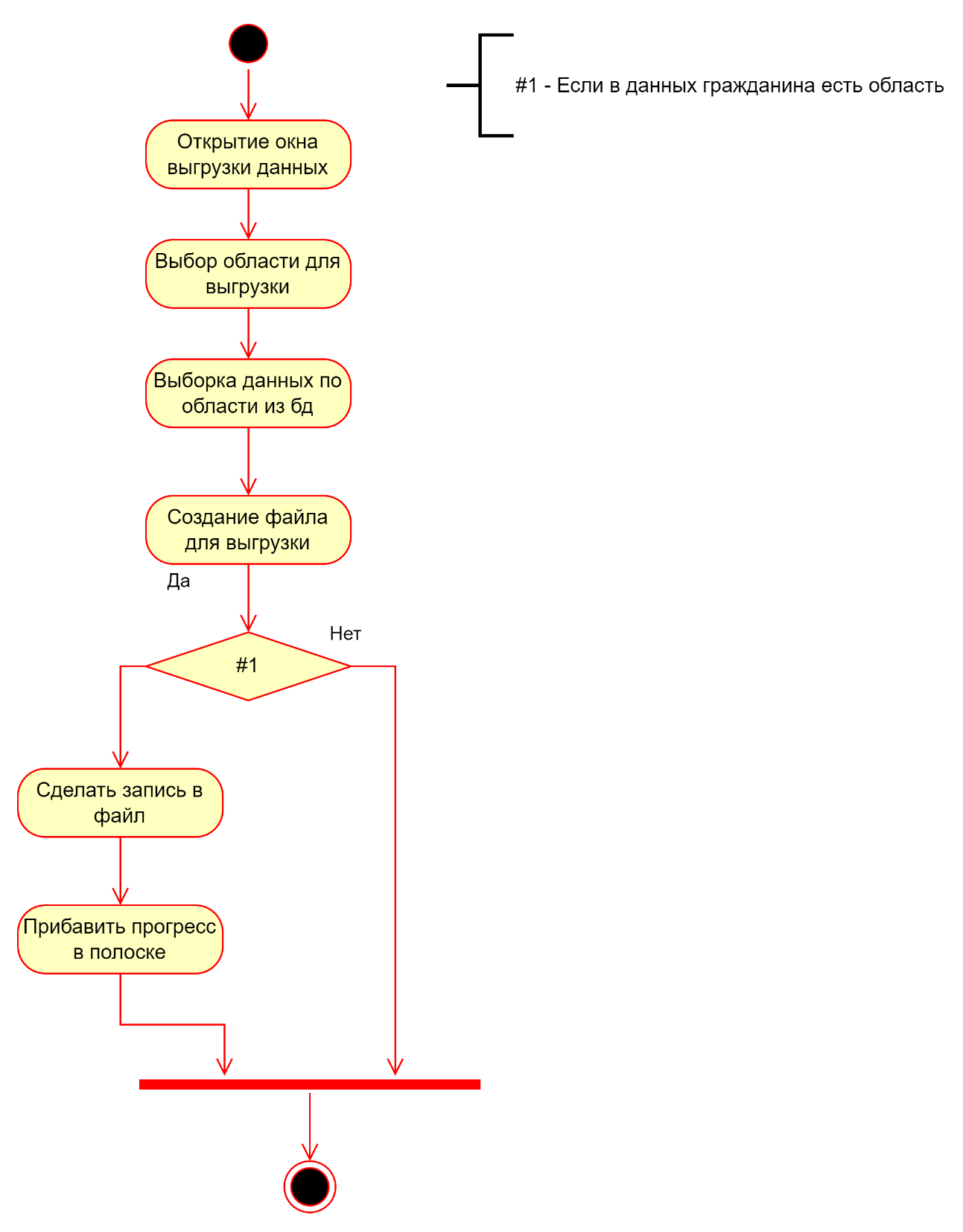
  
Рисунок 6 – Диаграмма состояний функции «Нахождение идентичный данных»

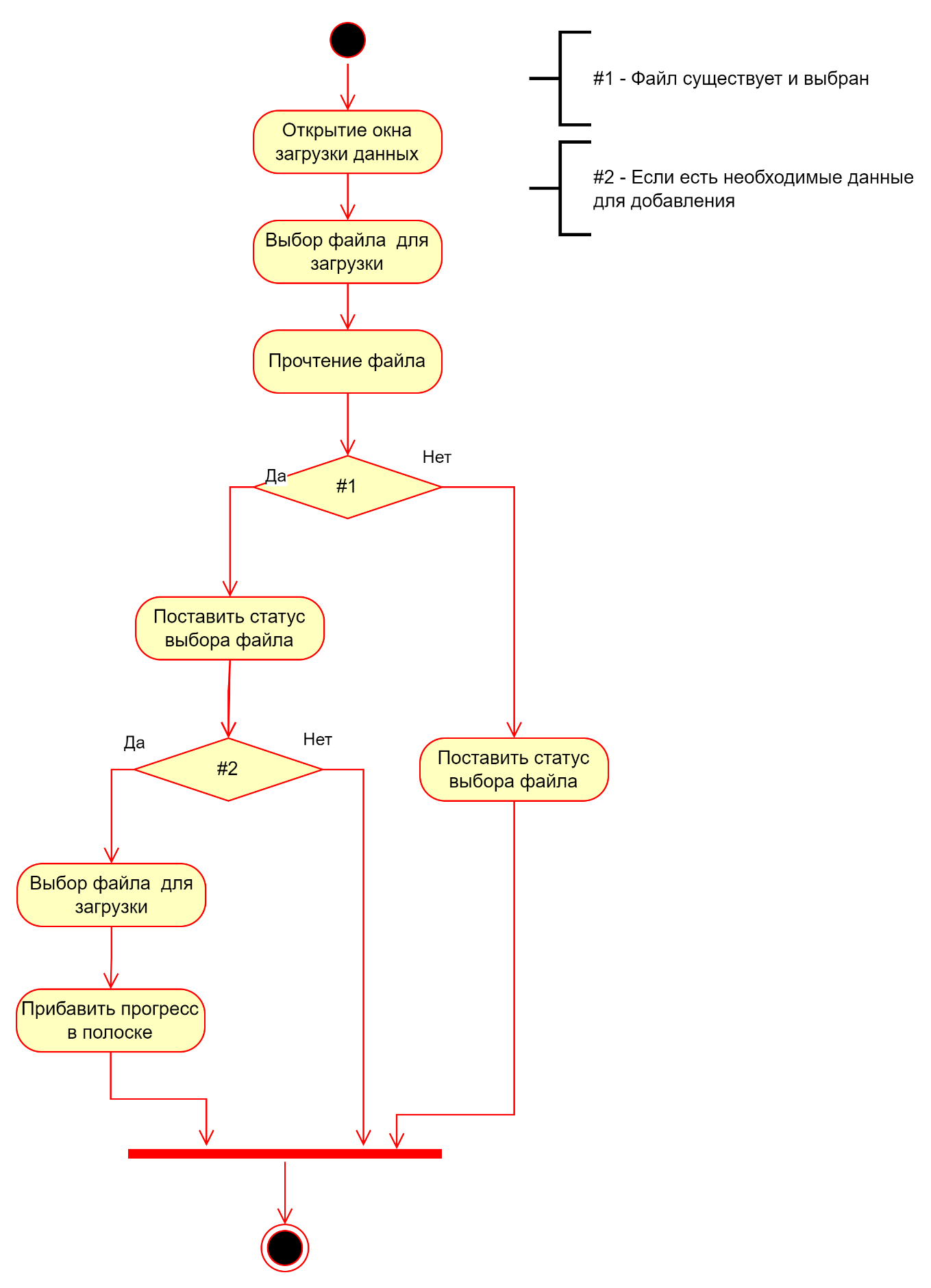
  
Рисунок 7 – Диаграмма состояний функции «Поиск данных гражданина»

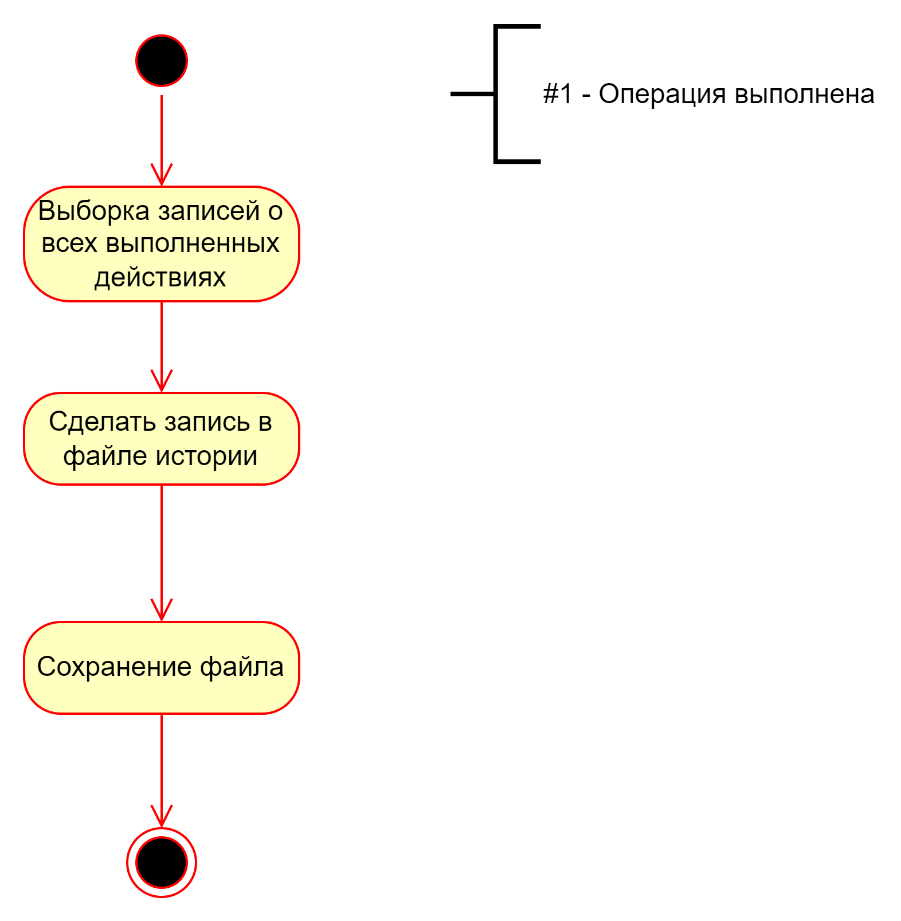
  
Рисунок 8 – Диаграмма состояний функции «Подсчет граждан по заданным критериям сотрудника»

  
Рисунок 9 – Диаграмма состояний функции «Создание учетных записей пользователей»

  
Рисунок 10 – Диаграмма состояний функции «Вывод остальной информации»

  
Рисунок 11 – Диаграмма состояний функции «Выгрузка данных граждан по субъектам»

  
Рисунок 12 – Диаграмма состояний функции «Загрузки данных граждан из файла»

  
Рисунок 13 – Диаграмма состояний функции «Сохранение истории выполненных действий»

* 1. Схема данных

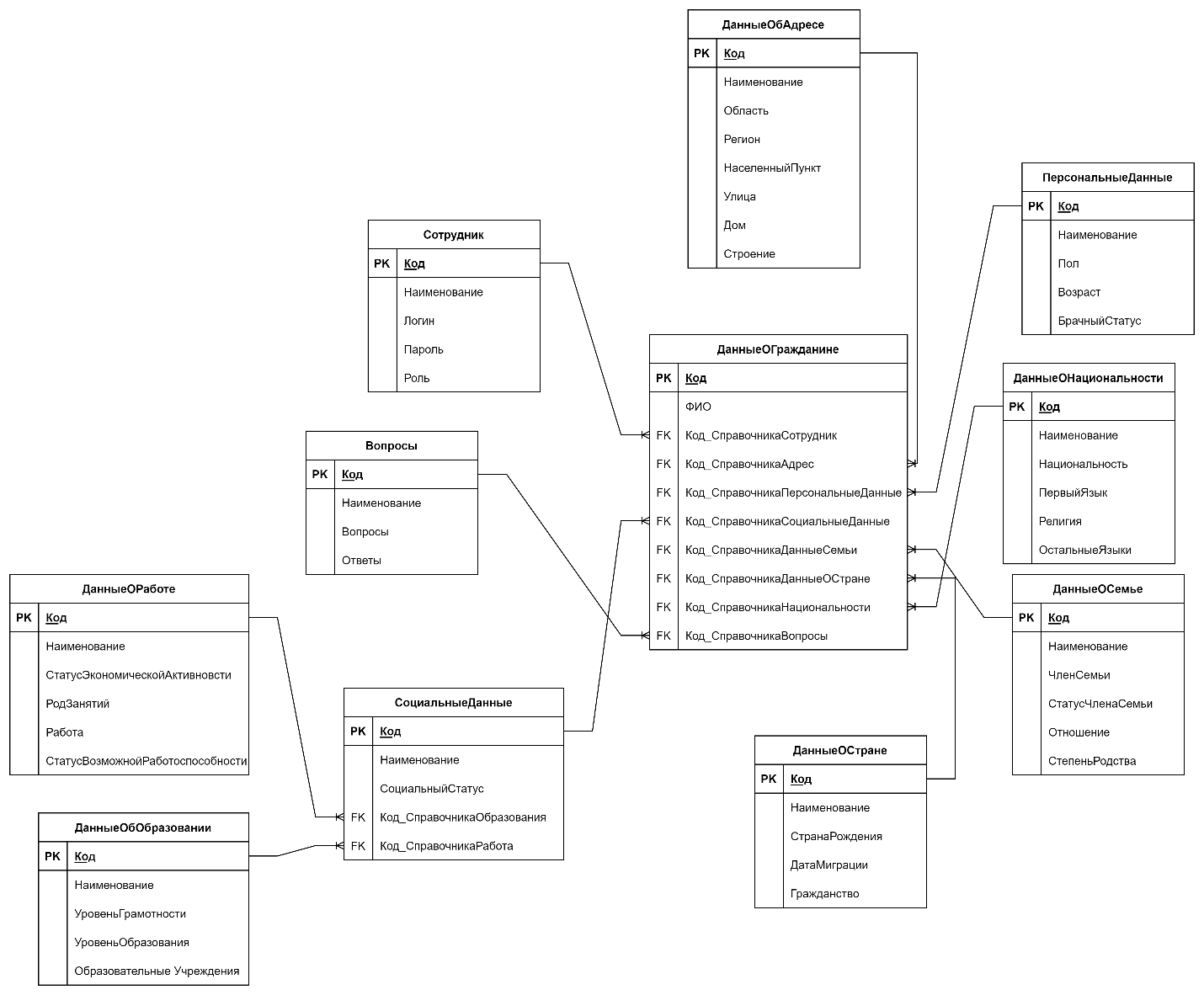
  
Рисунок 14 – Логическая модель базы данных

Таблица 1 – Схема данных справочника «ДанныеОбАдресе»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число(9) | Да | Идентификатор справочника |
|  | Наименование | Строка(25) | Да | Наименование справочника |
|  | Область | Строка(30) |  | Название область |
|  | Регион | Строка(25) |  | Название регион |
|  | НаселенныйПункт | Строка(20) |  | Название населенного пункта |
|  | Улица | Строка(20) |  | Название улицы |
|  | Дом | Число(3) |  | Номер дома |
|  | Строение | Строка(3) |  | Номер строение |

Таблица 2 – Схема данных справочника «Сотрудник»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число(9) | Да | Идентификатор справочника |
|  | Наименование | Строка(25) | Да | Наименование справочника |
|  | Логин | Строка(10) | Да | Логин сотрудника |
|  | Пароль | Строка(15) | Да | Пароль сотрудника |
|  | Роль | Булево |  | True - админ-сотрудник, False - сотрудник |

Таблица 3 – Схема данных справочника «Вопросы»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число(9) | Да | Идентификатор справочника |
|  | Наименование | Строка(25) | Да | Наименование справочника |
|  | Вопросы | Строка |  | Вопросы, заданные гражданину |
|  | Ответы | Строка |  | Ответы гражданина |

Таблица 4 – Схема данных справочника «ПерсональныеДанные»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число(9) | Да | Идентификатор справочника |
|  | Наименование | Строка(25) | Да | Наименование справочника |
|  | Пол | Булево |  | True - мужчина, False - женщина |
|  | Возраст | Число(2) |  | Возраст гражданина |
|  | БрачныйСтатус | Строка(25) |  | Семейное положение |

Таблица 5 – Схема данных справочника «ДанныеОНациональности»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число | Да | Идентификатор справочника |
|  | Наименование | Строка(25) | Да | Наименование справочника |
|  | Национальность | Строка(20) |  | Национальность |
|  | ПервыйЯзык | Строка(20) |  | Родной язык |
|  | Религия | Строка(15) |  | Религия |
|  | ДругиеЯзыки | Строка |  | Остальные языки |

Таблица 6 – Схема данных справочника «ДанныеОСемье»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число(9) | Да | Идентификатор справочника |
|  | Наименование | Строка(25) | Да | Наименование справочника |
|  | ЧленСемьи | Строка |  | Член семьи |
|  | СтатусЧленаСемьи | Строка |  | Статус члена семьи |
|  | Отношение | Строка |  | Отношение к члену семьи |
|  | СтепеньРодства | Строка |  | Стадия родства с членом семьи |

Таблица 7 – Схема данных справочника «ДанныеОСтране»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число(9) | Да | Идентификатор справочника |
|  | Наименование | Строка(25) | Да | Наименование справочника |
|  | СтранаРождения | Строка(10) |  | Страна рождения |
|  | ДатаМиграции | Дата |  | Дата переезда в РФ |
|  | Гражданство | Строка(10) |  | Гражданство |

Таблица 8 – Схема данных справочника «ДанныеОРаботе»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число(9) | Да | Идентификатор справочника |
|  | Наименование | Строка(25) | Да | Наименование справочника |
|  | СтатусЭкономическойАктивности | Строка(20) |  | Статус экономической активности |
|  | РодЗанятий | Строка(20) |  | Род занятий |
|  | Работа | Строка(25) |  | Работа |
|  | СтатусВозможнойРаботоспособности | Строка |  | Возможность найти работу |

Таблица 9 – Схема данных справочника «ДанныеОбОбразовании»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число(9) | Да | Идентификатор справочника |
|  | Наименование | Строка(25) | Да | Наименование справочника |
|  | УровеньГрамотности | Строка |  | Уровень грамотности |
|  | УровеньОбразования | Строка |  | Уровень образованности |
|  | ОбразовательныеУчреждения | Строка |  | Образовательные учреждения |

Таблица 10 – Схема данных справочника «СоциальныеДанные»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число(9) | Да | Идентификатор справочника |
|  | Наименование | Строка(25) | Да | Наименование справочника |
|  | СоцальныйСтатус | Строка |  | Социальный статус |
| FK | Код\_СправочникаОбразования | Число | Да | Внешний ключ от Справочник.ДанныеОбОбразовании |
| FK | Код\_СправочникаРабота | Число | Да | Внешний ключ от Справочник.ДанныеОРаботе |

Таблица 11 – Схема данных справочника «ДанныеОГражданине»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Имя реквизита | Тип данных и размер | Обязательное поле? | Комментарий |
| PK | Код | Число(9) | Y | Идентификатор справочника |
|  | Наименование | Строка(25) | Y | Наименование справочника |
|  | ФИО | Строка(50) | Y | Фамилия Имя Отчество гражданина |
| FK | Код\_СправочникаСотрудник | Ссылка | Y | Внешний ключ от Справочник.Сотрудник |
| FK | Код\_СправочникаАдрес | Ссылка | Y | Внешний ключ от Справочник.ДанныеОбАдресе |
| FK | Код\_СправочникаПерсональныеДанные | Ссылка | Y | Внешний ключ от Справочник.ПерсональныеДанные |
| FK | Код\_СправочникаСоциальныеДанные | Ссылка | Y | Внешний ключ от Справочник.СоциальныеДанные |
| FK | Код\_СправочникаДанныеСемьи | Ссылка | Y | Внешний ключ от Справочник.ДанныеОСемье |
| FK | Код\_СправочникаДанныеОСтране | Ссылка | Y | Внешний ключ от Справочник.ДанныеОСтране |
| FK | Код\_СправочникаНациональности | Ссылка | Y | Внешний ключ от Справочник.ДанныеОНациальности |
| FK | Код\_СправочникаВопросы | Ссылка | Y | Внешний ключ от Справочник.Вопросы |

* 1. Пользовательские сценарии

После запуска системы перед пользователем откроется «Окно авторизации», если пользователь не является сотрудником или администратором системы, то он может воспользоваться функцией добавления данных гражданина на окне авторизации, нажав на вторую доступную кнопку. После нажатия на кнопку перед пользователем откроется окно с пустыми полями, которые он может заполнить и после заполнения нажать на кнопку, которая отвечает за добавление данных, чтобы записать данных в базу данных.

Если является сотрудником или администратором системы, то должен логин и пароль для доступа к остальным функциям, которые выделены для данного пользователя.

Если сотрудник авторизировался в систему, то перед откроется окно «Меню сотрудника», в котором отображены функции в виде кнопок. Каждая кнопка, которая есть в меню сотрудника, кроме кнопки сохранения история действий, открывает окно, которое отвечает за функцию, которая привязана к этой кнопке.

Если администратор системы авторизируется, то перед ним откроется окно «Меню админа», но данное очень схоже по функционалу с «Меню сотрудника», кроме одной добавленной функцией, которая есть только у администратора. Функция «Создание сотрудников» отвечает за добавление учетных данных для сотрудников и администраторов системы.

 Кроме этой функции есть в меню сотрудника и администратора такие как: добавление данных гражданина; удаление данных гражданина; изменение данных гражданина; выгрузка данных граждан; загрузка данных граждан; поиск совпадения данных в таблицах; подсчет данных по определенным критериям; поиск данных гражданина по ФИО; вывод других данных, записанные в базу данных системы; сохранение выполненных действий во время работы системы. На каждом окне есть дополнительные функции выхода из системы и возвращения в меню. Выход из системы закрывает все открытые окна и отправляет вас на окно авторизации, возвращение в меню отвечает за закрытие окон и отправку в меню, если авторизировался пользователь сотрудник, то отправит в меню сотрудника, ели администратор, то в меню админа.

Большинство функций выполняют то, что скрывается в их названии. Так если разбирать функцию «Добавления данных гражданина», то это функция добавляет записанные данными поля в базу данных системы. Функция «Удаление данных гражданина» выполняет удаление записи из нескольких таблиц, с которыми связана данная запись гражданина.  
 Функция «Изменения данных» выполняет изменение данных, которые были выбраны по введенному в поле ФИО. Если гражданин с таким существует, то найденные данные гражданина заполняются в подписанные поля, если же такой гражданин не найден, то появляется надпись об отсутствии данного гражданина.

Функция «Выгрузка данных граждан» выполняет выгрузку данных из базы данных по выбранной области и записывает в текстовый документ эти данные.

Функция «Загрузка данных граждан» выполняет загрузку данных из текстового файла в базу данных. Запись в текстовом файле должна соблюдаться правилам для корректной работы системы, данное правило было отображено в инструкции пользователя. Если данная запись построена по нужному правилу системы, то система разбивает на несколько частей и заполняет её в таблицу базы данных, которой она принадлежит.

Функция «Поиск совпадения данных в таблицах» выполняет нахождение дубликатов в таблице базы данных, которую выберет пользователь, и вывод результатов поиска в виде таблицы.

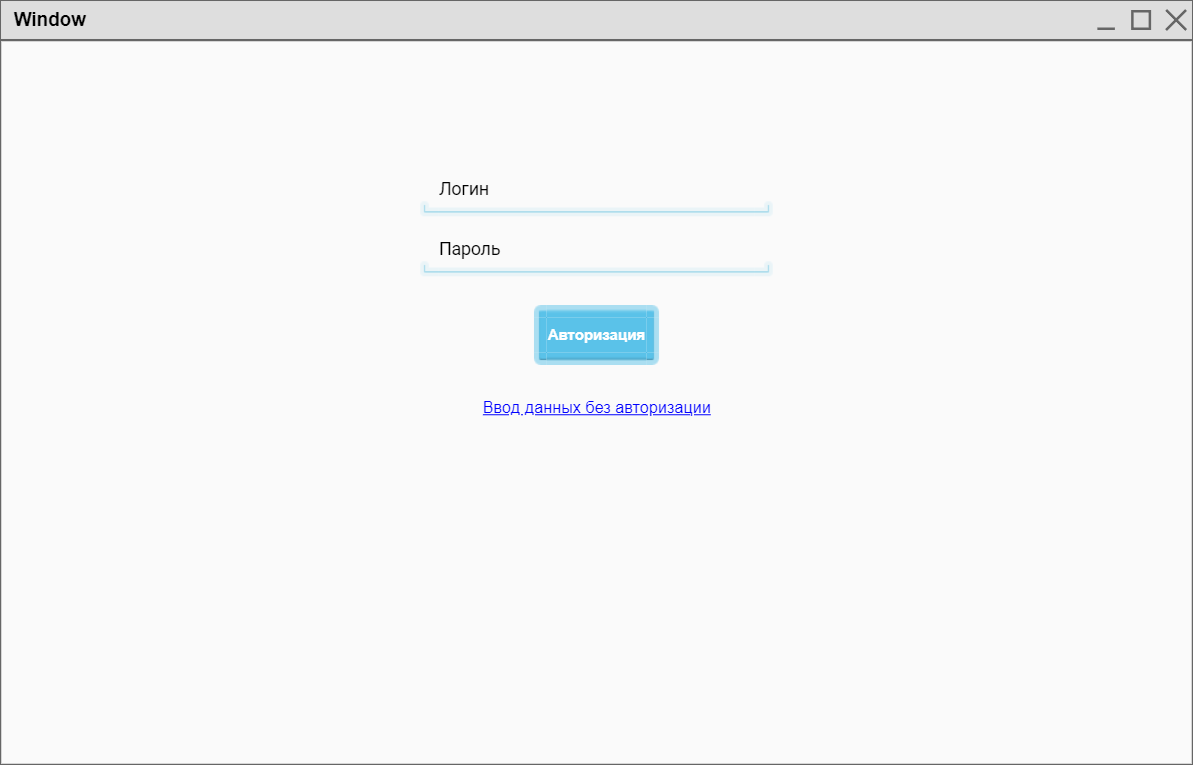
Функция «Подсчет граждан по определенным критериям» выполняет нахождение в базе данных записей, которые соблюдают по выбранному критерию пользователя, найденные записи суммируются и выводится результат суммирования на экран пользователю.

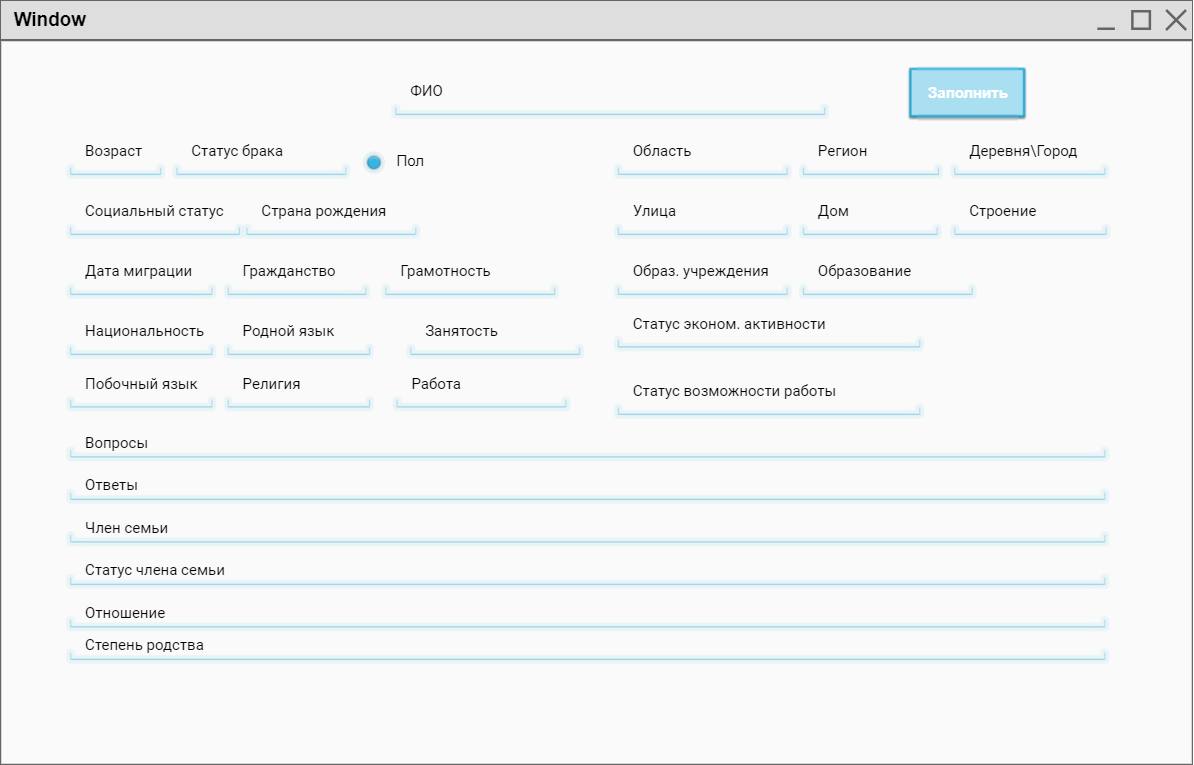
Функция «Поиск данных гражданина по ФИО» выполняет поиск данных гражданина по введенному в поле ФИО гражданина на наличие записей по этому ФИО гражданина. Если запись есть, то найденные записи выводятся в виде таблицы, если запись не была найдена, то таблица будет пустая.

Функция «Вывод других данных» выполняет вывод всех данных, которые существуют в базе данных, таблицы, которую выбрал пользователь. Если данные существуют, то выводятся в виде таблицы и выводятся на экран.

Функция «Сохранение выполненных действий» выполняет выгрузку в текстовый документ записи о выполненных действиях во время работы системы.

* 1. Прототип основных интерфейсов

  
Рисунок 15 – Прототип окна авторизации

  
Рисунок 16 – Прототип окна добавления данных гражданина

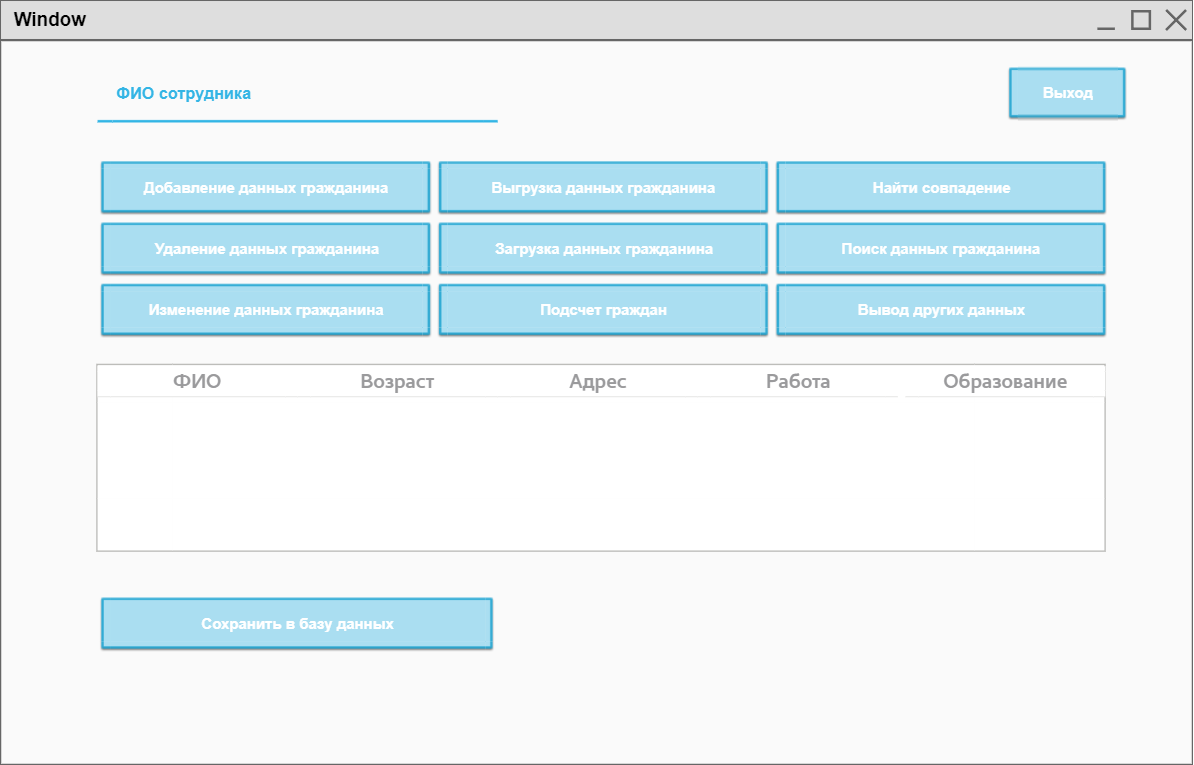
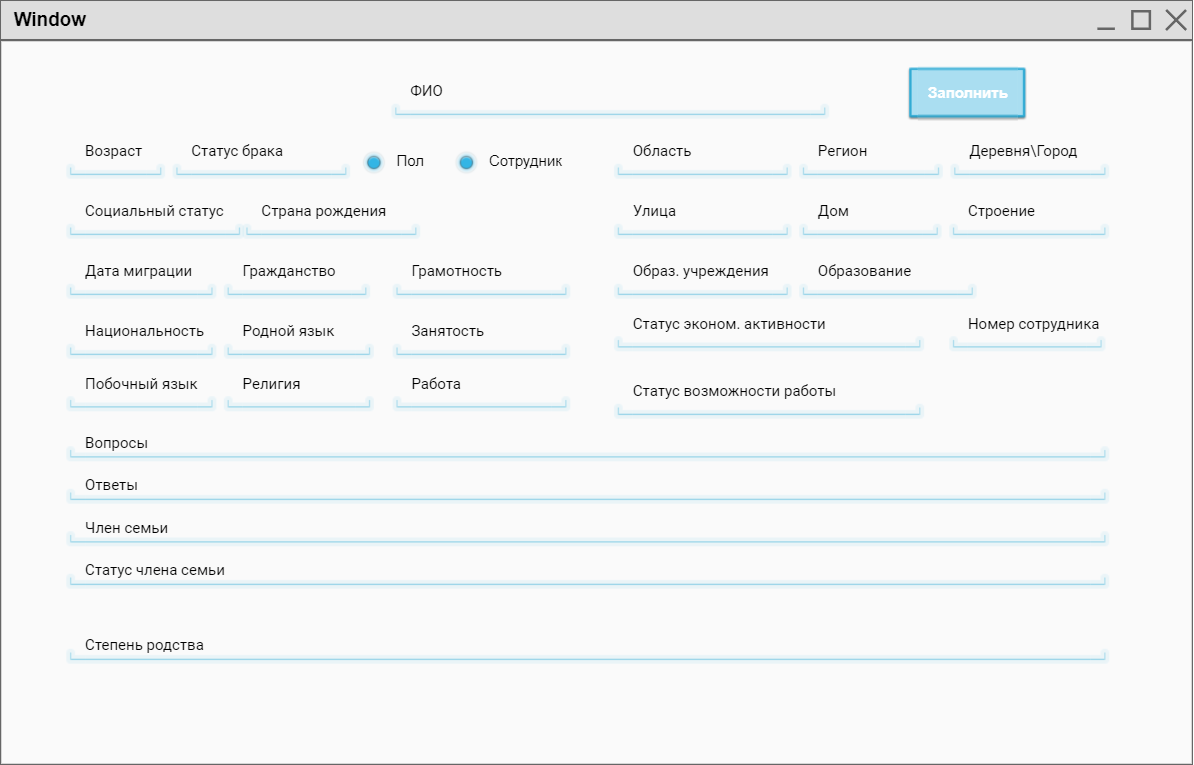
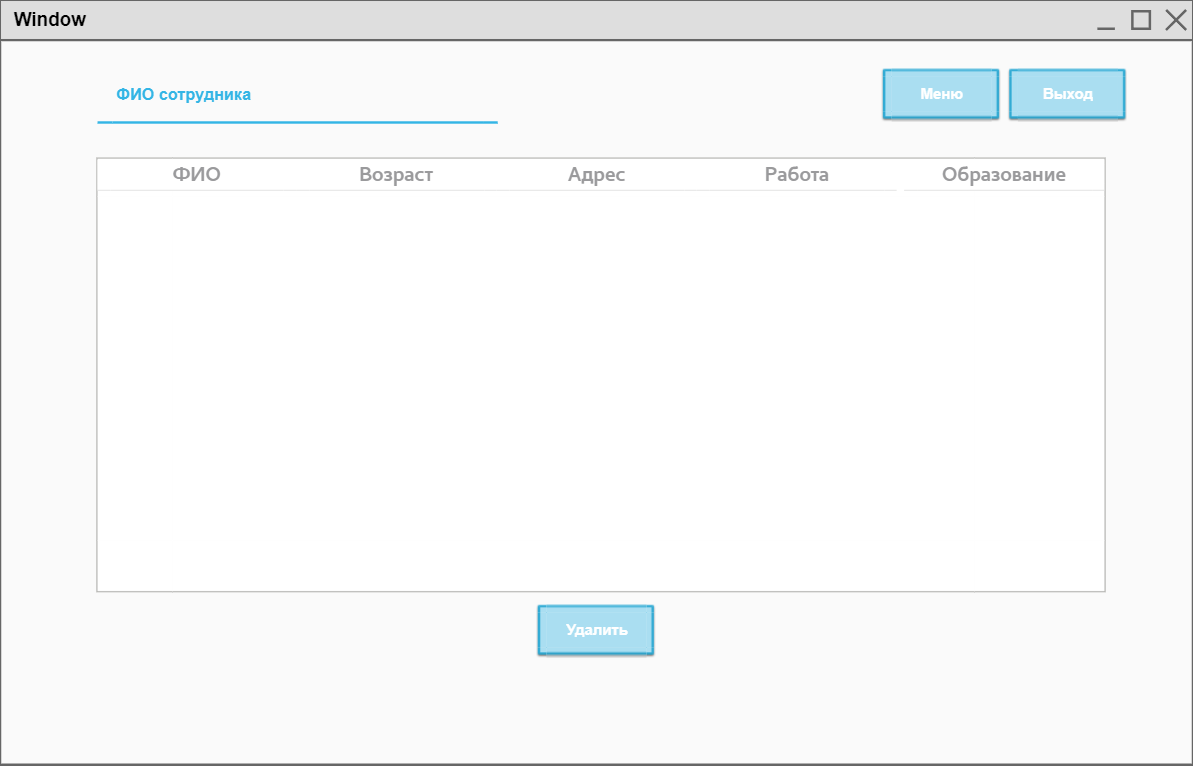
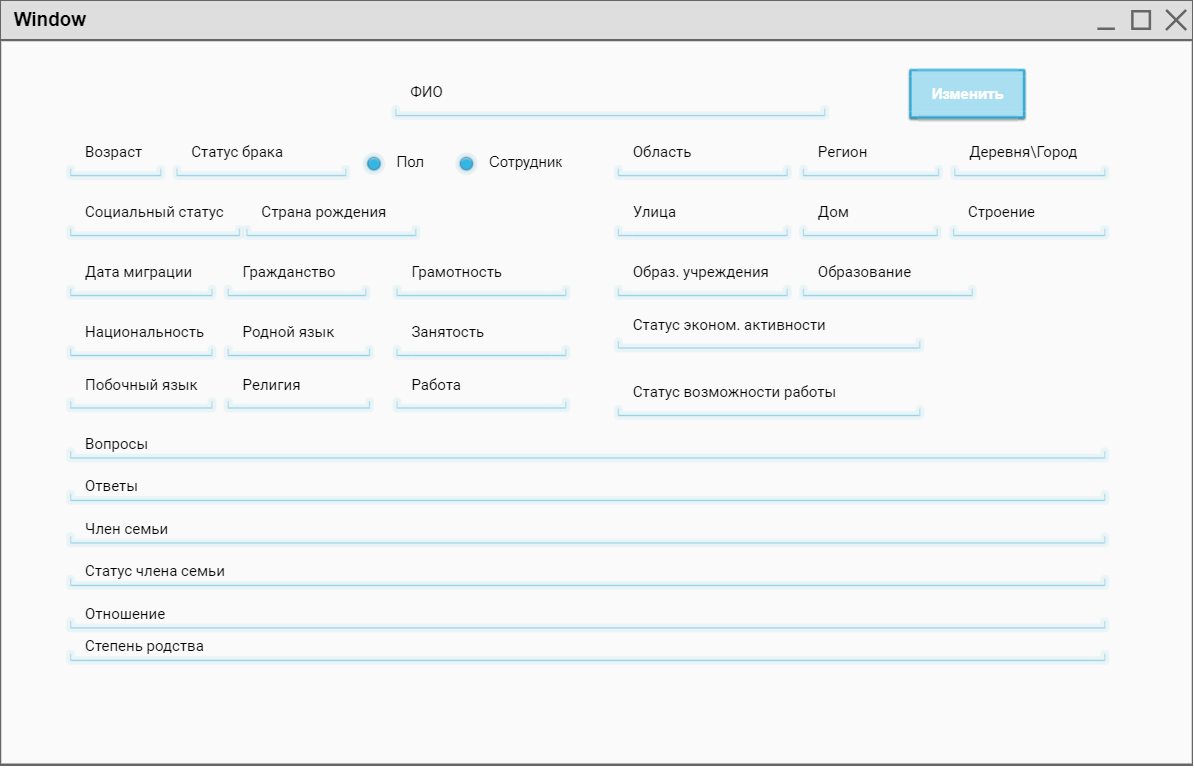
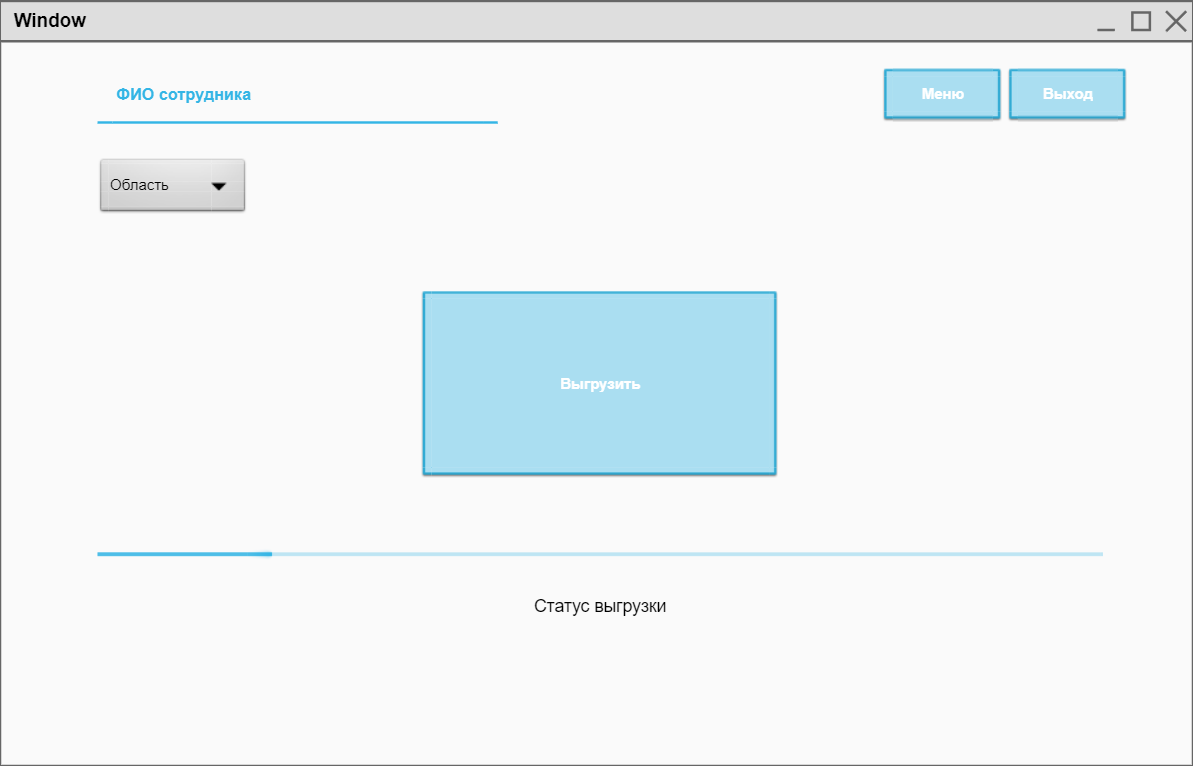
  
Рисунок 17 – Прототип окна меню сотрудника

  
Рисунок 18 – Прототип окна меню администратора

  
Рисунок 19 - Прототип окна добавления данных гражданина сотрудником

  
Рисунок 20 – Прототип окна удаления данных гражданина

  
Рисунок 21 – Прототип окна изменения данных

  
Рисунок 22 – Прототип окна выгрузки данных граждан

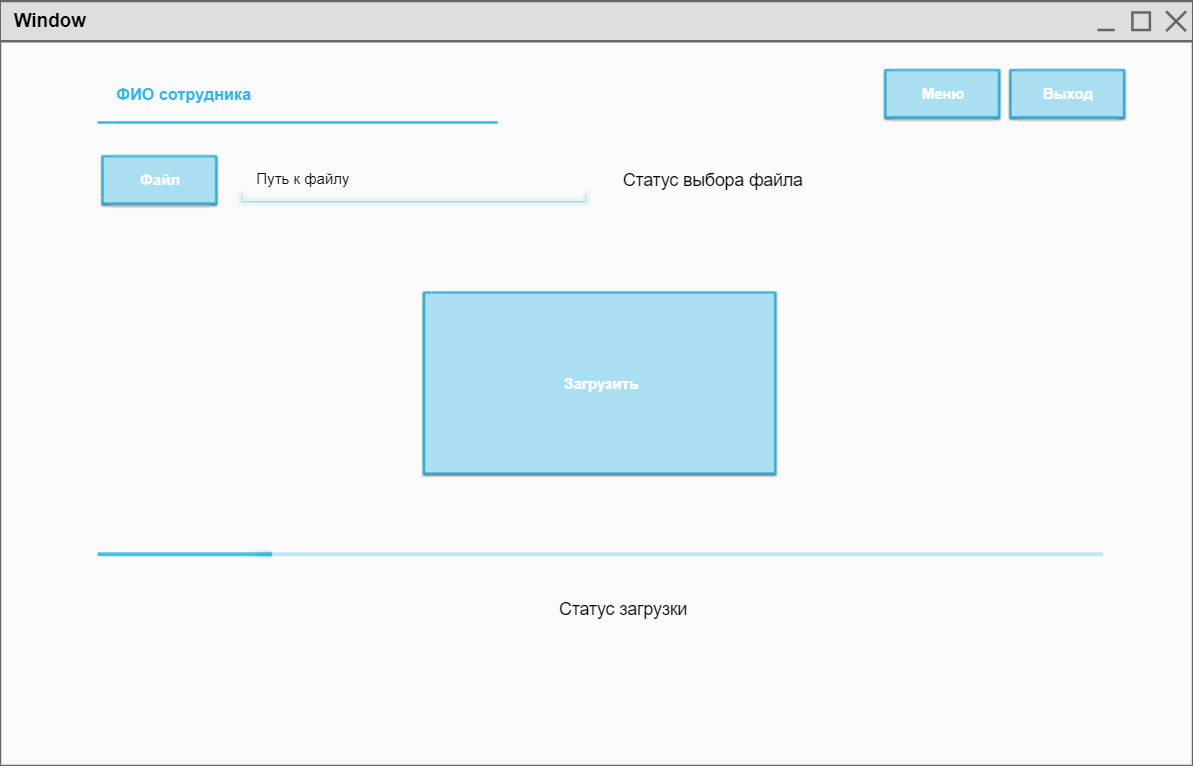
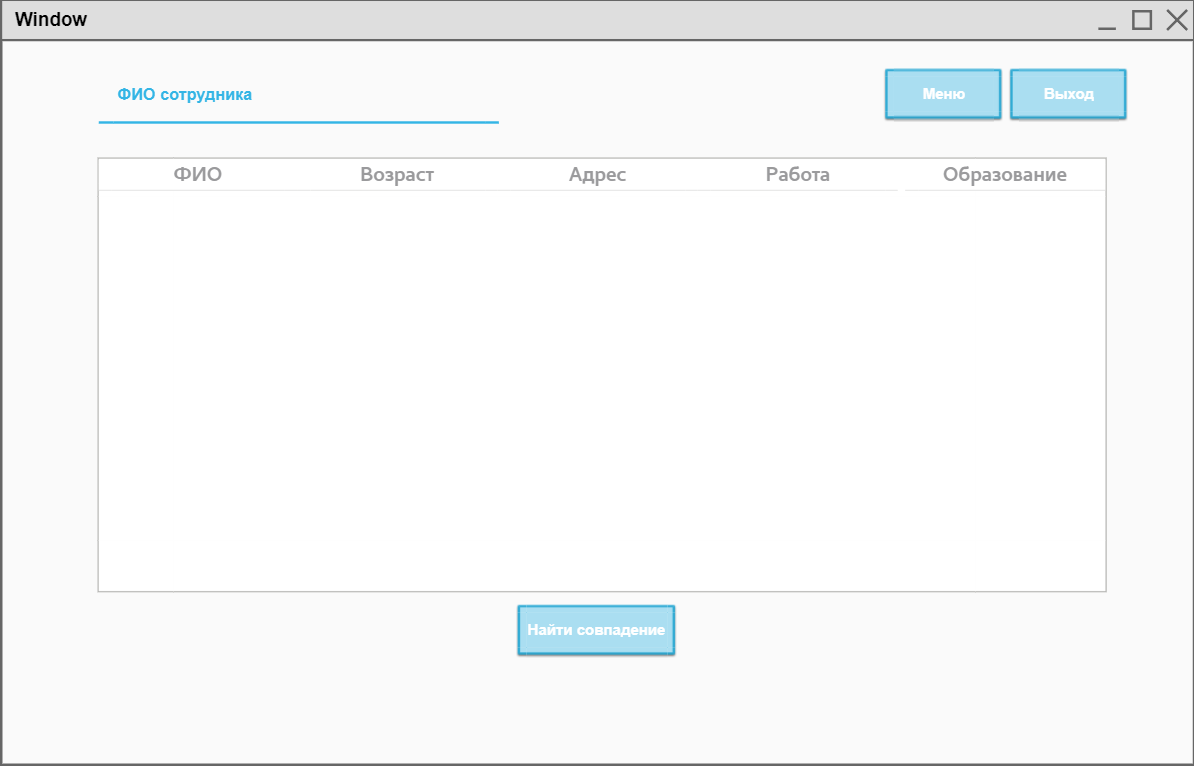
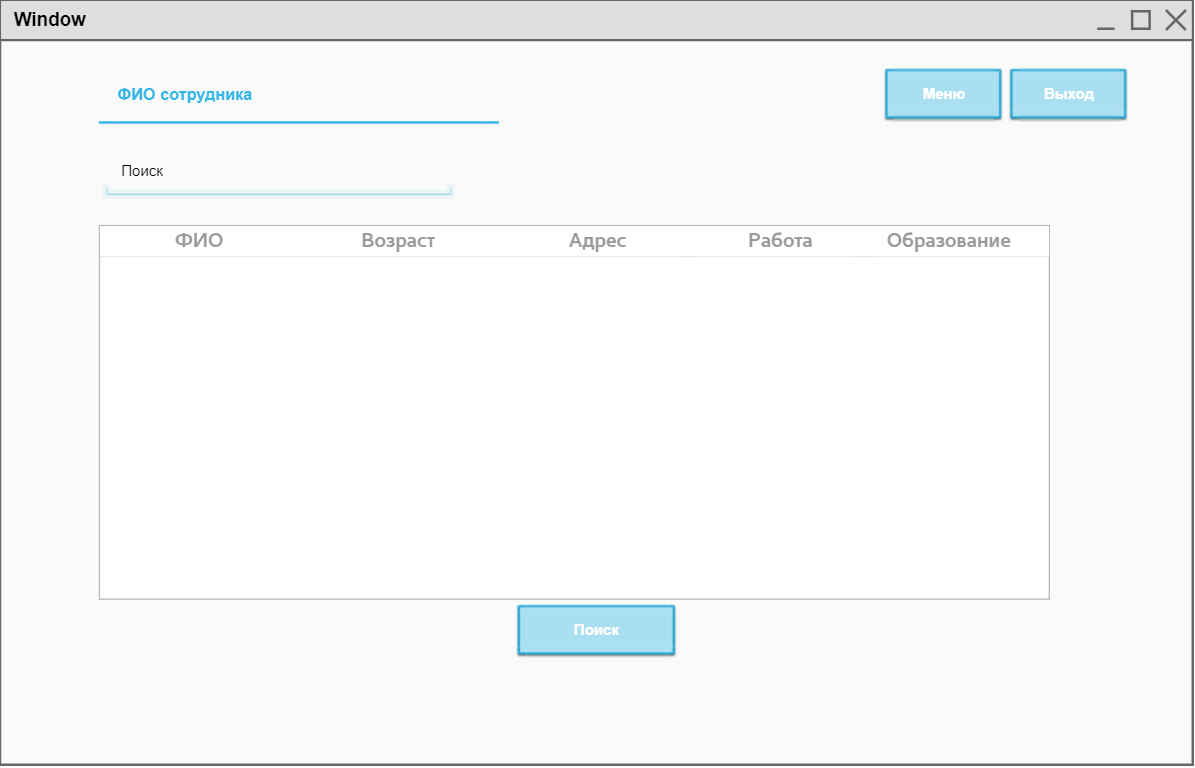
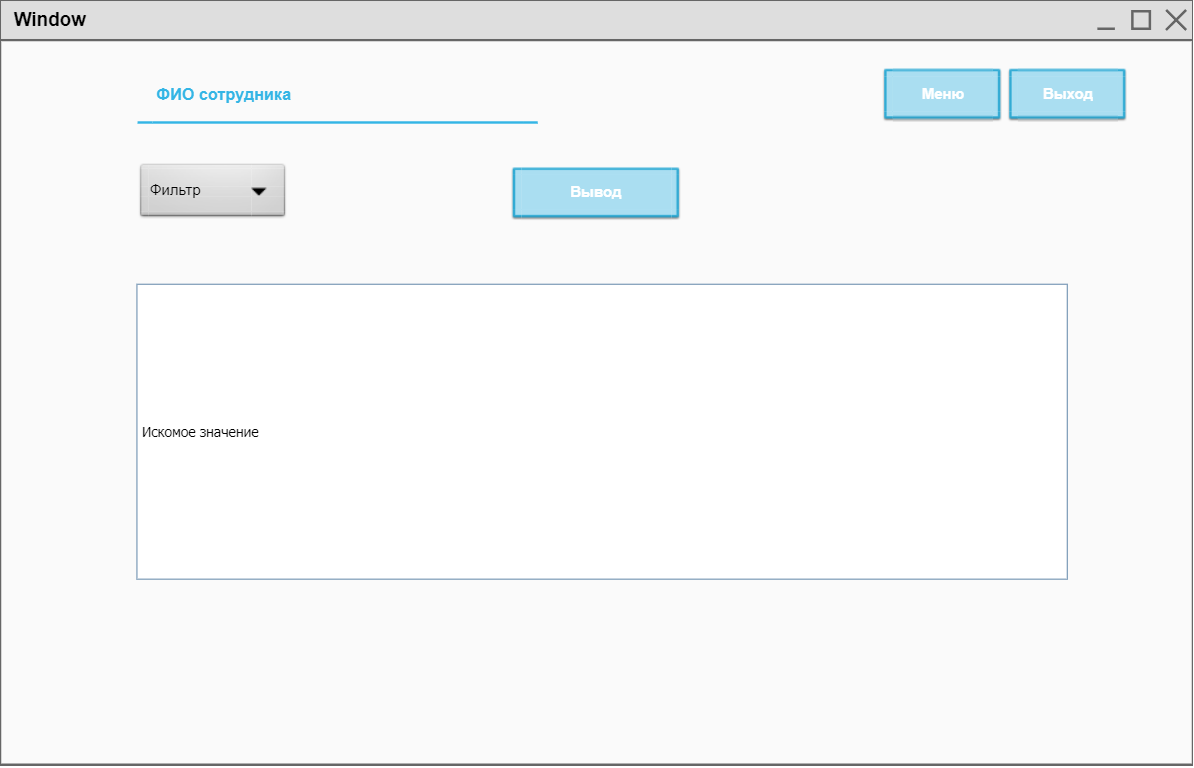
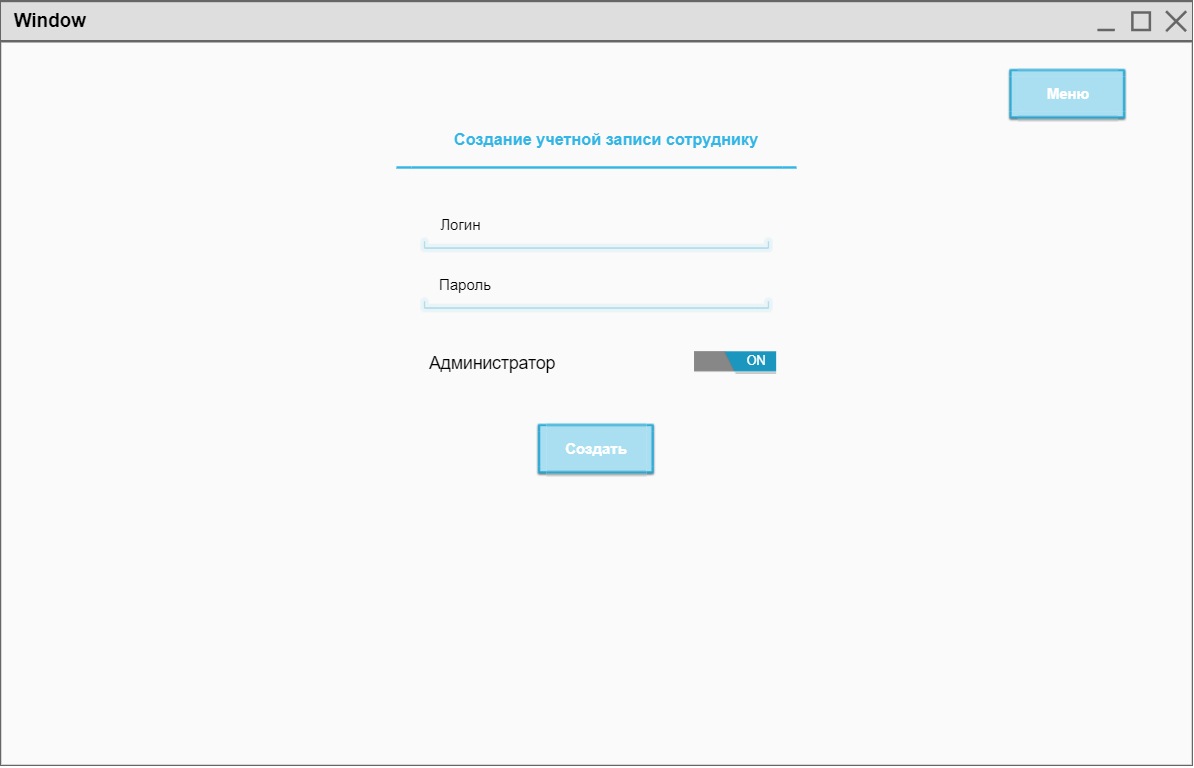
  
Рисунок 23 – Прототип окна загрузки данных граждан

  
Рисунок 24 – Прототип окна подсчета граждан по определенным критериям

  
Рисунок 25 – Прототип окна нахождения идентичных данных

  
Рисунок 26 – Прототип окна поиска данных граждан

  
Рисунок 27 – Прототип окна вывода остальных данных

  
Рисунок 28 – Прототип окна создания учетной записи пользователя

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были разработаны:

1. Требования функций для системы

В требованиях описывается, что должна выполнять функция, какие данные получает и выводит. Появились такие функции, как:

1. авторизация пользователей
2. добавление данных гражданина;
3. удаление данных гражданина;
4. изменение данных гражданина;
5. выход из учетной записи пользователя;
6. выгрузка данных граждан;
7. загрузка данных граждан;
8. поиск совпадения данных в таблицах;
9. подсчет данных по определенным критериям;
10. поиск данных гражданина по ФИО;
11. вывод других данных, записанные в базу данных системы;
12. сохранение выполненных действий во время работы системы;
13. создание учетных записей.
14. Диаграмма состояний для каждой функции

Данные диаграммы схематично показывают, как должна работать система и функции системы, когда приложение запущено.

1. Диаграмма вариантов использования

Данная диаграмма показывает возможности пользователей взаимодействия с данной системой. А также показывает доступ пользователей к функциям системы, которыми могут воспользоваться.

1. Прототип интерфейсов системы

Изображения окон системы, которые отображают возможный дизайн системы и функций.

1. Схема данных для элементов базы данных системы

Схема данных отображает структуру данных, по которой созданы элементы базы данных и какие данные будет содержать.

1. Инструкция пользователя

Данная инструкция показывает пошаговую работу с системой, которую может проделать пользователь.

1. Пользовательские сценарии

Описание системы или обзор системы для понимания пользователю.

1. Разработка информационной системы с функциями системы

Разработан минималистичный дизайн для системы, а также программно прописаны разработаны функции:

1. авторизацию
2. добавление данных гражданина;
3. удаление данных гражданина;
4. изменение данных гражданина;
5. выход из учетной записи пользователя;
6. выгрузка данных граждан;
7. загрузка данных граждан;
8. поиск совпадения данных в таблицах;
9. подсчет данных по определенным критериям;
10. поиск данных гражданина по ФИО;
11. вывод других данных, записанные в базу данных системы;
12. сохранение выполненных действий во время работы системы;
13. создание учетных записей;
14. возврат в основное меню пользователя.
15. тестирование информационной системы.

Проведены ручное тестирование, в ходе которого были выведены результаты работы системы.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. База данных. – Текст : электронный // Wikipedia : [сайт]. – URL: https://ru.wikipedia.org/wiki/База\_данных (дата обращения 28.03.23)
2. Образовательные учреждения. – Текст : электронный // Didacts : [сайт]. – URL: https://didacts.ru/termin/obrazovatelnye-uchrezhdenija.html (дата обращения 28.03.23)
3. Пользователь. – Текст : электронный // Wikipedia : [сайт]. – URL: https://ru.wikipedia.org/wiki/Пользователь (дата обращения 28.03.23)
4. Субъекты Российской Федерации. – Текст : электронный // Wikipedia : [сайт]. – URL: https://ru.wikipedia.org/wiki/Субъекты\_Российской\_Федерации (дата обращения 28.03.23)
5. Файл. – Текст : электронный // Wikipedia : [сайт]. – URL: https://ru.wikipedia.org/wiki/Файл (дата обращения 28.03.23)

ПРИЛОЖЕНИЕ А

Листинг программы

Код программы: окно авторизации

&НаКлиенте

Процедура Авторизация(Команда)

МассивПользователя = Новый Массив;

МассивПользователя = ВыборкаПользователей();

Параметр = Новый Структура("Логин",Логин);

Если Логин = МассивПользователя[0] И Пароль = МассивПользователя[1] Тогда

Если МассивПользователя[2] = Истина Тогда

ОткрытьФорму("ОбщаяФорма.МенюАдмина",Параметр);

СозданиеДокумента();

ИначеЕсли МассивПользователя[2] = Ложь Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",Параметр);

СозданиеДокумента();

КонецЕсли;

Иначе

СтатусАвторизации = "Ошибка авторизации!";

КонецЕсли;

Логин = "";

Пароль = "";

КонецПроцедуры

&НаСервере

Функция СозданиеДокумента()

ДокументАвторизации = Документы.Авторизация.СоздатьДокумент();

ДокументАвторизации.Дата = ТекущаяДата();

ДокументАвторизации.ИмяДокумента = "Авторизировался " + Логин;

ДокументАвторизации.Авторизировался = Логин;

ДокументАвторизации.Записать();

КонецФункции

&НаСервере

Функция ВыборкаПользователей()

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин);

ВыборкаЛогина = Пользователь.Логин;

ВыборкаПароль = Пользователь.Пароль;

ВыборкаРоли = Пользователь.Роль;

Массив = Новый Массив;

Массив.Вставить(0,ВыборкаЛогина);

Массив.Вставить(1,ВыборкаПароль);

Массив.Вставить(2,ВыборкаРоли);

Возврат Массив;

КонецФункции

&НаКлиенте

Процедура ВводДанных(Команда)

Параметр1 = Новый Структура("Логин","");

ОткрытьФорму("Документ.ДобавлениеДанныхГражданином.Форма.ФормаДобавленияДанныхГражданином",Параметр1);

КонецПроцедуры

Код программы: окно меню сотрудника

&НаКлиенте

Процедура Выход(Команда)

Закрыть();

КонецПроцедуры

&НаКлиенте

Процедура Добавление(Команда)

ПараметрЛогин = Новый Структура("Логин", Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.ДобавлениеДанныхГражданином.Форма.ФормаДобавленияДанныхГражданином",ПараметрЛогин);

КонецПроцедуры

&НаКлиенте

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Процедура Удаление(Команда)

ПараметрФИО = Новый Структура("ФИО",ФИОСотрудника);

ПараметрФИО.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.УдалениеДанных.Форма.ФормаУдаления",ПараметрФИО);

КонецПроцедуры

&НаКлиенте

Процедура Изменение(Команда)

ПараметрФИО = Новый Структура("ФИО",ФИОСотрудника);

ПараметрФИО.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.ИзменениеДанных.Форма.ФормаИзмененияДанных",ПараметрФИО);

КонецПроцедуры

&НаКлиенте

Процедура Выгрузка(Команда)

Параметр = Новый Структура("ФИО",ФИОСотрудника);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Обработка.СписокОбластей.Форма.ФормаВыгрузки", Параметр);

КонецПроцедуры

&НаКлиенте

Процедура Загрузка(Команда)

Параметр = Новый Структура("ФИО",ФИОСотрудника);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.ЗагрузкаДанных.Форма.ФормаЗагрузкаДанных", Параметр);

КонецПроцедуры

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

&НаКлиенте

Процедура НайтиСовпадение(Команда)

Параметр = Новый Структура("ФИО",ФИОСотрудника);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Обработка.СписокТаблиц.Форма.ФормаНайтиСовпадений", Параметр);

КонецПроцедуры

&НаКлиенте

Процедура Подсчет(Команда)

Параметр = Новый Структура("ФИО",ФИОСотрудника);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Обработка.ПодсчетГраждан.Форма.ФормаПодсчетГраждан",Параметр);

КонецПроцедуры

&НаКлиенте

Процедура ПоискДанных(Команда)

Параметр = Новый Структура("ФИО",ФИОСотрудника);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.ПоискГраждан.Форма.ФормаПоискаДанных",Параметр);

КонецПроцедуры

&НаКлиенте

Процедура ВыводДругихДанных(Команда)

Параметр = Новый Структура("ФИО",ФИОСотрудника);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Обработка.ВыводОстальныхДанных.Форма.ФормаВыводаДругихДанных",Параметр);

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

КонецПроцедуры

&НаКлиенте

Процедура СохранитьИсторию(Команда)

СохранитьИсториюНаСервере();

Сообщить("Сохранение завершено, проверьте C:\Users\Public\Documents\history.txt");

КонецПроцедуры

&НаСервере

Функция СохранитьИсториюНаСервере()

Запрос0 = Новый Запрос;

Запрос0.Текст = "ВЫБРАТЬ

| Авторизация.Дата КАК Дата,

| Авторизация.ИмяДокумента КАК ИмяДокумента,

| Авторизация.Авторизировался КАК Авторизировался

|ИЗ

| Документ.Авторизация КАК Авторизация";

Выборка0 = Запрос0.Выполнить().Выбрать();

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДобавлениеДанныхГражданином.Дата КАК Дата,

| ДобавлениеДанныхГражданином.ИмяДокумента КАК ИмяДокумента,

| ДобавлениеДанныхГражданином.Добавил КАК Добавил,

| ДобавлениеДанныхГражданином.СсылкаНаГражданина.ФИО КАК СсылкаНаГражданинаФИО

|ИЗ

| Документ.ДобавлениеДанныхГражданином КАК ДобавлениеДанныхГражданином";

Выборка = Запрос.Выполнить().Выбрать();

Запрос1 = Новый Запрос;

Запрос1.Текст = "ВЫБРАТЬ

| ВыгрузкаДанных.Дата КАК Дата,

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ВыгрузкаДанных.ИмяДокумента КАК ИмяДокумента,

| ВыгрузкаДанных.Выгрузил КАК Выгрузил,

| ВыгрузкаДанных.КоличествоЗаписей КАК КоличествоЗаписей

|ИЗ

| Документ.ВыгрузкаДанных КАК ВыгрузкаДанных";

Выборка1 = Запрос1.Выполнить().Выбрать();

Запрос2 = Новый Запрос;

Запрос2.Текст = "ВЫБРАТЬ

| УдалениеДанных.Дата КАК Дата,

| УдалениеДанных.ИмяДокумента КАК ИмяДокумента,

| УдалениеДанных.Удалил КАК Удалил,

| УдалениеДанных.Гражданин КАК Гражданин

|ИЗ

| Документ.УдалениеДанных КАК УдалениеДанных";

Выборка2 = Запрос2.Выполнить().Выбрать();

Запрос3 = Новый Запрос;

Запрос3.Текст = "ВЫБРАТЬ

| ИзменениеДанных.Дата КАК Дата,

| ИзменениеДанных.ИмяДокумента КАК ИмяДокумента,

| ИзменениеДанных.Изменил КАК Изменил,

| ИзменениеДанных.Гражданин.ФИО КАК ГражданинФИО

|ИЗ

| Документ.ИзменениеДанных КАК ИзменениеДанных";

Выборка3 = Запрос3.Выполнить().Выбрать();

Запрос4 = Новый Запрос;

Запрос4.Текст = "ВЫБРАТЬ

| ИзменениеДанных.Дата КАК Дата,

| ИзменениеДанных.ИмяДокумента КАК ИмяДокумента,

| ИзменениеДанных.Изменил КАК Изменил,

| ИзменениеДанных.Гражданин.ФИО КАК ГражданинФИО

|ИЗ

| Документ.ИзменениеДанных КАК ИзменениеДанных";

Выборка4 = Запрос4.Выполнить().Выбрать();

Запрос5 = Новый Запрос;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Запрос5.Текст = "ВЫБРАТЬ

| НайтиСовпадение.Дата КАК Дата,

| НайтиСовпадение.ИмяДокумента КАК ИмяДокумента,

| НайтиСовпадение.Искал КАК Искал

|ИЗ

| Документ.НайтиСовпадение КАК НайтиСовпадение";

Выборка5 = Запрос5.Выполнить().Выбрать();

Запрос6 = Новый Запрос;

Запрос6.Текст = "ВЫБРАТЬ

| ЗагрузкаДанных.Дата КАК Дата,

| ЗагрузкаДанных.ИмяДокумента КАК ИмяДокумента,

| ЗагрузкаДанных.Загрузил КАК Загрузил,

| ЗагрузкаДанных.ЧислоЗаписей КАК ЧислоЗаписей

|ИЗ

| Документ.ЗагрузкаДанных КАК ЗагрузкаДанных";

Выборка6 = Запрос6.Выполнить().Выбрать();

Запрос7 = Новый Запрос;

Запрос7.Текст = "ВЫБРАТЬ

| ПодсчетГраждан.Дата КАК Дата,

| ПодсчетГраждан.ИмяДокумента КАК ИмяДокумента,

| ПодсчетГраждан.Подсчитал КАК Подсчитал,

| ПодсчетГраждан.КоличествоПодсчитанных КАК КоличествоПодсчитанных

|ИЗ

| Документ.ПодсчетГраждан КАК ПодсчетГраждан";

Выборка7 = Запрос7.Выполнить().Выбрать();

Запрос8 = Новый Запрос;

Запрос8.Текст = "ВЫБРАТЬ

| ПоискГраждан.Дата КАК Дата,

| ПоискГраждан.ИмяДокумента КАК ИмяДокумента,

| ПоискГраждан.Искал КАК Искал,

| ПоискГраждан.ПоискПо КАК ПоискПо

|ИЗ

| Документ.ПоискГраждан КАК ПоискГраждан";

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Выборка8 = Запрос8.Выполнить().Выбрать();

Запрос9 = Новый Запрос;

Запрос9.Текст = "ВЫБРАТЬ

| ВыводДругихДанных.Дата КАК Дата,

| ВыводДругихДанных.ИмяДокумента КАК ИмяДокумента,

| ВыводДругихДанных.Искал КАК Искал,

| ВыводДругихДанных.ФильтрПоиска КАК ФильтрПоиска

|ИЗ

| Документ.ВыводДругихДанных КАК ВыводДругихДанных";

Выборка9 = Запрос9.Выполнить().Выбрать();

Запрос10 = Новый Запрос;

Запрос10.Текст = "ВЫБРАТЬ

| СохранениеИстории.Дата КАК Дата,

| СохранениеИстории.ИмяДокумента КАК ИмяДокумента,

| СохранениеИстории.Сохранил КАК Сохранил

|ИЗ

| Документ.СохранениеИстории КАК СохранениеИстории";

Выборка10 = Запрос10.Выполнить().Выбрать();

Запрос11 = Новый Запрос;

Запрос11.Текст = "ВЫБРАТЬ

| СозданиеСотрудника.Дата КАК Дата,

| СозданиеСотрудника.ИмяДокумента КАК ИмяДокумента,

| СозданиеСотрудника.Создал КАК Создал,

| СозданиеСотрудника.ЛогинНовогоСотрудника КАК ЛогинНовогоСотрудника

|ИЗ

| Документ.СозданиеСотрудника КАК СозданиеСотрудника";

Выборка11 = Запрос11.Выполнить().Выбрать();

Текст = Новый ЗаписьТекста("C:\Users\Public\Documents\history.txt",КодировкаТекста.ANSI);

Пока Выборка0.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка0.Дата)+" "+Строка(Выборка0.ИмяДокумента)+

" "+Строка(Выборка0.Авторизировался));

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

КонецЦикла;

Пока Выборка.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка.Дата)+" "+Строка(Выборка.ИмяДокумента)+

" "+Строка(Выборка.Добавил)+" "+Строка(Выборка.СсылкаНаГражданинаФИО));

КонецЦикла;

Пока Выборка1.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка1.Дата)+" "+Строка(Выборка1.ИмяДокумента)+

" "+Строка(Выборка1.Выгрузил)+" "+Строка(Выборка1.КоличествоЗаписей));

КонецЦикла;

Пока Выборка2.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка2.Дата)+" "+Строка(Выборка2.ИмяДокумента)+

" "+Строка(Выборка2.Удалил)+" "+Строка(Выборка2.Гражданин));

КонецЦикла;

Пока Выборка3.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка3.Дата)+" "+Строка(Выборка3.ИмяДокумента)+

" "+Строка(Выборка3.Изменил)+" "+Строка(Выборка3.ГражданинФИО));

КонецЦикла;

Пока Выборка4.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка4.Дата)+" "+Строка(Выборка4.ИмяДокумента)+

" "+Строка(Выборка4.Изменил)+" "+Строка(Выборка4.ГражданинФИО));

КонецЦикла;

Пока Выборка5.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка5.Дата)+" "+Строка(Выборка5.ИмяДокумента)+

" "+Строка(Выборка5.Искал));

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

КонецЦикла;

Пока Выборка6.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка6.Дата)+" "+Строка(Выборка6.ИмяДокумента)+

" "+Строка(Выборка6.Загрузил)+" "+Строка(Выборка6.ЧислоЗаписей));

КонецЦикла;

Пока Выборка7.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка7.Дата)+" "+Строка(Выборка7.ИмяДокумента)+

" "+Строка(Выборка7.Подсчитал)+" "+Строка(Выборка7.КоличествоПодсчитанных));

КонецЦикла;

Пока Выборка8.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка8.Дата)+" "+Строка(Выборка8.ИмяДокумента)+

" "+Строка(Выборка8.Искал)+" "+Строка(Выборка8.ПоискПо));

КонецЦикла;

Пока Выборка9.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка9.Дата)+" "+Строка(Выборка9.ИмяДокумента)+

" "+Строка(Выборка9.Искал)+" "+Строка(Выборка9.ФильтрПоиска));

КонецЦикла;

Пока Выборка10.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка10.Дата)+" "+Строка(Выборка10.ИмяДокумента)+

" "+Строка(Выборка10.Сохранил));

КонецЦикла;

Пока Выборка11.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка11.Дата)+" "+Строка(Выборка11.ИмяДокумента)+

" "+Строка(Выборка11.Создал)+" "+Строка(Выборка11.ЛогинНовогоСотрудника));

КонецЦикла;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Текст.Закрыть();

ДокументСохранения = Документы.СохранениеИстории.СоздатьДокумент();

ДокументСохранения.Дата = ТекущаяДата();

ДокументСохранения.ИмяДокумента = "Сохранение записей";

ДокументСохранения.Сохранил = Логин;

ДокументСохранения.Записать();

КонецФункции

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Перем ЛогинПользователя, СотрудникСправочник;

ЛогинПользователя = Параметры.Логин;

Логин = ЛогинПользователя;

СотрудникСправочник = Справочники.Сотрудник.НайтиПоРеквизиту("Логин",ЛогинПользователя).Ссылка;

ФИОСотрудника = Справочники.ДанныеОГражданине.НайтиПоРеквизиту("Код\_СправочникаСотрудник",СотрудникСправочник).ФИО;

КонецПроцедуры

&НаКлиенте

Процедура ПриОткрытии(Отказ)

ЗаполнениеТаблицы();

КонецПроцедуры

&НаСервере

Процедура ЗаполнениеТаблицы()

Запрос = Новый Запрос();

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОГражданине.ФИО КАК ФИО,

| ДанныеОГражданине.Код\_СправочникаСотрудник КАК ДанныеСотрудника,

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ДанныеОГражданине.Код\_СправочникАдрес КАК ДанныеОбАдрес,

| ДанныеОГражданине.Код\_СправочникаПерсональныеДанные КАК ПерсональныеДанные,

| ДанныеОГражданине.Код\_СправочникаСоциальныеДанные КАК СоциальныеДанные,

| ДанныеОГражданине.Код\_СправочникаДанныеСемьи КАК ДанныеСемьи,

| ДанныеОГражданине.Код\_СправочнниаДанныеОСтране КАК ДанныеОСтране,

| ДанныеОГражданине.Код\_СправочникаНациональности КАК Национальности,

| ДанныеОГражданине.Код\_СправочникаВопросы КАК Вопросы

|ИЗ

| Справочник.ДанныеОГражданине КАК ДанныеОГражданине";

Результат = Запрос.Выполнить();

Таблица = Результат.Выгрузить(ОбходРезультатаЗапроса.Прямой);

ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(Таблица, "ДанныеГраждан", "Таблица1");

КонецПроцедуры

&НаСервере

Процедура ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(ТабВКоде, ТабРеквизит, ТабНаФорме)

НовыеРеквизиты = Новый Массив;

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыеРеквизиты.Добавить(

Новый РеквизитФормы(

Колонка.Имя, Колонка.ТипЗначения,

ТабРеквизит));

КонецЦикла;

ИзменитьРеквизиты(НовыеРеквизиты);

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

НовыйЭлемент = Элементы.Добавить(

ТабРеквизит + "\_" + Колонка.Имя, Тип("ПолеФормы"), Элементы[ТабНаФорме]);

НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;

НовыйЭлемент.ПутьКДанным = ТабРеквизит + "." + Колонка.Имя;

КонецЦикла;

ЗначениеВРеквизитФормы(ТабВКоде, ТабРеквизит);

КонецПроцедуры

Код программы: окно меню админа

&НаКлиенте

Процедура Выход(Команда)

Закрыть();

КонецПроцедуры

&НаКлиенте

Процедура Добавление(Команда)

ПараметрЛогин = Новый Структура("Логин", Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.ДобавлениеДанныхГражданином.Форма.ФормаДобавленияДанныхГражданином",ПараметрЛогин);

КонецПроцедуры

&НаКлиенте

Процедура Удаление(Команда)

ПараметрФИО = Новый Структура("ФИО",ФИОАдмина);

ПараметрФИО.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.УдалениеДанных.Форма.ФормаУдаления",ПараметрФИО);

КонецПроцедуры

&НаКлиенте

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Процедура Изменение(Команда)

ПараметрФИО = Новый Структура("ФИО",ФИОАдмина);

ПараметрФИО.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.ИзменениеДанных.Форма.ФормаИзмененияДанных",ПараметрФИО);

КонецПроцедуры

&НаКлиенте

Процедура Выгрузка(Команда)

Параметр = Новый Структура("ФИО",ФИОАдмина);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Обработка.СписокОбластей.Форма.ФормаВыгрузки", Параметр);

КонецПроцедуры

&НаКлиенте

Процедура Загрузка(Команда)

Параметр = Новый Структура("ФИО",ФИОАдмина);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.ЗагрузкаДанных.Форма.ФормаЗагрузкаДанных", Параметр);

КонецПроцедуры

&НаКлиенте

Процедура НайтиСовпадение(Команда)

Параметр = Новый Структура("ФИО",ФИОАдмина);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Обработка.СписокТаблиц.Форма.ФормаНайтиСовпадений", Параметр);

КонецПроцедуры

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

&НаКлиенте

Процедура Подсчет(Команда)

Параметр = Новый Структура("ФИО",ФИОАдмина);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Обработка.ПодсчетГраждан.Форма.ФормаПодсчетГраждан",Параметр);

КонецПроцедуры

&НаКлиенте

Процедура ПоискДанных(Команда)

Параметр = Новый Структура("ФИО",ФИОАдмина);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.ПоискГраждан.Форма.ФормаПоискаДанных",Параметр);

КонецПроцедуры

&НаКлиенте

Процедура ВыводДругихДанных(Команда)

Параметр = Новый Структура("ФИО",ФИОАдмина);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Обработка.ВыводОстальныхДанных.Форма.ФормаВыводаДругихДанных",Параметр);

КонецПроцедуры

&НаКлиенте

Процедура СохранитьИсторию(Команда)

СохранитьИсториюНаСервере();

Сообщить("Сохранение завершено, проверьте C:\Users\Public\Documents\history.txt");

КонецПроцедуры

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

&НаСервере

Функция СохранитьИсториюНаСервере()

Запрос0 = Новый Запрос;

Запрос0.Текст = "ВЫБРАТЬ

| Авторизация.Дата КАК Дата,

| Авторизация.ИмяДокумента КАК ИмяДокумента,

| Авторизация.Авторизировался КАК Авторизировался

|ИЗ

| Документ.Авторизация КАК Авторизация";

Выборка0 = Запрос0.Выполнить().Выбрать();

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДобавлениеДанныхГражданином.Дата КАК Дата,

| ДобавлениеДанныхГражданином.ИмяДокумента КАК ИмяДокумента,

| ДобавлениеДанныхГражданином.Добавил КАК Добавил,

| ДобавлениеДанныхГражданином.СсылкаНаГражданина.ФИО КАК СсылкаНаГражданинаФИО

|ИЗ

| Документ.ДобавлениеДанныхГражданином КАК ДобавлениеДанныхГражданином";

Выборка = Запрос.Выполнить().Выбрать();

Запрос1 = Новый Запрос;

Запрос1.Текст = "ВЫБРАТЬ

| ВыгрузкаДанных.Дата КАК Дата,

| ВыгрузкаДанных.ИмяДокумента КАК ИмяДокумента,

| ВыгрузкаДанных.Выгрузил КАК Выгрузил,

| ВыгрузкаДанных.КоличествоЗаписей КАК КоличествоЗаписей

|ИЗ

| Документ.ВыгрузкаДанных КАК ВыгрузкаДанных";

Выборка1 = Запрос1.Выполнить().Выбрать();

Запрос2 = Новый Запрос;

Запрос2.Текст = "ВЫБРАТЬ

| УдалениеДанных.Дата КАК Дата,

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| УдалениеДанных.ИмяДокумента КАК ИмяДокумента,

| УдалениеДанных.Удалил КАК Удалил,

| УдалениеДанных.Гражданин КАК Гражданин

|ИЗ

| Документ.УдалениеДанных КАК УдалениеДанных";

Выборка2 = Запрос2.Выполнить().Выбрать();

Запрос3 = Новый Запрос;

Запрос3.Текст = "ВЫБРАТЬ

| ИзменениеДанных.Дата КАК Дата,

| ИзменениеДанных.ИмяДокумента КАК ИмяДокумента,

| ИзменениеДанных.Изменил КАК Изменил,

| ИзменениеДанных.Гражданин.ФИО КАК ГражданинФИО

|ИЗ

| Документ.ИзменениеДанных КАК ИзменениеДанных";

Выборка3 = Запрос3.Выполнить().Выбрать();

Запрос4 = Новый Запрос;

Запрос4.Текст = "ВЫБРАТЬ

| ИзменениеДанных.Дата КАК Дата,

| ИзменениеДанных.ИмяДокумента КАК ИмяДокумента,

| ИзменениеДанных.Изменил КАК Изменил,

| ИзменениеДанных.Гражданин.ФИО КАК ГражданинФИО

|ИЗ

| Документ.ИзменениеДанных КАК ИзменениеДанных";

Выборка4 = Запрос4.Выполнить().Выбрать();

Запрос5 = Новый Запрос;

Запрос5.Текст = "ВЫБРАТЬ

| НайтиСовпадение.Дата КАК Дата,

| НайтиСовпадение.ИмяДокумента КАК ИмяДокумента,

| НайтиСовпадение.Искал КАК Искал

|ИЗ

| Документ.НайтиСовпадение КАК НайтиСовпадение";

Выборка5 = Запрос5.Выполнить().Выбрать();

Запрос6 = Новый Запрос;

Запрос6.Текст = "ВЫБРАТЬ

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ЗагрузкаДанных.Дата КАК Дата,

| ЗагрузкаДанных.ИмяДокумента КАК ИмяДокумента,

| ЗагрузкаДанных.Загрузил КАК Загрузил,

| ЗагрузкаДанных.ЧислоЗаписей КАК ЧислоЗаписей

|ИЗ

| Документ.ЗагрузкаДанных КАК ЗагрузкаДанных";

Выборка6 = Запрос6.Выполнить().Выбрать();

Запрос7 = Новый Запрос;

Запрос7.Текст = "ВЫБРАТЬ

| ПодсчетГраждан.Дата КАК Дата,

| ПодсчетГраждан.ИмяДокумента КАК ИмяДокумента,

| ПодсчетГраждан.Подсчитал КАК Подсчитал,

| ПодсчетГраждан.КоличествоПодсчитанных КАК КоличествоПодсчитанных

|ИЗ

| Документ.ПодсчетГраждан КАК ПодсчетГраждан";

Выборка7 = Запрос7.Выполнить().Выбрать();

Запрос8 = Новый Запрос;

Запрос8.Текст = "ВЫБРАТЬ

| ПоискГраждан.Дата КАК Дата,

| ПоискГраждан.ИмяДокумента КАК ИмяДокумента,

| ПоискГраждан.Искал КАК Искал,

| ПоискГраждан.ПоискПо КАК ПоискПо

|ИЗ

| Документ.ПоискГраждан КАК ПоискГраждан";

Выборка8 = Запрос8.Выполнить().Выбрать();

Запрос9 = Новый Запрос;

Запрос9.Текст = "ВЫБРАТЬ

| ВыводДругихДанных.Дата КАК Дата,

| ВыводДругихДанных.ИмяДокумента КАК ИмяДокумента,

| ВыводДругихДанных.Искал КАК Искал,

| ВыводДругихДанных.ФильтрПоиска КАК ФильтрПоиска

|ИЗ

| Документ.ВыводДругихДанных КАК ВыводДругихДанных";

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Выборка9 = Запрос9.Выполнить().Выбрать();

Запрос10 = Новый Запрос;

Запрос10.Текст = "ВЫБРАТЬ

| СохранениеИстории.Дата КАК Дата,

| СохранениеИстории.ИмяДокумента КАК ИмяДокумента,

| СохранениеИстории.Сохранил КАК Сохранил

|ИЗ

| Документ.СохранениеИстории КАК СохранениеИстории";

Выборка10 = Запрос10.Выполнить().Выбрать();

Запрос11 = Новый Запрос;

Запрос11.Текст = "ВЫБРАТЬ

| СозданиеСотрудника.Дата КАК Дата,

| СозданиеСотрудника.ИмяДокумента КАК ИмяДокумента,

| СозданиеСотрудника.Создал КАК Создал,

| СозданиеСотрудника.ЛогинНовогоСотрудника КАК ЛогинНовогоСотрудника

|ИЗ

| Документ.СозданиеСотрудника КАК СозданиеСотрудника";

Выборка11 = Запрос11.Выполнить().Выбрать();

Текст = Новый ЗаписьТекста("C:\Users\Public\Documents\history.txt",КодировкаТекста.ANSI);

Пока Выборка0.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка0.Дата)+" "+Строка(Выборка0.ИмяДокумента)+

" "+Строка(Выборка0.Авторизировался));

КонецЦикла;

Пока Выборка.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка.Дата)+" "+Строка(Выборка.ИмяДокумента)+

" "+Строка(Выборка.Добавил)+" "+Строка(Выборка.СсылкаНаГражданинаФИО));

КонецЦикла;

Пока Выборка1.Следующий() Цикл

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Текст.ЗаписатьСтроку(Строка(Выборка1.Дата)+" "+Строка(Выборка1.ИмяДокумента)+

" "+Строка(Выборка1.Выгрузил)+" "+Строка(Выборка1.КоличествоЗаписей));

КонецЦикла;

Пока Выборка2.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка2.Дата)+" "+Строка(Выборка2.ИмяДокумента)+

" "+Строка(Выборка2.Удалил)+" "+Строка(Выборка2.Гражданин));

КонецЦикла;

Пока Выборка3.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка3.Дата)+" "+Строка(Выборка3.ИмяДокумента)+

" "+Строка(Выборка3.Изменил)+" "+Строка(Выборка3.ГражданинФИО));

КонецЦикла;

Пока Выборка4.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка4.Дата)+" "+Строка(Выборка4.ИмяДокумента)+

" "+Строка(Выборка4.Изменил)+" "+Строка(Выборка4.ГражданинФИО));

КонецЦикла;

Пока Выборка5.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка5.Дата)+" "+Строка(Выборка5.ИмяДокумента)+

" "+Строка(Выборка5.Искал));

КонецЦикла;

Пока Выборка6.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка6.Дата)+" "+Строка(Выборка6.ИмяДокумента)+

" "+Строка(Выборка6.Загрузил)+" "+Строка(Выборка6.ЧислоЗаписей));

КонецЦикла;

Пока Выборка7.Следующий() Цикл

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Текст.ЗаписатьСтроку(Строка(Выборка7.Дата)+" "+Строка(Выборка7.ИмяДокумента)+

" "+Строка(Выборка7.Подсчитал)+" "+Строка(Выборка7.КоличествоПодсчитанных));

КонецЦикла;

Пока Выборка8.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка8.Дата)+" "+Строка(Выборка8.ИмяДокумента)+

" "+Строка(Выборка8.Искал)+" "+Строка(Выборка8.ПоискПо));

КонецЦикла;

Пока Выборка9.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка9.Дата)+" "+Строка(Выборка9.ИмяДокумента)+

" "+Строка(Выборка9.Искал)+" "+Строка(Выборка9.ФильтрПоиска));

КонецЦикла;

Пока Выборка10.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка10.Дата)+" "+Строка(Выборка10.ИмяДокумента)+

" "+Строка(Выборка10.Сохранил));

КонецЦикла;

Пока Выборка11.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(Выборка11.Дата)+" "+Строка(Выборка11.ИмяДокумента)+

" "+Строка(Выборка11.Создал)+" "+Строка(Выборка11.ЛогинНовогоСотрудника));

КонецЦикла;

Текст.Закрыть();

ДокументСохранения = Документы.СохранениеИстории.СоздатьДокумент();

ДокументСохранения.Дата = ТекущаяДата();

ДокументСохранения.ИмяДокумента = "Сохранение записей";

ДокументСохранения.Сохранил = Логин;

ДокументСохранения.Записать();

КонецФункции

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Перем ЛогинПользователя, СотрудникСправочник;

ЛогинПользователя = Параметры.Логин;

Логин = ЛогинПользователя;

СотрудникСправочник = Справочники.Сотрудник.НайтиПоРеквизиту("Логин",ЛогинПользователя).Ссылка;

ФИОАдмина = Справочники.ДанныеОГражданине.НайтиПоРеквизиту("Код\_СправочникаСотрудник",СотрудникСправочник).ФИО;

КонецПроцедуры

&НаКлиенте

Процедура ПриОткрытии(Отказ)

ЗаполнениеТаблицы();

КонецПроцедуры

&НаСервере

Процедура ЗаполнениеТаблицы()

Запрос = Новый Запрос();

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОГражданине.ФИО КАК ФИО,

| ДанныеОГражданине.Код\_СправочникаСотрудник КАК Код\_СправочникаСотрудник,

| ДанныеОГражданине.Код\_СправочникАдрес КАК Код\_СправочникАдрес,

| ДанныеОГражданине.Код\_СправочникаПерсональныеДанные КАК Код\_СправочникаПерсональныеДанные,

| ДанныеОГражданине.Код\_СправочникаСоциальныеДанные КАК Код\_СправочникаСоциальныеДанные,

| ДанныеОГражданине.Код\_СправочникаДанныеСемьи КАК Код\_СправочникаДанныеСемьи,

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ДанныеОГражданине.Код\_СправочнниаДанныеОСтране КАК Код\_СправочнниаДанныеОСтране,

| ДанныеОГражданине.Код\_СправочникаНациональности КАК Код\_СправочникаНациональности,

| ДанныеОГражданине.Код\_СправочникаВопросы КАК Код\_СправочникаВопросы

|ИЗ

| Справочник.ДанныеОГражданине КАК ДанныеОГражданине";

Результат = Запрос.Выполнить();

Таблица = Результат.Выгрузить(ОбходРезультатаЗапроса.Прямой);

ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(Таблица, "ДанныеГраждан", "Таблица1");

КонецПроцедуры

&НаСервере

Процедура ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(ТабВКоде, ТабРеквизит, ТабНаФорме)

НовыеРеквизиты = Новый Массив;

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыеРеквизиты.Добавить(

Новый РеквизитФормы(

Колонка.Имя, Колонка.ТипЗначения,

ТабРеквизит));

КонецЦикла;

ИзменитьРеквизиты(НовыеРеквизиты);

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыйЭлемент = Элементы.Добавить(

ТабРеквизит + "\_" + Колонка.Имя, Тип("ПолеФормы"), Элементы[ТабНаФорме]);

НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;

НовыйЭлемент.ПутьКДанным = ТабРеквизит + "." + Колонка.Имя;

КонецЦикла;

ЗначениеВРеквизитФормы(ТабВКоде, ТабРеквизит);

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

КонецПроцедуры

&НаКлиенте

Процедура СозданиеСотрудника(Команда)

Параметр = Новый Структура("ФИО",ФИОАдмина);

Параметр.Вставить("Логин",Логин);

ЭтаФорма.Закрыть();

ОткрытьФорму("Документ.СозданиеСотрудника.Форма.ФормаСозданияСотрудника",Параметр);

КонецПроцедуры

Код программы: окно функции «Добавление данных гражданина»

&НаКлиенте

Процедура Сохранить(Команда)

Перем Счетчик;

Счетчик = РазмерСправочника() + 2;

Сообщить(Счетчик);

СоздатьЗапись(Счетчик);

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Логин <> "" Тогда

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

Закрыть(ЭтаФорма);

КонецЕсли;

Иначе

Закрыть(ЭтаФорма);

КонецЕсли;

КонецПроцедуры

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

&НаСервере

Функция СоздатьЗапись(ЧислоЗаписей)

Перем АдресСправочник,ВопросыСправочник,ПерсональныеДанныеСправочник,

СемьяСправочник,СтранаСправочник,РаботаСправочник,ОбразованиеСправочник,

СоциальныеДанныеСправочник,ГражданинСправочник,ЛогинДляДокумента;

АдресСправочник = Справочники.ДанныеОбАдресе.СоздатьЭлемент();

АдресСправочник.Код = ЧислоЗаписей;

АдресСправочник.Наименование = "Адрес " + ФИО;

АдресСправочник.Область = Область;

АдресСправочник.Регион = Регион;

АдресСправочник.Улица = Улица;

АдресСправочник.Дом = Дом;

АдресСправочник.Строение = Строение;

АдресСправочник.Записать();

ВопросыСправочник = Справочники.ВопросыДляГражданина.СоздатьЭлемент();

ВопросыСправочник.Код = ЧислоЗаписей;

ВопросыСправочник.Наименование = "Данные " + ФИО;

ВопросыСправочник.Вопросы = Вопросы;

ВопросыСправочник.Ответы = Ответы;

ВопросыСправочник.Записать();

ПерсональныеДанныеСправочник = Справочники.ПерсональныеДанные.СоздатьЭлемент();

ПерсональныеДанныеСправочник.Код = ЧислоЗаписей;

ПерсональныеДанныеСправочник.Наименование = "Данные " + ФИО;

ПерсональныеДанныеСправочник.Пол = Пол;

ПерсональныеДанныеСправочник.Возраст = Возраст;

ПерсональныеДанныеСправочник.БрачныйСтатус = БрачныйСтатус;

ПерсональныеДанныеСправочник.Записать();

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

НациональностьСправочник = Справочники.ДанныеОНациональность.СоздатьЭлемент();

НациональностьСправочник.Код = ЧислоЗаписей+1;

НациональностьСправочник.Наименование = "Данные " + ФИО;

НациональностьСправочник.Национальность = Национальность;

НациональностьСправочник.ПервыйЯзык = ПервыйЯзык;

НациональностьСправочник.Религия = Религия;

НациональностьСправочник.ДругиеЯзыки = ДругиеЯзыки;

НациональностьСправочник.Записать();

СемьяСправочник = Справочники.ДанныеОСемье.СоздатьЭлемент();

СемьяСправочник.Код = ЧислоЗаписей;

СемьяСправочник.Наименование = "Данные " + ФИО;

СемьяСправочник.ЧленСемьи = ЧленСемьи;

СемьяСправочник.СтатусЧленаСемьи = СтатусЧленаСемьи;

СемьяСправочник.Отношение = Отношение;

СемьяСправочник.СтепеньРодства = СтепеньРодства;

СемьяСправочник.Записать();

СтранаСправочник = Справочники.ДанныеОСтране.СоздатьЭлемент();

СтранаСправочник.Код = ЧислоЗаписей;

СтранаСправочник.Наименование = "Данные " + ФИО;

СтранаСправочник.СтранаРождения = СтранаРождения;

СтранаСправочник.ДатаМиграции = ДатаМиграции;

СтранаСправочник.Гражданство = Гражданство;

СтранаСправочник.Записать();

РаботаСправочник = Справочники.ДанныеОРаботе.СоздатьЭлемент();

РаботаСправочник.Код = ЧислоЗаписей;

РаботаСправочник.Наименование = "Данные " + ФИО;

РаботаСправочник.СтатусЭкономическойАктичности = СтатусЭкономическойАктивности;

РаботаСправочник.РодЗанятий = РодЗанятий;

РаботаСправочник.Работа = Работа;

РаботаСправочник.СтатусВозможнойРаботоспособности = СтатусВозможнойРаботоспособности;

РаботаСправочник.Записать();

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

ОбразованиеСправочник = Справочники.ДанныеООбразовании.СоздатьЭлемент();

ОбразованиеСправочник.Код = ЧислоЗаписей;

ОбразованиеСправочник.Наименование = "Данные " + ФИО;

ОбразованиеСправочник.УровеньГрамотности = УровеньГрамотности;

ОбразованиеСправочник.УровеньОбразования = УровеньОбразования;

ОбразованиеСправочник.ОбразовательныеУчреждения = ОбразовательныеУчреждения;

ОбразованиеСправочник.Записать();

СоциальныеДанныеСправочник = Справочники.СоциальныеДанные.СоздатьЭлемент();

СоциальныеДанныеСправочник.Код = ЧислоЗаписей;

СоциальныеДанныеСправочник.Наименование = "Данные " + ФИО;

СоциальныеДанныеСправочник.СоциальныйСтатус = СоциальныйСтатус;

СоциальныеДанныеСправочник.Код\_СправочникаОбразование = Справочники.ДанныеООбразовании.НайтиПоКоду(ЧислоЗаписей).Ссылка;

СоциальныеДанныеСправочник.Код\_СправочникаРабота = Справочники.ДанныеОРаботе.НайтиПоКоду(ЧислоЗаписей).Ссылка;

СоциальныеДанныеСправочник.Записать();

ГражданинСправочник = Справочники.ДанныеОГражданине.СоздатьЭлемент();

ГражданинСправочник.Код = ЧислоЗаписей;

ГражданинСправочник.Наименование = "Данные " + ФИО;

ГражданинСправочник.ФИО = ФИО;

ГражданинСправочник.Код\_СправочникАдрес = Справочники.ДанныеОбАдресе.НайтиПоКоду(ЧислоЗаписей).Ссылка;

ГражданинСправочник.Код\_СправочникаПерсональныеДанные = Справочники.ПерсональныеДанные.НайтиПоКоду(ЧислоЗаписей).Ссылка;

ГражданинСправочник.Код\_СправочникаСоциальныеДанные = Справочники.СоциальныеДанные.НайтиПоКоду(ЧислоЗаписей).Ссылка;

ГражданинСправочник.Код\_СправочникаДанныеСемьи = Справочники.ДанныеОСемье.НайтиПоКоду(ЧислоЗаписей).Ссылка;

ГражданинСправочник.Код\_СправочнниаДанныеОСтране = Справочники.ДанныеОСтране.НайтиПоКоду(ЧислоЗаписей).Ссылка;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

ГражданинСправочник.Код\_СправочникаНациональности = Справочники.ДанныеОНациональность.НайтиПоКоду(ЧислоЗаписей).Ссылка;

ГражданинСправочник.Код\_СправочникаВопросы = Справочники.ВопросыДляГражданина.НайтиПоКоду(ЧислоЗаписей).Ссылка;

ГражданинСправочник.Записать();

ДобавлениеДокумент = Документы.ДобавлениеДанныхГражданином.СоздатьДокумент();

ДобавлениеДокумент.Дата = ТекущаяДата();

ДобавлениеДокумент.ИмяДокумента = "Добавление записи о " + ФИО;

ЛогинДляДокумента = Логин;

Если ЛогинДляДокумента = "" Тогда

ДобавлениеДокумент.Добавил = "Гражданин";

Иначе

ДобавлениеДокумент.Добавил = ЛогинДляДокумента;

КонецЕсли;

ДобавлениеДокумент.СсылкаНаГражданина = Справочники.ДанныеОГражданине.НайтиПоКоду(ЧислоЗаписей).Ссылка;

ДобавлениеДокумент.Записать();

КонецФункции

&НаСервере

Функция РазмерСправочника()

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| МАКСИМУМ(isnull(Справочник.Код,0)) КАК Код1

|ИЗ

| Справочник.ДанныеОГражданине КАК Справочник";

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Если Выборка.Следующий() Тогда

МаксимумКод = Выборка.Код1;

Возврат МаксимумКод;

КонецЕсли;

КонецФункции

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Логин = Параметры.Логин;

КонецПроцедуры

&НаКлиенте

Процедура Меню(Команда)

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Логин <> "" Тогда

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

Закрыть(ЭтаФорма);

КонецЕсли;

Иначе

Закрыть(ЭтаФорма);

КонецЕсли;

КонецПроцедуры

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

&НаСервере

Функция ПроверкаМеню()

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин).Роль;

Если Пользователь = Ложь Тогда

Возврат 0;

Иначе

Возврат 1;

КонецЕсли;

КонецФункции

Код программы: окно функции «Удаления данных»

&НаКлиенте

Процедура Удалить(Команда)

Если ВыборКода = 0 Тогда

Сообщить("Введите код");

ИначеЕсли ВыборКода <> 0 Тогда

Удаление(ВыборКода);

Сообщить("Запись удалена, список обновится при повторном открытии");

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция Удаление(Айди)

Попытка

ДокументУдаления = Документы.УдалениеДанных.СоздатьДокумент();

ДокументУдаления.Дата = ТекущаяДата();

ФИОГражданина = Справочники.ДанныеОГражданине.НайтиПоКоду(Айди).ФИО;

ДокументУдаления.ИмяДокумента = "Гражданин " + ФИОГражданина;

ДокументУдаления.Удалил = Логин;

ДокументУдаления.Гражданин = ФИОГражданина;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

ДокументУдаления.Записать();

Справочники.ДанныеОГражданине.НайтиПоКоду(Айди).ПолучитьОбъект().Удалить();

Справочники.ДанныеОбАдресе.НайтиПоКоду(Айди).ПолучитьОбъект().Удалить();

Справочники.ДанныеОНациональность.НайтиПоКоду(Айди).ПолучитьОбъект().Удалить();

Справочники.ДанныеООбразовании.НайтиПоКоду(Айди).ПолучитьОбъект().Удалить();

Справочники.ДанныеОРаботе.НайтиПоКоду(Айди).ПолучитьОбъект().Удалить();

Справочники.ДанныеОСемье.НайтиПоКоду(Айди).ПолучитьОбъект().Удалить();

Справочники.ДанныеОСтране.НайтиПоКоду(Айди).ПолучитьОбъект().Удалить();

Справочники.ПерсональныеДанные.НайтиПоКоду(Айди).ПолучитьОбъект().Удалить();

Справочники.СоциальныеДанные.НайтиПоКоду(Айди).ПолучитьОбъект().Удалить();

Справочники.ВопросыДляГражданина.НайтиПоКоду(Айди).ПолучитьОбъект().Удалить();

Исключение

КонецПопытки;

КонецФункции

&НаКлиенте

Процедура Меню(Команда)

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Закрыть(ЭтаФорма);

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПроверкаМеню()

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин).Роль;

Если Пользователь = Ложь Тогда

Возврат 0;

Иначе

Возврат 1;

КонецЕсли;

КонецФункции

&НаКлиенте

Процедура Выход(Команда)

Перем ЗакрытиеОкна;

Попытка

ЗакрытиеОкон = ПолучитьОкна();

Для Каждого Элемент Из ЗакрытиеОкон Цикл

Если Не Элемент.Основное Тогда

Форма = Элемент.ПолучитьСодержимое();

Форма.Закрыть();

КонецЕсли;

КонецЦикла;

Исключение

КонецПопытки;

КонецПроцедуры

&НаКлиенте

Процедура ПриОткрытии(Отказ)

ЗаполнениеТаблицы();

КонецПроцедуры

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

&НаСервере

Процедура ЗаполнениеТаблицы()

Запрос = Новый Запрос();

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОГражданине.Код КАК Код,

| ДанныеОГражданине.ФИО КАК ФИО,

| ДанныеОГражданине.Код\_СправочникаСотрудник КАК Код\_СправочникаСотрудник,

| ДанныеОГражданине.Код\_СправочникАдрес КАК Код\_СправочникАдрес,

| ДанныеОГражданине.Код\_СправочникаПерсональныеДанные КАК Код\_СправочникаПерсональныеДанные,

| ДанныеОГражданине.Код\_СправочникаСоциальныеДанные КАК Код\_СправочникаСоциальныеДанные,

| ДанныеОГражданине.Код\_СправочникаДанныеСемьи КАК Код\_СправочникаДанныеСемьи,

| ДанныеОГражданине.Код\_СправочнниаДанныеОСтране КАК Код\_СправочнниаДанныеОСтране,

| ДанныеОГражданине.Код\_СправочникаНациональности КАК Код\_СправочникаНациональности,

| ДанныеОГражданине.Код\_СправочникаВопросы КАК Код\_СправочникаВопросы

|ИЗ

| Справочник.ДанныеОГражданине КАК ДанныеОГражданине";

Результат = Запрос.Выполнить();

Таблица = Результат.Выгрузить(ОбходРезультатаЗапроса.Прямой);

ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(Таблица, "ДанныеГраждан", "Таблица1");

КонецПроцедуры

&НаСервере

Процедура ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(ТабВКоде, ТабРеквизит, ТабНаФорме)

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

НовыеРеквизиты = Новый Массив;

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыеРеквизиты.Добавить(

Новый РеквизитФормы(

Колонка.Имя, Колонка.ТипЗначения,

ТабРеквизит));

КонецЦикла;

ИзменитьРеквизиты(НовыеРеквизиты);

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыйЭлемент = Элементы.Добавить(

ТабРеквизит + "\_" + Колонка.Имя, Тип("ПолеФормы"), Элементы[ТабНаФорме]);

НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;

НовыйЭлемент.ПутьКДанным = ТабРеквизит + "." + Колонка.Имя;

КонецЦикла;

ЗначениеВРеквизитФормы(ТабВКоде, ТабРеквизит);

КонецПроцедуры

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Перем ФИОПользователя;

ФИОСотрудник = Параметры.ФИО;

Логин = Параметры.Логин;

КонецПроцедуры

Код программы: окно функции «Изменение данных»

&НаКлиенте

Процедура Изменить(Команда)

ИзменитьЗапись(ФИО1);

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

КонецПроцедуры

&НаСервере

Функция ИзменитьЗапись(ФИОГражданина)

Если ФИО <> "" Тогда

Справочник1 = Справочники.ДанныеОГражданине.НайтиПоРеквизиту("ФИО",ФИОГражданина).Ссылка;

Код\_Адрес = Справочник1.Код\_СправочникАдрес.Код;

Код\_Вопрос = Справочник1.Код\_СправочникаВопросы.Код;

Код\_Национальность = Справочник1.Код\_СправочникаНациональности.Код;

Код\_Семья = Справочник1.Код\_СправочникаДанныеСемьи.Код;

Код\_Страна = Справочник1.Код\_СправочнниаДанныеОСтране.Код;

Код\_СоцДанные = Справочник1.Код\_СправочникаСоциальныеДанные.Код;

Код\_ПерДанные = Справочник1.Код\_СправочникаПерсональныеДанные.Код;

Код\_Образование = Справочник1.Код\_СправочникаСоциальныеДанные.Код\_СправочникаОбразование.Код;

Код\_Работа = Справочник1.Код\_СправочникаСоциальныеДанные.Код\_СправочникаРабота.Код;

Справочник2 = Справочники.ДанныеОбАдресе.НайтиПоКоду(Код\_Адрес).Ссылка;

Справочник3 = Справочники.ВопросыДляГражданина.НайтиПоКоду(Код\_Вопрос).Ссылка;

Справочник4 = Справочники.ДанныеОНациональность.НайтиПоКоду(Код\_Национальность).Ссылка;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Справочник5 = Справочники.ДанныеОСемье.НайтиПоКоду(Код\_Семья).Ссылка;

Справочник6 = Справочники.ДанныеОСтране.НайтиПоКоду(Код\_Страна).Ссылка;

Справочник7 = Справочники.СоциальныеДанные.НайтиПоКоду(Код\_СоцДанные).Ссылка;

Справочник8 = Справочники.ПерсональныеДанные.НайтиПоКоду(Код\_ПерДанные).Ссылка;

Справочник9 = Справочники.ДанныеООбразовании.НайтиПоКоду(Код\_Образование).Ссылка;

Справочник10 = Справочники.ДанныеОРаботе.НайтиПоКоду(Код\_Работа).Ссылка;

Гражданин\_Объект = Справочник1.ПолучитьОбъект();

Если Справочник2.Пустая() Тогда

Адрес\_Объект = Справочники.ДанныеОбАдресе.СоздатьЭлемент();

Адрес\_Объект.Код = Справочник1.Код;

Иначе

Адрес\_Объект = Справочник2.ПолучитьОбъект();

КонецЕсли;

Если Справочник3.Пустая() Тогда

Вопросы\_Объект = Справочники.ВопросыДляГражданина.СоздатьЭлемент();

Вопросы\_Объект.Код = Справочник1.Код;

Иначе

Вопросы\_Объект = Справочник3.ПолучитьОбъект();

КонецЕсли;

Если Справочник4.Пустая() Тогда

Национальность\_Объект = Справочники.ДанныеОНациональность.СоздатьЭлемент();

Национальность\_Объект.Код = Справочник1.Код;

Иначе

Национальность\_Объект = Справочник4.ПолучитьОбъект();

КонецЕсли;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Если Справочник5.Пустая() Тогда

Семья\_Объект = Справочники.ДанныеОСемье.СоздатьЭлемент();

Семья\_Объект = Справочник1.Код;

Иначе

Семья\_Объект = Справочник5.ПолучитьОбъект();

КонецЕсли;

Если Справочник6.Пустая() Тогда

Страна\_Объект = Справочники.ДанныеОСтране.СоздатьЭлемент();

Страна\_Объект.Код = Справочник1.Код;

Иначе

Страна\_Объект = Справочник6.ПолучитьОбъект();

КонецЕсли;

Если Справочник7.Пустая() Тогда

СоцДанные\_Объект = Справочники.СоциальныеДанные.СоздатьЭлемент();

СоцДанные\_Объект.Код = Справочник1.Код;

Иначе

СоцДанные\_Объект = Справочник7.ПолучитьОбъект();

КонецЕсли;

Если Справочник8.Пустая() Тогда

ПерДанные\_Объект = Справочники.ПерсональныеДанные.СоздатьЭлемент();

ПерДанные\_Объект.Код = Справочник1.Код;

Иначе

ПерДанные\_Объект = Справочник8.ПолучитьОбъект();

КонецЕсли;

Если Справочник9.Пустая() Тогда

Образование\_Объект = Справочники.ДанныеООбразовании.СоздатьЭлемент();

Образование\_Объект.Код = Справочник1.Код;

Иначе

Образование\_Объект = Справочник9.ПолучитьОбъект();

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

КонецЕсли;

Если Справочник10.Пустая() Тогда

Работа\_Объект = Справочники.ДанныеОРаботе.СоздатьЭлемент();

Работа\_Объект.Код = Справочник1.Код;

Иначе

Работа\_Объект = Справочник10.ПолучитьОбъект();

КонецЕсли;

Гражданин\_Объект.ФИО = ФИО;

Адрес\_Объект.Область = Область;

Адрес\_Объект.Регион = Регион;

Адрес\_Объект.НаселенныйПункт = НаселенныйПункт;

Адрес\_Объект.Улица = Улица;

Адрес\_Объект.Дом = Дом;

Адрес\_Объект.Строение = Строение;

Вопросы\_Объект.Вопросы = Вопросы;

Вопросы\_Объект.Ответы = Ответы;

Национальность\_Объект.Национальность = Национальность;

Национальность\_Объект.ПервыйЯзык = ПервыйЯзык;

Национальность\_Объект.ДругиеЯзыки = ДругиеЯзыки;

Национальность\_Объект.Религия = Религия;

Семья\_Объект.ЧленСемьи = ЧленСемьи;

Семья\_Объект.СтатусЧленаСемьи = СтатусЧленаСемьи;

Семья\_Объект.Отношение = Отношение;

Семья\_Объект.СтепеньРодства = СтепеньРодства;

Страна\_Объект.Гражданство = Гражданство;

Страна\_Объект.ДатаМиграции = ДатаМиграции;

Страна\_Объект.СтранаРождения = СтранаРождения;

СоцДанные\_Объект.СоциальныйСтатус = СоциальныйСтатус;

ПерДанные\_Объект.Пол = Пол;

ПерДанные\_Объект.Возраст = Возраст;

ПерДанные\_Объект.БрачныйСтатус = БрачныйСтатус;

Образование\_Объект.УровеньГрамотности = УровеньГрамотности;

Образование\_Объект.УровеньОбразования = УровеньОбразования;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Образование\_Объект.ОбразовательныеУчреждения = ОбразовательныеУчреждения;

Работа\_Объект.СтатусЭкономическойАктичности = СтатусЭкономическойАктивности;

Работа\_Объект.СтатусВозможнойРаботоспособности = СтатусВозможнойРаботоспособности;

Работа\_Объект.РодЗанятий = РодЗанятий;

Работа\_Объект.Работа = Работа;

Работа\_Объект.ОбменДанными.Загрузка = Истина;

Работа\_Объект.Записать();

Образование\_Объект.ОбменДанными.Загрузка = Истина;

Образование\_Объект.Записать();

СоцДанные\_Объект.ОбменДанными.Загрузка = Истина;

СоцДанные\_Объект.Записать();

ПерДанные\_Объект.ОбменДанными.Загрузка = Истина;

ПерДанные\_Объект.Записать();

Адрес\_Объект.ОбменДанными.Загрузка = Истина;

Адрес\_Объект.Записать();

Вопросы\_Объект.ОбменДанными.Загрузка = Истина;

Вопросы\_Объект.Записать();

Национальность\_Объект.ОбменДанными.Загрузка = Истина;

Национальность\_Объект.Записать();

Страна\_Объект.ОбменДанными.Загрузка = Истина;

Страна\_Объект.Записать();

Семья\_Объект.ОбменДанными.Загрузка = Истина;

Семья\_Объект.Записать();

ДокументИзменения = Документы.ИзменениеДанных.СоздатьДокумент();

ДокументИзменения.Дата = ТекущаяДата();

ДокументИзменения.ИмяДокумента = "Изменения данных о " + ФИОГражданина;

ДокументИзменения.Изменил = Логин;

ДокументИзменения.Гражданин = Справочник1;

ДокументИзменения.Записать();

Гражданин\_Объект.Записать();

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Иначе

Сообщить("Заполните все поля");

КонецЕсли;

КонецФункции

&НаКлиенте

Процедура Меню(Команда)

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

Закрыть(ЭтаФорма);

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПроверкаМеню()

Пользователь = Справочники.Сотрудник.НайтиПоРеквизиту("Логин",Логин).Роль;

Если Пользователь = Ложь Тогда

Возврат 0;

Иначе

Возврат 1;

КонецЕсли;

КонецФункции

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Логин = Параметры.Логин;

КонецПроцедуры

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

&НаКлиенте

Процедура НайтиДанные(Команда)

Перем ФИОГражданина;

ФИОГражданина = ПоискФИО;

ФИО1 = ПоискФИО;

НайтиНаСервере(ФИОГражданина);

КонецПроцедуры

&НаСервере

Функция НайтиНаСервере(ФИОГражданина)

Справочник1 = Справочники.ДанныеОГражданине.НайтиПоРеквизиту("ФИО",ФИОГражданина).Ссылка;

Если Не Справочники.ДанныеОГражданине.НайтиПоРеквизиту("ФИО",ФИОГражданина).Пустая() Тогда

ФИО = Справочник1.ФИО;

Область = Справочник1.Код\_СправочникАдрес.Область;

Регион = Справочник1.Код\_СправочникАдрес.Регион;

НаселенныйПункт = Справочник1.Код\_СправочникАдрес.НаселенныйПункт;

Улица = Справочник1.Код\_СправочникАдрес.Улица;

Дом = Справочник1.Код\_СправочникАдрес.Дом;

Строение = Справочник1.Код\_СправочникАдрес.Строение;

СтранаРождения = Справочник1.Код\_СправочнниаДанныеОСтране.СтранаРождения;

ДатаМиграции = Справочник1.Код\_СправочнниаДанныеОСтране.ДатаМиграции;

Гражданство = Справочник1.Код\_СправочнниаДанныеОСтране.Гражданство;

Вопросы = Справочник1.Код\_СправочникаВопросы.Вопросы;

Ответы = Справочник1.Код\_СправочникаВопросы.Ответы;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

УровеньГрамотности = Справочник1.Код\_СправочникаСоциальныеДанные.Код\_СправочникаОбразование.УровеньГрамотности;

УровеньОбразования = Справочник1.Код\_СправочникаСоциальныеДанные.Код\_СправочникаОбразование.УровеньОбразования;

ОбразовательныеУчреждения = Справочник1.Код\_СправочникаСоциальныеДанные.Код\_СправочникаОбразование.ОбразовательныеУчреждения;

СоциальныйСтатус = Справочник1.Код\_СправочникаСоциальныеДанные.СоциальныйСтатус;

Пол = Справочник1.Код\_СправочникаПерсональныеДанные.Пол;

Возраст = Справочник1.Код\_СправочникаПерсональныеДанные.Возраст;

БрачныйСтатус = Справочник1.Код\_СправочникаПерсональныеДанные.БрачныйСтатус;

Национальность = Справочник1.Код\_СправочникаНациональности.Национальность;

ПервыйЯзык = Справочник1.Код\_СправочникаНациональности.ПервыйЯзык;

ДругиеЯзыки = Справочник1.Код\_СправочникаНациональности.ДругиеЯзыки;

Религия = Справочник1.Код\_СправочникаНациональности.Религия;

СтатусЭкономическойАктивности = Справочник1.Код\_СправочникаСоциальныеДанные.Код\_СправочникаРабота.СтатусЭкономическойАктичности;

РодЗанятий = Справочник1.Код\_СправочникаСоциальныеДанные.Код\_СправочникаРабота.РодЗанятий;

Работа = Справочник1.Код\_СправочникаСоциальныеДанные.Код\_СправочникаРабота.Работа;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

СтатусВозможнойРаботоспособности = Справочник1.Код\_СправочникаСоциальныеДанные.Код\_СправочникаРабота.СтатусВозможнойРаботоспособности;

ЧленСемьи = Справочник1.Код\_СправочникаДанныеСемьи.ЧленСемьи;

СтатусЧленаСемьи = Справочник1.Код\_СправочникаДанныеСемьи.СтатусЧленаСемьи;

Отношение = Справочник1.Код\_СправочникаДанныеСемьи.Отношение;

СтепеньРодства = Справочник1.Код\_СправочникаДанныеСемьи.СтепеньРодства;

Иначе

ФИО = "";

Область = "";

Регион = "";

НаселенныйПункт = "";

Улица = "";

Дом = "";

Строение = "";

СтранаРождения = "";

ДатаМиграции = Неопределено;

Гражданство = "";

Вопросы = "";

Ответы = "";

УровеньГрамотности = "";

УровеньОбразования = "";

ОбразовательныеУчреждения = "";

СоциальныйСтатус = "";

Возраст = "";

БрачныйСтатус = "";

Национальность = "";

ПервыйЯзык = "";

ДругиеЯзыки = "";

Религия = "";

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

СтатусЭкономическойАктивности = "";

РодЗанятий = "";

Работа = "";

СтатусВозможнойРаботоспособности = "";

ЧленСемьи = "";

СтатусЧленаСемьи = "";

Отношение = "";

СтепеньРодства = "";

СтатусПоиска = "Не найдено!";

КонецЕсли;

КонецФункции

&НаКлиенте

Процедура Выход(Команда)

Перем ЗакрытиеОкна;

Попытка

ЗакрытиеОкон = ПолучитьОкна();

Для Каждого Элемент Из ЗакрытиеОкон Цикл

Если Не Элемент.Основное Тогда

Форма = Элемент.ПолучитьСодержимое();

Форма.Закрыть();

КонецЕсли;

КонецЦикла;

Исключение

КонецПопытки;

КонецПроцедуры

Код программы: окно функции «Загрузка данных»

&НаКлиенте

Процедура Меню(Команда)

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

Закрыть(ЭтаФорма);

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПроверкаМеню()

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин).Роль;

Если Пользователь = Ложь Тогда

Возврат 0;

Иначе

Возврат 1;

КонецЕсли;

КонецФункции

&НаКлиенте

Процедура Выход(Команда)

Перем ЗакрытиеОкна;

Попытка

ЗакрытиеОкон = ПолучитьОкна();

Для Каждого Элемент Из ЗакрытиеОкон Цикл

Если Не Элемент.Основное Тогда

Форма = Элемент.ПолучитьСодержимое();

Форма.Закрыть();

КонецЕсли;

КонецЦикла;

Исключение

КонецПопытки;

КонецПроцедуры

&НаКлиенте

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Процедура ВыборФайла(Команда)

Перем Стр;

Стр = "";

СтандартнаяОбработка = Ложь;

Диалог = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Открытие);

Диалог.Заголовок = "Выберите файл";

Диалог.ПолноеИмяФайла = "";

Фильтр = "TXT (\*.txt)|\*.txt";

Диалог.Фильтр = Фильтр;

Диалог.МножественныйВыбор = Ложь;

Диалог.Каталог = "C:\Users\Public\Documents";

Если Диалог.Выбрать() Тогда

ПутьФайла = Диалог.ПолноеИмяФайла;

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

ФИОСотрудника = Параметры.ФИО;

Логин = Параметры.Логин;

КонецПроцедуры

&НаКлиенте

Процедура Загрузить(Команда)

Перем Счетчик;

Счетчик = РазмерСправочника() + 1;

Загрузка(Счетчик);

КонецПроцедуры

&НаСервере

Процедура Загрузка(Счетчик)

Перем Стр;

Стр = "";

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

МассивСтрок = Новый Массив;

ВыбранныйФайл = Новый Файл(ПутьФайла);

Если ВыбранныйФайл.Существует() Тогда

Текст = Новый ЧтениеТекста(ВыбранныйФайл.ПолноеИмя,КодировкаТекста.ANSI,Символы.ВК + Символы.ПС, Символы.ПС);

Стр = Текст.ПрочитатьСтроку();

Пока Стр <> Неопределено Цикл

МассивСтрок.Добавить(Стр);

Стр = Текст.ПрочитатьСтроку();

КонецЦикла;

Для Каждого ЭлементМассива Из МассивСтрок Цикл

КоличествоСтрок = КоличествоСтрок + 1;

СправочникГражданин = Справочники.ДанныеОГражданине.СоздатьЭлемент();

СправочникПерсДанные = Справочники.ПерсональныеДанные.СоздатьЭлемент();

СправочникСоцДанные = Справочники.СоциальныеДанные.СоздатьЭлемент();

СправочникОбразование = Справочники.ДанныеООбразовании.СоздатьЭлемент();

СправочникРабота = Справочники.ДанныеОРаботе.СоздатьЭлемент();

СправочникМиграции = Справочники.ДанныеОСтране.СоздатьЭлемент();

СправочникСемья = Справочники.ДанныеОСемье.СоздатьЭлемент();

СправочникНациональность = Справочники.ДанныеОНациональность.СоздатьЭлемент();

СправочникАдрес = Справочники.ДанныеОбАдресе.СоздатьЭлемент();

СправочникВопросы = Справочники.ВопросыДляГражданина.СоздатьЭлемент();

МассивРазбитойСтроки = Новый Массив;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

РазбитаяСтрока = СтрРазделить(ЭлементМассива,",");

Для Каждого ЧастьСтроки Из РазбитаяСтрока Цикл

МассивРазбитойСтроки.Добавить(ЧастьСтроки);

КонецЦикла;

ФИОГражданина = МассивРазбитойСтроки[0];

СправочникГражданин.ФИО = ФИОГражданина;

СправочникСоцДанные.Наименование = "Данные " + ФИОГражданина;

СправочникГражданин.Наименование = "Гражданин " + ФИОГражданина;

СправочникПерсДанные.Наименование = "Данные " + ФИОГражданина;

СправочникОбразование.Наименование = "Данные " + ФИОГражданина;

СправочникРабота.Наименование = "Данные " + ФИОГражданина;

СправочникМиграции.Наименование = "Данные " + ФИОГражданина;

СправочникСемья.Наименование = "Данные " + ФИОГражданина;

СправочникНациональность.Наименование = "Данные " + ФИОГражданина;

СправочникАдрес.Наименование = "Адрес " + ФИОГражданина;

СправочникВопросы.Наименование = "Данные " + ФИОГражданина;

СправочникГражданин.Код = Счетчик;

СправочникАдрес.Код = Счетчик;

СправочникВопросы.Код = Счетчик;

СправочникМиграции.Код = Счетчик;

СправочникНациональность.Код = Счетчик;

СправочникОбразование.Код = Счетчик;

СправочникПерсДанные.Код = Счетчик;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

СправочникРабота.Код = Счетчик;

СправочникСемья.Код = Счетчик;

СправочникСоцДанные.Код = Счетчик;

СправочникПерсДанные.Пол = МассивРазбитойСтроки[1];

СправочникПерсДанные.Возраст = МассивРазбитойСтроки[2];

СправочникСоцДанные.СоциальныйСтатус = МассивРазбитойСтроки[7];

СправочникОбразование.УровеньГрамотности = МассивРазбитойСтроки[8];

СправочникОбразование.УровеньОбразования = МассивРазбитойСтроки[9];

СправочникОбразование.ОбразовательныеУчреждения = МассивРазбитойСтроки[10];

СправочникРабота.СтатусЭкономическойАктичности = МассивРазбитойСтроки[11];

СправочникРабота.РодЗанятий = МассивРазбитойСтроки[12];

СправочникРабота.Работа = МассивРазбитойСтроки[13];

СправочникРабота.СтатусВозможнойРаботоспособности = МассивРазбитойСтроки[14];

СправочникМиграции.СтранаРождения = МассивРазбитойСтроки[15];

СправочникМиграции.ДатаМиграции = МассивРазбитойСтроки[16];

СправочникМиграции.Гражданство = МассивРазбитойСтроки[17];

СправочникСемья.ЧленСемьи = МассивРазбитойСтроки[18];

СправочникСемья.СтатусЧленаСемьи = МассивРазбитойСтроки[19];

СправочникСемья.Отношение = МассивРазбитойСтроки[20];

СправочникСемья.СтепеньРодства = МассивРазбитойСтроки[21];

СправочникНациональность.Национальность = МассивРазбитойСтроки[22];

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

СправочникНациональность.ПервыйЯзык = МассивРазбитойСтроки[23];

СправочникНациональность.Религия = МассивРазбитойСтроки[24];

СправочникНациональность.ДругиеЯзыки = МассивРазбитойСтроки[25];

СправочникАдрес.Область = МассивРазбитойСтроки[26];

СправочникАдрес.Регион = МассивРазбитойСтроки[27];

СправочникАдрес.НаселенныйПункт = МассивРазбитойСтроки[28];

СправочникАдрес.Улица = МассивРазбитойСтроки[29];

СправочникАдрес.Дом = МассивРазбитойСтроки[30];

СправочникАдрес.Строение = МассивРазбитойСтроки[31];

СправочникВопросы.Вопросы = МассивРазбитойСтроки[32];

СправочникВопросы.Ответы = МассивРазбитойСтроки[33];

СправочникАдрес.Записать();

СправочникВопросы.Записать();

СправочникМиграции.Записать();

СправочникНациональность.Записать();

СправочникОбразование.Записать();

СправочникПерсДанные.Записать();

СправочникРабота.Записать();

СправочникСемья.Записать();

СправочникСоцДанные.Записать();

СправочникГражданин.Код\_СправочникАдрес = Справочники.ДанныеОбАдресе.НайтиПоКоду(Счетчик).Ссылка;

СправочникГражданин.Код\_СправочникаПерсональныеДанные = Справочники.ПерсональныеДанные.НайтиПоКоду(Счетчик).Ссылка;

СправочникГражданин.Код\_СправочникаСоциальныеДанные = Справочники.СоциальныеДанные.НайтиПоКоду(Счетчик).Ссылка;

СправочникГражданин.Код\_СправочникаДанныеСемьи = Справочники.ДанныеОСемье.НайтиПоКоду(Счетчик).Ссылка;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

СправочникГражданин.Код\_СправочнниаДанныеОСтране = Справочники.ДанныеОСтране.НайтиПоКоду(Счетчик).Ссылка;

СправочникГражданин.Код\_СправочникаНациональности = Справочники.ДанныеОНациональность.НайтиПоКоду(Счетчик).Ссылка;

СправочникГражданин.Код\_СправочникаВопросы = Справочники.ВопросыДляГражданина.НайтиПоКоду(Счетчик).Ссылка;

СправочникГражданин.Записать();

СтатусЗагрузки = "Загрузка завершена! Количество введенных строк: " + КоличествоСтрок;

Счетчик = Счетчик + 1;

КонецЦикла;

Иначе

Сообщить("Файл не найден!");

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция РазмерСправочника()

КоличествоЗаписей = 0;

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| Справочник.Ссылка КАК Ссылка,

| МАКСИМУМ(Справочник.Код) КАК Код

|ИЗ

| Справочник.ДанныеОГражданине КАК Справочник

|

|СГРУППИРОВАТЬ ПО

| Справочник.Ссылка";

Выборка = Запрос.Выполнить().Выбрать();

Выборка.Следующий();

Если Выборка.Код = Неопределено Тогда

МаксимумКод = 0;

Возврат МаксимумКод;

Иначе

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

МаксимумКод = Выборка.Код;

Возврат МаксимумКод;

КонецЕсли;

КонецФункции

Код программы: окно функции «Поиск данных граждан»

&НаКлиенте

Процедура Меню(Команда)

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

Закрыть(ЭтаФорма);

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПроверкаМеню()

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин).Роль;

Если Пользователь = Ложь Тогда

Возврат 0;

Иначе

Возврат 1;

КонецЕсли;

КонецФункции

&НаКлиенте

Процедура Выход(Команда)

Перем ЗакрытиеОкна;

Попытка

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

ЗакрытиеОкон = ПолучитьОкна();

Для Каждого Элемент Из ЗакрытиеОкон Цикл

Если Не Элемент.Основное Тогда

Форма = Элемент.ПолучитьСодержимое();

Форма.Закрыть();

КонецЕсли;

КонецЦикла;

Исключение

КонецПопытки;

КонецПроцедуры

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

ФИОСотрудника = Параметры.ФИО;

Логин = Параметры.Логин;

КонецПроцедуры

&НаСервере

Процедура ПоискНаСервере()

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОГражданине.ФИО КАК ФИО,

| ПерсональныеДанные.Пол КАК Пол,

| ПерсональныеДанные.Возраст КАК Возраст,

| ПерсональныеДанные.БрачныйСтатус КАК БрачныйСтатус,

| Сотрудник.Логин КАК Логин,

| Сотрудник.Роль КАК Роль,

| СоциальныеДанные.СоциальныйСтатус КАК СоциальныйСтатус,

| СоциальныеДанные.Код\_СправочникаОбразование.УровеньГрамотности КАК УровеньГрамотности,

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| СоциальныеДанные.Код\_СправочникаОбразование.УровеньОбразования КАК УровеньОбразования,

| СоциальныеДанные.Код\_СправочникаОбразование.ОбразовательныеУчреждения КАК ОбразовательныеУчреждения,

| СоциальныеДанные.Код\_СправочникаРабота.СтатусЭкономическойАктичности КАК СтатусЭкономическойАктичности,

| СоциальныеДанные.Код\_СправочникаРабота.РодЗанятий КАК РодЗанятий,

| СоциальныеДанные.Код\_СправочникаРабота.Работа КАК Работа,

| СоциальныеДанные.Код\_СправочникаРабота.СтатусВозможнойРаботоспособности КАК СтатусВозможнойРаботоспособности,

| ДанныеОСтране.СтранаРождения КАК СтранаРождения,

| ДанныеОСтране.ДатаМиграции КАК ДатаМиграции,

| ДанныеОСтране.Гражданство КАК Гражданство,

| ДанныеОСемье.ЧленСемьи КАК ЧленСемьи,

| ДанныеОСемье.СтатусЧленаСемьи КАК СтатусЧленаСемьи,

| ДанныеОСемье.Отношение КАК Отношение,

| ДанныеОСемье.СтепеньРодства КАК СтепеньРодства,

| ДанныеОНациональность.Национальность КАК Национальность,

| ДанныеОНациональность.ПервыйЯзык КАК ПервыйЯзык,

| ДанныеОНациональность.Религия КАК Религия,

| ДанныеОНациональность.ДругиеЯзыки КАК ДругиеЯзыки,

| ДанныеОбАдресе.Область КАК Область,

| ДанныеОбАдресе.Регион КАК Регион,

| ДанныеОбАдресе.НаселенныйПункт КАК НаселенныйПункт,

| ДанныеОбАдресе.Улица КАК Улица,

| ДанныеОбАдресе.Дом КАК Дом,

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ВопросыДляГражданина.Вопросы КАК Вопросы,

| ВопросыДляГражданина.Ответы КАК Ответы,

| ДанныеОбАдресе.Строение КАК Строение

|ИЗ

| Справочник.ДанныеОГражданине КАК ДанныеОГражданине

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОбАдресе КАК ДанныеОбАдресе

| ПО ДанныеОГражданине.Код\_СправочникАдрес = ДанныеОбАдресе.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Сотрудник КАК Сотрудник

| ПО ДанныеОГражданине.Код\_СправочникаСотрудник = Сотрудник.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ВопросыДляГражданина КАК ВопросыДляГражданина

| ПО ДанныеОГражданине.Код\_СправочникаВопросы = ВопросыДляГражданина.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ПерсональныеДанные КАК ПерсональныеДанные

| ПО ДанныеОГражданине.Код\_СправочникаПерсональныеДанные = ПерсональныеДанные.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОНациональность КАК ДанныеОНациональность

| ПО ДанныеОГражданине.Код\_СправочникаНациональности = ДанныеОНациональность.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОСемье КАК ДанныеОСемье

| ПО ДанныеОГражданине.Код\_СправочникаДанныеСемьи = ДанныеОСемье.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОСтране КАК ДанныеОСтране

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ПО ДанныеОГражданине.Код\_СправочнниаДанныеОСтране = ДанныеОСтране.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.СоциальныеДанные КАК СоциальныеДанные

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОРаботе КАК ДанныеОРаботе

| ПО СоциальныеДанные.Код\_СправочникаРабота = ДанныеОРаботе.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеООбразовании КАК ДанныеООбразовании

| ПО СоциальныеДанные.Код\_СправочникаОбразование = ДанныеООбразовании.Ссылка

| ПО ДанныеОГражданине.Код\_СправочникаСоциальныеДанные = СоциальныеДанные.Ссылка

|ГДЕ

| ДанныеОГражданине.ФИО = &ПоискФИО";

Запрос.УстановитьПараметр("ПоискФИО",ПоискФИО);

ВыборкаТаблицы = Запрос.Выполнить().Выгрузить();

ТаблицаЗначений.Загрузить(ВыборкаТаблицы);

ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(ВыборкаТаблицы, "ТаблицаЗначений", "Таблица1");

ДокументПоиск = Документы.ПоискГраждан.СоздатьДокумент();

ДокументПоиск.Дата = ТекущаяДата();

ДокументПоиск.ИмяДокумента = "Поиск по " + ПоискФИО;

ДокументПоиск.Искал = Логин;

ДокументПоиск.ПоискПо = ПоискФИО;

ДокументПоиск.Записать();

КонецПроцедуры

&НаСервере

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Процедура ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(ТабВКоде, ТабРеквизит, ТабНаФорме)

Попытка

НовыеРеквизиты = Новый Массив;

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыеРеквизиты.Добавить(

Новый РеквизитФормы(

Колонка.Имя, Колонка.ТипЗначения,

ТабРеквизит));

КонецЦикла;

ИзменитьРеквизиты(НовыеРеквизиты);

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыйЭлемент = Элементы.Добавить(

ТабРеквизит + "\_" + Колонка.Имя, Тип("ПолеФормы"), Элементы[ТабНаФорме]);

НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;

НовыйЭлемент.ПутьКДанным = ТабРеквизит + "." + Колонка.Имя;

КонецЦикла;

ЗначениеВРеквизитФормы(ТабВКоде, ТабРеквизит);

Исключение

КонецПопытки;

КонецПроцедуры

&НаКлиенте

Процедура Поиск(Команда)

Если ПоискФИО = "" Тогда

Сообщить("Выберете сначала ФИО для поиска");

Иначе

ПоискНаСервере();

КонецЕсли;

КонецПроцедуры

Код программы: окно функции «Создание учетных записей сотрудника»

&НаКлиенте

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Процедура Меню(Команда)

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

Закрыть(ЭтаФорма);

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПроверкаМеню()

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин).Роль;

Если Пользователь = Ложь Тогда

Возврат 0;

Иначе

Возврат 1;

КонецЕсли;

КонецФункции

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

ЛогинАвторизованный = Параметры.Логин;

ЗаголовокНаСтранице = "Создание учетной записи сотрудника";

КонецПроцедуры

&НаСервере

Процедура СоздатьНаСервере()

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин);

ВыборкаЛогина = Пользователь.Логин;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

ВыборкаПароль = Пользователь.Пароль;

ВыборкаРоли = Пользователь.Роль;

Массив = Новый Массив;

Массив.Вставить(0,ВыборкаЛогина);

Массив.Вставить(1,ВыборкаПароль);

Массив.Вставить(2,ВыборкаРоли);

Если Логин = Массив[0] И Пароль = Массив[1] Тогда

Сообщить("Запись сотрудника уже существует");

Иначе

ЗаписьСотрудник = Справочники.Сотрудник.СоздатьЭлемент();

ЗаписьСотрудник.Наименование = Логин;

ЗаписьСотрудник.Логин = Логин;

ЗаписьСотрудник.Пароль = Пароль;

ЗаписьСотрудник.Роль = Роль;

ЗаписьСотрудник.Записать();

ДокументСозданияСотрудник = Документы.СозданиеСотрудника.СоздатьДокумент();

ДокументСозданияСотрудник.Дата = ТекущаяДата();

ДокументСозданияСотрудник.ИмяДокумента = "Создание запись " + Логин;

ДокументСозданияСотрудник.Создал = ЛогинАвторизованный;

ДокументСозданияСотрудник.ЛогинНовогоСотрудника = Логин;

ДокументСозданияСотрудник.Записать();

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура Создать(Команда)

Если Логин = "" И Пароль = "" Тогда

Сообщить("Заполните логин и пароль");

ИначеЕсли Логин <> "" И Пароль <> "" Тогда

СоздатьНаСервере();

Сообщить("Запись создана");

КонецЕсли;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

КонецПроцедуры

Код программы: окно функции «Выгрузка данных по субъектах»

&НаКлиенте

Процедура Выгрузить(Команда)

Если ВыборОбласти = "" Тогда

Сообщить("Не выбрана область");

Иначе

Выгрузка();

Сообщить("Выгрузка завершена, проверьте C:\Users\Public\Documents\export.txt");

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция Выгрузка()

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОГражданине.ФИО КАК ФИО,

| ПерсональныеДанные.Пол КАК Пол,

| ПерсональныеДанные.Возраст КАК Возраст,

| ПерсональныеДанные.БрачныйСтатус КАК БрачныйСтатус,

| Сотрудник.Логин КАК Логин,

| Сотрудник.Пароль КАК Пароль,

| Сотрудник.Роль КАК Роль,

| СоциальныеДанные.СоциальныйСтатус КАК СоциальныйСтатус,

| СоциальныеДанные.Код\_СправочникаОбразование.УровеньГрамотности КАК УровеньГрамотности,

| СоциальныеДанные.Код\_СправочникаОбразование.УровеньОбразования КАК УровеньОбразования,

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| СоциальныеДанные.Код\_СправочникаОбразование.ОбразовательныеУчреждения КАК ОбразовательныеУчреждения,

| СоциальныеДанные.Код\_СправочникаРабота.СтатусЭкономическойАктичности КАК СтатусЭкономическойАктичности,

| СоциальныеДанные.Код\_СправочникаРабота.РодЗанятий КАК РодЗанятий,

| СоциальныеДанные.Код\_СправочникаРабота.Работа КАК Работа,

| СоциальныеДанные.Код\_СправочникаРабота.СтатусВозможнойРаботоспособности КАК СтатусВозможнойРаботоспособности,

| ДанныеОСтране.СтранаРождения КАК СтранаРождения,

| ДанныеОСтране.ДатаМиграции КАК ДатаМиграции,

| ДанныеОСтране.Гражданство КАК Гражданство,

| ДанныеОСемье.ЧленСемьи КАК ЧленСемьи,

| ДанныеОСемье.СтатусЧленаСемьи КАК СтатусЧленаСемьи,

| ДанныеОСемье.Отношение КАК Отношение,

| ДанныеОСемье.СтепеньРодства КАК СтепеньРодства,

| ДанныеОНациональность.Национальность КАК Национальность,

| ДанныеОНациональность.ПервыйЯзык КАК ПервыйЯзык,

| ДанныеОНациональность.Религия КАК Религия,

| ДанныеОНациональность.ДругиеЯзыки КАК ДругиеЯзыки,

| ДанныеОбАдресе.Область КАК Область,

| ДанныеОбАдресе.Регион КАК Регион,

| ДанныеОбАдресе.НаселенныйПункт КАК НаселенныйПункт,

| ДанныеОбАдресе.Улица КАК Улица,

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ДанныеОбАдресе.Дом КАК Дом,

| ВопросыДляГражданина.Вопросы КАК Вопросы,

| ВопросыДляГражданина.Ответы КАК Ответы,

| ДанныеОбАдресе.Строение КАК Строение

|ИЗ

| Справочник.ДанныеОГражданине КАК ДанныеОГражданине

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОбАдресе КАК ДанныеОбАдресе

| ПО ДанныеОГражданине.Код\_СправочникАдрес = ДанныеОбАдресе.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Сотрудник КАК Сотрудник

| ПО ДанныеОГражданине.Код\_СправочникаСотрудник = Сотрудник.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ВопросыДляГражданина КАК ВопросыДляГражданина

| ПО ДанныеОГражданине.Код\_СправочникаВопросы = ВопросыДляГражданина.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ПерсональныеДанные КАК ПерсональныеДанные

| ПО ДанныеОГражданине.Код\_СправочникаПерсональныеДанные = ПерсональныеДанные.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОНациональность КАК ДанныеОНациональность

| ПО ДанныеОГражданине.Код\_СправочникаНациональности = ДанныеОНациональность.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОСемье КАК ДанныеОСемье

| ПО ДанныеОГражданине.Код\_СправочникаДанныеСемьи = ДанныеОСемье.Ссылка

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОСтране КАК ДанныеОСтране

| ПО ДанныеОГражданине.Код\_СправочнниаДанныеОСтране = ДанныеОСтране.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.СоциальныеДанные КАК СоциальныеДанные

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОРаботе КАК ДанныеОРаботе

| ПО СоциальныеДанные.Код\_СправочникаРабота = ДанныеОРаботе.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеООбразовании КАК ДанныеООбразовании

| ПО СоциальныеДанные.Код\_СправочникаОбразование = ДанныеООбразовании.Ссылка

| ПО ДанныеОГражданине.Код\_СправочникаСоциальныеДанные = СоциальныеДанные.Ссылка

|ГДЕ

| ДанныеОбАдресе.Область = &ВыборОбласти";

Запрос.УстановитьПараметр("ВыборОбласти",ВыборОбласти);

Результат = Запрос.Выполнить().Выбрать();

Количество = Запрос.Выполнить().Выбрать().Количество();

Текст = Новый ЗаписьТекста("C:\Users\Public\Documents\export.txt", КодировкаТекста.ANSI);

РезультатДок = Запрос.Выполнить().Выбрать();

Пока РезультатДок.Следующий() Цикл

Текст.ЗаписатьСтроку(Строка(РезультатДок.ФИО)+","+Строка(РезультатДок.Пол)+","+Строка(РезультатДок.Возраст)

+","+Строка(РезультатДок.БрачныйСтатус)+","+Строка(РезультатДок.Логин)+","+Строка(РезультатДок.Пароль)

+","+Строка(РезультатДок.Роль)+","+Строка(РезультатДок.СоциальныйСтатус)+","+Строка(РезультатДок.УровеньГрамотности)

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

+","+Строка(РезультатДок.УровеньОбразования)+","+Строка(РезультатДок.ОбразовательныеУчреждения)

+","+Строка(РезультатДок.СтатусЭкономическойАктичности)+","+Строка(РезультатДок.РодЗанятий)+","+Строка(РезультатДок.Работа)

+","+Строка(РезультатДок.СтатусВозможнойРаботоспособности)+","+Строка(РезультатДок.СтранаРождения)+","+Строка(РезультатДок.ДатаМиграции)

+","+Строка(РезультатДок.Гражданство)+","+Строка(РезультатДок.ЧленСемьи)+","+Строка(РезультатДок.СтатусЧленаСемьи)

+","+Строка(РезультатДок.Отношение)+","+Строка(РезультатДок.СтепеньРодства)+","+Строка(РезультатДок.Национальность)

+","+Строка(РезультатДок.ПервыйЯзык)+","+Строка(РезультатДок.Религия)+","+Строка(РезультатДок.ДругиеЯзыки)

+","+Строка(РезультатДок.Область)+","+Строка(РезультатДок.Регион)+","+Строка(РезультатДок.НаселенныйПункт)

+","+Строка(РезультатДок.Улица)+","+Строка(РезультатДок.Дом)+","+Строка(РезультатДок.Строение)

+","+Строка(РезультатДок.Вопросы)+","+Строка(РезультатДок.Ответы));

КонецЦикла;

Текст.Закрыть();

СтатусВыгрузки = "Выгрузка завершена. Количество записей: " + Количество;

ДокументВыгрузки = Документы.ВыгрузкаДанных.СоздатьДокумент();

ДокументВыгрузки.Дата = ТекущаяДата();

ДокументВыгрузки.ИмяДокумента = "Выгрузка по " + ВыборОбласти;

ДокументВыгрузки.Выгрузил = Логин;

ДокументВыгрузки.КоличествоЗаписей = Количество;

ДокументВыгрузки.Записать();

КонецФункции

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

ФИОСотрудника = Параметры.ФИО;

Логин = Параметры.Логин;

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ДанныеОбАдресе.Область КАК Область

|ИЗ

| Справочник.ДанныеОбАдресе КАК ДанныеОбАдресе";

Результат = Запрос.Выполнить();

Элементы.ВыборОбласти.СписокВыбора.ЗагрузитьЗначения(Результат.Выгрузить().ВыгрузитьКолонку("Область"));

ВыборОбласти = Элементы.ВыборОбласти.СписокВыбора[0].Значение;

КонецПроцедуры

&НаКлиенте

Процедура Меню(Команда)

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

Закрыть(ЭтаФорма);

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПроверкаМеню()

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин).Роль;

Если Пользователь = Ложь Тогда

Возврат 0;

Иначе

Возврат 1;

КонецЕсли;

КонецФункции

&НаКлиенте

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Процедура Выход(Команда)

Перем ЗакрытиеОкна;

Попытка

ЗакрытиеОкон = ПолучитьОкна();

Для Каждого Элемент Из ЗакрытиеОкон Цикл

Если Не Элемент.Основное Тогда

Форма = Элемент.ПолучитьСодержимое();

Форма.Закрыть();

КонецЕсли;

КонецЦикла;

Исключение

КонецПопытки;

КонецПроцедуры

Код программы: окно функции «Нахождение совпадения»

&НаКлиенте

Процедура Меню(Команда)

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

Закрыть(ЭтаФорма);

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПроверкаМеню()

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин).Роль;

Если Пользователь = Ложь Тогда

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Возврат 0;

Иначе

Возврат 1;

КонецЕсли;

КонецФункции

&НаКлиенте

Процедура Выход(Команда)

Перем ЗакрытиеОкна;

Попытка

ЗакрытиеОкон = ПолучитьОкна();

Для Каждого Элемент Из ЗакрытиеОкон Цикл

Если Не Элемент.Основное Тогда

Форма = Элемент.ПолучитьСодержимое();

Форма.Закрыть();

КонецЕсли;

КонецЦикла;

Исключение

КонецПопытки;

КонецПроцедуры

&НаКлиенте

Процедура НайтиСовпадение(Команда)

НайтиИЗаполнениеНаСервере();

КонецПроцедуры

&НаСервере

Процедура НайтиИЗаполнениеНаСервере()

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| " + СтрЗаменить(ВыборТаблицы,"Справочник.","") + ".Наименование КАК Наименование,

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| КОЛИЧЕСТВО(" + СтрЗаменить(ВыборТаблицы,"Справочник.","") + ".Ссылка) КАК КоличествоСовпадений

|ИЗ

| " + ВыборТаблицы + " КАК " + СтрЗаменить(ВыборТаблицы,"Справочник.","") + "

|

|СГРУППИРОВАТЬ ПО

| " + СтрЗаменить(ВыборТаблицы,"Справочник.","") + ".Наименование";

Результат = Запрос.Выполнить().Выгрузить();

ТаблицаЗначений.Загрузить(Результат);

ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(Результат, "ТаблицаЗначений", "Таблица1");

ДокументСовпадение = Документы.НайтиСовпадение.СоздатьДокумент();

ДокументСовпадение.Дата = ТекущаяДата();

ДокументСовпадение.ИмяДокумента = "Поиск совпадения от " + Логин;

ДокументСовпадение.Искал = Логин;

ДокументСовпадение.Записать();

КонецПроцедуры

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

ФИОГражданина = Параметры.ФИО;

Логин = Параметры.Логин;

Для Каждого Справочник Из Метаданные.Справочники Цикл

Элементы.ВыборТаблицы.СписокВыбора.Добавить(Справочник.ПолноеИмя());

КонецЦикла;

ВыборТаблицы = Элементы.ВыборТаблицы.СписокВыбора[0].Значение;

КонецПроцедуры

&НаСервере

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Процедура ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(ТабВКоде, ТабРеквизит, ТабНаФорме)

Попытка

НовыеРеквизиты = Новый Массив;

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыеРеквизиты.Добавить(

Новый РеквизитФормы(

Колонка.Имя, Колонка.ТипЗначения,

ТабРеквизит));

КонецЦикла;

ИзменитьРеквизиты(НовыеРеквизиты);

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыйЭлемент = Элементы.Добавить(

ТабРеквизит + "\_" + Колонка.Имя, Тип("ПолеФормы"), Элементы[ТабНаФорме]);

НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;

НовыйЭлемент.ПутьКДанным = ТабРеквизит + "." + Колонка.Имя;

КонецЦикла;

ЗначениеВРеквизитФормы(ТабВКоде, ТабРеквизит);

Исключение

КонецПопытки;

КонецПроцедуры

Код программы: окно функции «Подсчет граждан»

&НаКлиенте

Процедура Меню(Команда)

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

Закрыть(ЭтаФорма);

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура Подсчет(Команда)

Если Фильтр = "" И ИмяФильтра = "" Тогда

Сообщить("Выберете фильтр поиска и имя фильтра");

ИначеЕсли Фильтр <> "" И ИмяФильтра = "" Тогда

Сообщить("Выберете имя фильтра");

ИначеЕсли Фильтр <> "" И ИмяФильтра <> "" Тогда

ПодсчетГраждан();

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПодсчетГраждан()

Перем ИмяСправочника;

Если Фильтр = "Область" ИЛИ Фильтр = "Регион" ИЛИ Фильтр = "НаселенныйПункт" ИЛИ Фильтр = "Улица" ИЛИ Фильтр = "Дом"

ИЛИ Фильтр = "Строение" Тогда

ИмяСправочника = "ДанныеОбАдресе";

ИначеЕсли Фильтр = "Пол" ИЛИ Фильтр = "Возраст" ИЛИ Фильтр = "БрачныйСтатус" Тогда

ИмяСправочника = "ПерсональныеДанные";

ИначеЕсли Фильтр = "Национальность" ИЛИ Фильтр = "ПервыйЯзык" ИЛИ Фильтр = "ДругиеЯзыки" ИЛИ Фильтр = "Религия" Тогда

ИмяСправочника = "ДанныеОНациональность";

ИначеЕсли Фильтр = "СтранаРождения" ИЛИ Фильтр = "ДатаМиграции" ИЛИ Фильтр = "Гражданство" Тогда

ИмяСправочника = "ДанныеОСтране";

ИначеЕсли Фильтр = "СтатусЭкономическойАктичности" ИЛИ Фильтр = "РодЗанятий" ИЛИ Фильтр = "Работа"

ИЛИ Фильтр = "СтатусВозможнойРаботоспособности" Тогда

ИмяСправочника = "СоциальныеДанные.Код\_СправочникаРабота"

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

ИначеЕсли Фильтр = "УровеньГрамотности" ИЛИ Фильтр = "УровеньОбразования" ИЛИ Фильтр = "ОбразовательныеУчреждения" Тогда

ИмяСправочника = "СоциальныеДанные.Код\_СправочникаОбразование";

ИначеЕсли Фильтр = "СоциальныйСтатус" Тогда

ИмяСправочника = "СоциальныеДанные";

КонецЕсли;

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОГражданине.ФИО КАК ФИО,

| ПерсональныеДанные.Пол КАК Пол,

| ПерсональныеДанные.Возраст КАК Возраст,

| ПерсональныеДанные.БрачныйСтатус КАК БрачныйСтатус,

| Сотрудник.Логин КАК Логин,

| Сотрудник.Роль КАК Роль,

| СоциальныеДанные.СоциальныйСтатус КАК СоциальныйСтатус,

| СоциальныеДанные.Код\_СправочникаОбразование.УровеньГрамотности КАК УровеньГрамотности,

| СоциальныеДанные.Код\_СправочникаОбразование.УровеньОбразования КАК УровеньОбразования,

| СоциальныеДанные.Код\_СправочникаОбразование.ОбразовательныеУчреждения КАК ОбразовательныеУчреждения,

| СоциальныеДанные.Код\_СправочникаРабота.СтатусЭкономическойАктичности КАК СтатусЭкономическойАктичности,

| СоциальныеДанные.Код\_СправочникаРабота.РодЗанятий КАК РодЗанятий,

| СоциальныеДанные.Код\_СправочникаРабота.Работа КАК Работа,

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| СоциальныеДанные.Код\_СправочникаРабота.СтатусВозможнойРаботоспособности КАК СтатусВозможнойРаботоспособности,

| ДанныеОСтране.СтранаРождения КАК СтранаРождения,

| ДанныеОСтране.ДатаМиграции КАК ДатаМиграции,

| ДанныеОСтране.Гражданство КАК Гражданство,

| ДанныеОСемье.ЧленСемьи КАК ЧленСемьи,

| ДанныеОСемье.СтатусЧленаСемьи КАК СтатусЧленаСемьи,

| ДанныеОСемье.Отношение КАК Отношение,

| ДанныеОСемье.СтепеньРодства КАК СтепеньРодства,

| ДанныеОНациональность.Национальность КАК Национальность,

| ДанныеОНациональность.ПервыйЯзык КАК ПервыйЯзык,

| ДанныеОНациональность.Религия КАК Религия,

| ДанныеОНациональность.ДругиеЯзыки КАК ДругиеЯзыки,

| ДанныеОбАдресе.Область КАК Область,

| ДанныеОбАдресе.Регион КАК Регион,

| ДанныеОбАдресе.НаселенныйПункт КАК НаселенныйПункт,

| ДанныеОбАдресе.Улица КАК Улица,

| ДанныеОбАдресе.Дом КАК Дом,

| ВопросыДляГражданина.Вопросы КАК Вопросы,

| ВопросыДляГражданина.Ответы КАК Ответы,

| ДанныеОбАдресе.Строение КАК Строение

|ИЗ

| Справочник.ДанныеОГражданине КАК ДанныеОГражданине

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОбАдресе КАК ДанныеОбАдресе

| ПО ДанныеОГражданине.Код\_СправочникАдрес = ДанныеОбАдресе.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Сотрудник КАК Сотрудник

| ПО ДанныеОГражданине.Код\_СправочникаСотрудник = Сотрудник.Ссылка

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ВопросыДляГражданина КАК ВопросыДляГражданина

| ПО ДанныеОГражданине.Код\_СправочникаВопросы = ВопросыДляГражданина.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ПерсональныеДанные КАК ПерсональныеДанные

| ПО ДанныеОГражданине.Код\_СправочникаПерсональныеДанные = ПерсональныеДанные.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОНациональность КАК ДанныеОНациональность

| ПО ДанныеОГражданине.Код\_СправочникаНациональности = ДанныеОНациональность.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОСемье КАК ДанныеОСемье

| ПО ДанныеОГражданине.Код\_СправочникаДанныеСемьи = ДанныеОСемье.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОСтране КАК ДанныеОСтране

| ПО ДанныеОГражданине.Код\_СправочнниаДанныеОСтране = ДанныеОСтране.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.СоциальныеДанные КАК СоциальныеДанные

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеОРаботе КАК ДанныеОРаботе

| ПО СоциальныеДанные.Код\_СправочникаРабота = ДанныеОРаботе.Ссылка

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ДанныеООбразовании КАК ДанныеООбразовании

| ПО СоциальныеДанные.Код\_СправочникаОбразование = ДанныеООбразовании.Ссылка

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| ПО ДанныеОГражданине.Код\_СправочникаСоциальныеДанные = СоциальныеДанные.Ссылка

|ГДЕ

| "+ ИмяСправочника +"."+ Фильтр +" = &ЗначениеФильтра";

Запрос.УстановитьПараметр("ЗначениеФильтра",ИмяФильтра);

ВыборкаТаблицы = Запрос.Выполнить().Выгрузить();

Результат = "Было найдено: " + ВыборкаТаблицы.количество() + " граждан";

ДокументПодсчет = Документы.ПодсчетГраждан.СоздатьДокумент();

ДокументПодсчет.ИмяДокумента = "Подсчитано по " + Фильтр + "." + ИмяФильтра;

ДокументПодсчет.Дата = ТекущаяДата();

ДокументПодсчет.Подсчитал = Логин;

ДокументПодсчет.КоличествоПодсчитанных = ВыборкаТаблицы.Количество();

ДокументПодсчет.Записать();

КонецФункции

&НаСервере

Функция ПроверкаМеню()

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин).Роль;

Если Пользователь = Ложь Тогда

Возврат 0;

Иначе

Возврат 1;

КонецЕсли;

КонецФункции

&НаКлиенте

Процедура Выход(Команда)

Перем ЗакрытиеОкна;

Попытка

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

ЗакрытиеОкон = ПолучитьОкна();

Для Каждого Элемент Из ЗакрытиеОкон Цикл

Если Не Элемент.Основное Тогда

Форма = Элемент.ПолучитьСодержимое();

Форма.Закрыть();

КонецЕсли;

КонецЦикла;

Исключение

КонецПопытки;

КонецПроцедуры

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

ФИОСотрудника = Параметры.ФИО;

Логин = Параметры.Логин;

КонецПроцедуры

&НаКлиенте

Процедура ЗагрузитьФильтры(Команда)

Если Фильтр = Неопределено Тогда

Сообщить("Выберете фильтр");

Иначе

ЗагрузитьФильтрыНаСервере();

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ЗагрузитьФильтрыНаСервере()

Если Фильтр = "Область" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОбАдресе.Область КАК Область

|ИЗ

| Справочник.ДанныеОбАдресе КАК ДанныеОбАдресе";

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Область"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "Регион" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОбАдресе.Регион КАК Регион

|ИЗ

| Справочник.ДанныеОбАдресе КАК ДанныеОбАдресе";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Регион"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "НаселенныйПункт" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОбАдресе.НаселенныйПункт КАК НаселенныйПункт

|ИЗ

| Справочник.ДанныеОбАдресе КАК ДанныеОбАдресе";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("НаселенныйПункт"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "Улица" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОбАдресе.Улица КАК Улица

|ИЗ

| Справочник.ДанныеОбАдресе КАК ДанныеОбАдресе";

Выборка = Запрос.Выполнить();

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Улица"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "Дом" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОбАдресе.Дом КАК Дом

|ИЗ

| Справочник.ДанныеОбАдресе КАК ДанныеОбАдресе";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Дом"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "Строение" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОбАдресе.Строение КАК Строение

|ИЗ

| Справочник.ДанныеОбАдресе КАК ДанныеОбАдресе";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Строение"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "Пол" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ПерсональныеДанные.Пол КАК Пол

|ИЗ

| Справочник.ПерсональныеДанные КАК ПерсональныеДанные";

Выборка = Запрос.Выполнить();

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Пол"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "Возраст" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ПерсональныеДанные.Возраст КАК Возраст

|ИЗ

| Справочник.ПерсональныеДанные КАК ПерсональныеДанные";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Возраст"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "БрачныйСтатус" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ПерсональныеДанные.БрачныйСтатус КАК БрачныйСтатус

|ИЗ

| Справочник.ПерсональныеДанные КАК ПерсональныеДанные";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("БрачныйСтатус"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "Национальность" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОНациональность.Национальность КАК Национальность

|ИЗ

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

| Справочник.ДанныеОНациональность КАК ДанныеОНациональность";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Национальность"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "ПервыйЯзык" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОНациональность.ПервыйЯзык КАК ПервыйЯзык

|ИЗ

| Справочник.ДанныеОНациональность КАК ДанныеОНациональность";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("ПервыйЯзык"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "ДругиеЯзыки" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОНациональность.ДругиеЯзыки КАК ДругиеЯзыки

|ИЗ

| Справочник.ДанныеОНациональность КАК ДанныеОНациональность";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("ДругиеЯзыки"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "Религия" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОНациональность.Религия КАК Религия

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

|ИЗ

| Справочник.ДанныеОНациональность КАК ДанныеОНациональность";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Религия"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "СтранаРождения" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОСтране.СтранаРождения КАК СтранаРождения

|ИЗ

| Справочник.ДанныеОСтране КАК ДанныеОСтране";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("СтранаРождения"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "ДатаМиграции" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОСтране.ДатаМиграции КАК ДатаМиграции

|ИЗ

| Справочник.ДанныеОСтране КАК ДанныеОСтране";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("ДатаМиграции"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "Гражданство" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОСтране.Гражданство КАК Гражданство

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

|ИЗ

| Справочник.ДанныеОСтране КАК ДанныеОСтране";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Гражданство"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "СтатусЭкономическойАктивности" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОРаботе.СтатусЭкономическойАктичности КАК СтатусЭкономическойАктичности

|ИЗ

| Справочник.ДанныеОРаботе КАК ДанныеОРаботе";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("СтатусЭкономическойАктичности"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "РодЗанятий" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОРаботе.РодЗанятий КАК РодЗанятий

|ИЗ

| Справочник.ДанныеОРаботе КАК ДанныеОРаботе";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("РодЗанятий"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "Работа" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОРаботе.Работа КАК Работа

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

|ИЗ

| Справочник.ДанныеОРаботе КАК ДанныеОРаботе";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("Работа"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "СтатусВозможнойРаботоспособности" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеОРаботе.СтатусВозможнойРаботоспособности КАК СтатусВозможнойРаботоспособности

|ИЗ

| Справочник.ДанныеОРаботе КАК ДанныеОРаботе";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("СтатусВозможнойРаботоспособности"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "УровеньГрамотности" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеООбразовании.УровеньГрамотности КАК УровеньГрамотности

|ИЗ

| Справочник.ДанныеООбразовании КАК ДанныеООбразовании";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("УровеньГрамотности"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "УровеньОбразования" Тогда

Запрос = Новый Запрос;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Запрос.Текст = "ВЫБРАТЬ

| ДанныеООбразовании.УровеньОбразования КАК УровеньОбразования

|ИЗ

| Справочник.ДанныеООбразовании КАК ДанныеООбразовании";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("УровеньОбразования"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "ОбразовательныеУчреждения" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ДанныеООбразовании.ОбразовательныеУчреждения КАК ОбразовательныеУчреждения

|ИЗ

| Справочник.ДанныеООбразовании КАК ДанныеООбразовании";

Выборка = Запрос.Выполнить();

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("ОбразовательныеУчреждения"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

ИначеЕсли Фильтр = "СоциальныйСтатус" Тогда

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| СоциальныеДанные.СоциальныйСтатус КАК СоциальныйСтатус

|ИЗ

| Справочник.СоциальныеДанные КАК СоциальныеДанные";

Выборка = Запрос.Выполнить();

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Элементы.ИмяФильтра.СписокВыбора.ЗагрузитьЗначения(Выборка.Выгрузить().ВыгрузитьКолонку("СоциальныйСтатус"));

ИмяФильтра = Элементы.ИмяФильтра.СписокВыбора[0].Значение;

КонецЕсли;

КонецФункции

Код программы: окно функции «Вывод остальных данных»

&НаКлиенте

Процедура Меню(Команда)

Значение = ПроверкаМеню();

ПараметрыЛогин = Новый Структура("Логин",Логин);

Если Значение = 0 Тогда

ОткрытьФорму("ОбщаяФорма.МенюСотрудника",ПараметрыЛогин);

Закрыть(ЭтаФорма);

Иначе

ОткрытьФорму("ОбщаяФорма.МенюАдмина",ПараметрыЛогин);

Закрыть(ЭтаФорма);

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПроверкаМеню()

Пользователь = Справочники.Сотрудник.НайтиПоНаименованию(Логин).Роль;

Если Пользователь = Ложь Тогда

Возврат 0;

Иначе

Возврат 1;

КонецЕсли;

КонецФункции

&НаСервере

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

ФИОСотрудника = Параметры.ФИО;

Логин = Параметры.Логин;

Для Каждого Справочник Из Метаданные.Справочники Цикл

Элементы.ПолеВыборТаблиц.СписокВыбора.Добавить(Справочник.ПолноеИмя());

КонецЦикла;

Фильтр = Элементы.ПолеВыборТаблиц.СписокВыбора[0].Значение;

КонецПроцедуры

&НаКлиенте

Процедура Выход(Команда)

Перем ЗакрытиеОкна;

Попытка

ЗакрытиеОкон = ПолучитьОкна();

Для Каждого Элемент Из ЗакрытиеОкон Цикл

Если Не Элемент.Основное Тогда

Форма = Элемент.ПолучитьСодержимое();

Форма.Закрыть();

КонецЕсли;

КонецЦикла;

Исключение

КонецПопытки;

КонецПроцедуры

&НаСервере

Процедура ПоискНаСервере()

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| \*

|ИЗ

| " + Фильтр + " КАК " + СтрЗаменить(Фильтр,"Справочник.","");

Результат = Запрос.Выполнить().Выгрузить();

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

ТаблицаЗначений.Загрузить(Результат);

ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(Результат, "ТаблицаЗначений", "Таблица1");

ДокументПоискаДругихДанных = Документы.ВыводДругихДанных.СоздатьДокумент();

ДокументПоискаДругихДанных.Дата = ТекущаяДата();

ДокументПоискаДругихДанных.ИмяДокумента = "Искал данные по " + Фильтр;

ДокументПоискаДругихДанных.Искал = Логин;

ДокументПоискаДругихДанных.ФильтрПоиска = Фильтр;

ДокументПоискаДругихДанных.Записать();

КонецПроцедуры

&НаКлиенте

Процедура Поиск(Команда)

Если Фильтр = "" Тогда

Сообщить("Выберете фильтр");

Иначе

ПоискНаСервере();

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура ВывестиТаблицуЗначенияВКодеВТаблицуНаФорме(ТабВКоде, ТабРеквизит, ТабНаФорме)

Попытка

НовыеРеквизиты = Новый Массив;

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыеРеквизиты.Добавить(

Новый РеквизитФормы(

Колонка.Имя, Колонка.ТипЗначения,

ТабРеквизит));

КонецЦикла;

ИзменитьРеквизиты(НовыеРеквизиты);

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

Для Каждого Колонка Из ТабВКоде.Колонки Цикл

НовыйЭлемент = Элементы.Добавить(

ТабРеквизит + "\_" + Колонка.Имя, Тип("ПолеФормы"), Элементы[ТабНаФорме]);

НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;

НовыйЭлемент.ПутьКДанным = ТабРеквизит + "." + Колонка.Имя;

КонецЦикла;

ЗначениеВРеквизитФормы(ТабВКоде, ТабРеквизит);

Исключение

КонецПопытки;

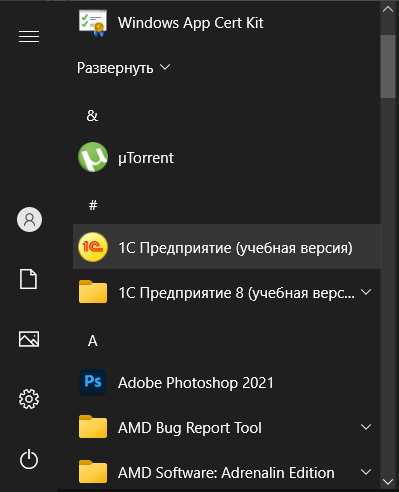
КонецПроцедуры

ПРИЛОЖЕНИЯ Б

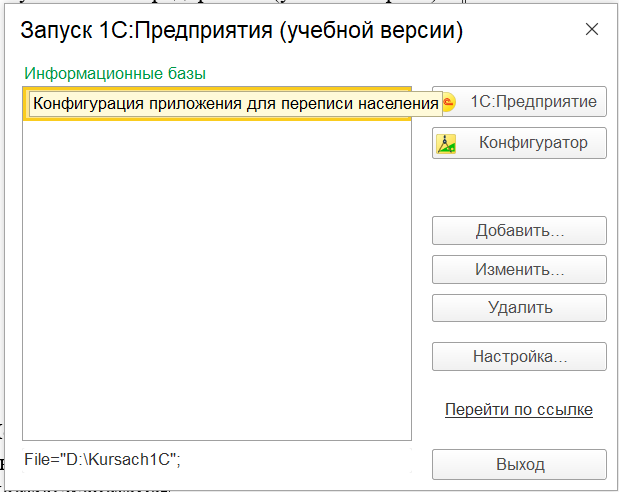
Инструкция пользователя

Запуск приложения

Чтобы запустить приложение с данной системой, необходимо установленная учебная версия платформы 1С «Предприятие 8.3». Если платформа уже есть на устройстве необходимо зайти в меню «Пуск» и запустить «1С Предприятие (учебная версия)» (Рисунок 29).

  
 Рисунок 29 – 1С Предприятие в меню «Пуск»

После запуска платформы запуститься окно с выбором информационной базы, в которой нужно выбрать конфигурацию с переписью населения и нажать на кнопку «1С: Предприятие» (Рисунок 30).

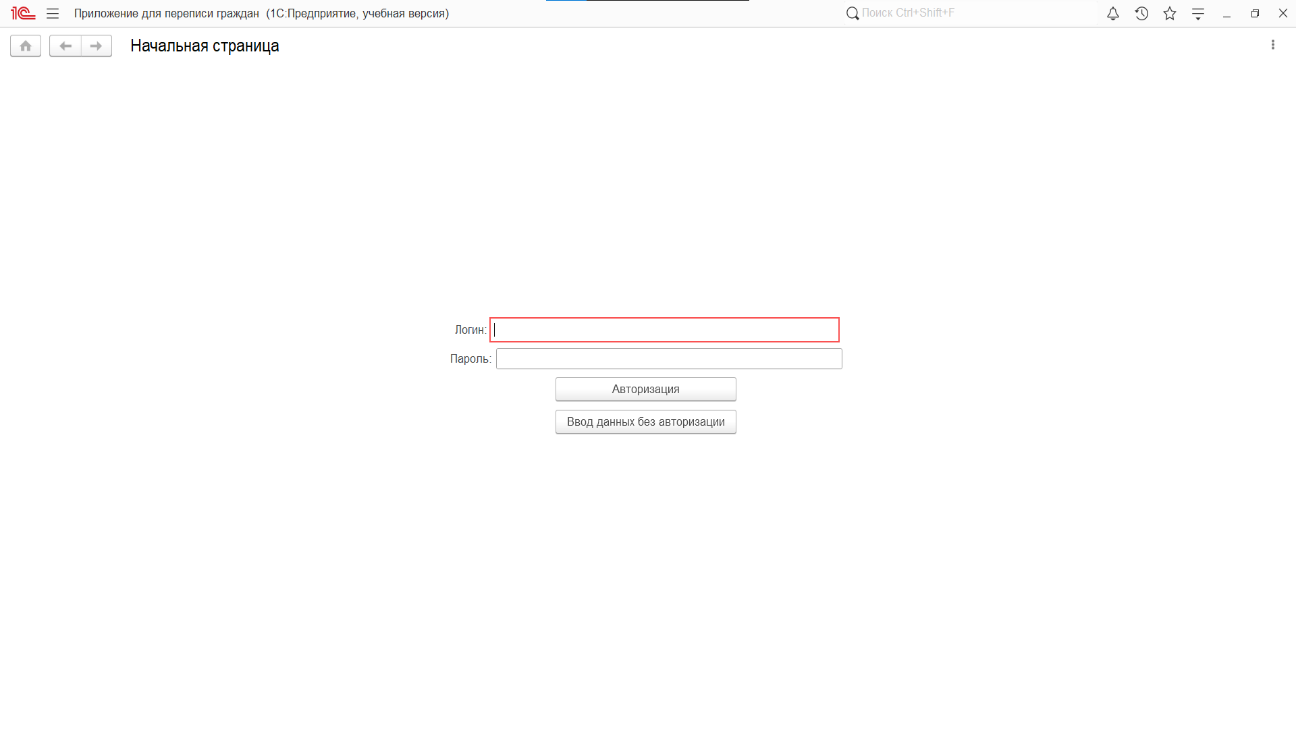
  
 Рисунок 30 – Окно с выбором конфигурации

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

Доступ к системе

После запуска конфигурации перед пользователем открывается начальное окно с функциями (Рисунок 31):

1. Авторизация
2. Ввод данных без авторизации

  
 Рисунок 31 – Начальная страница

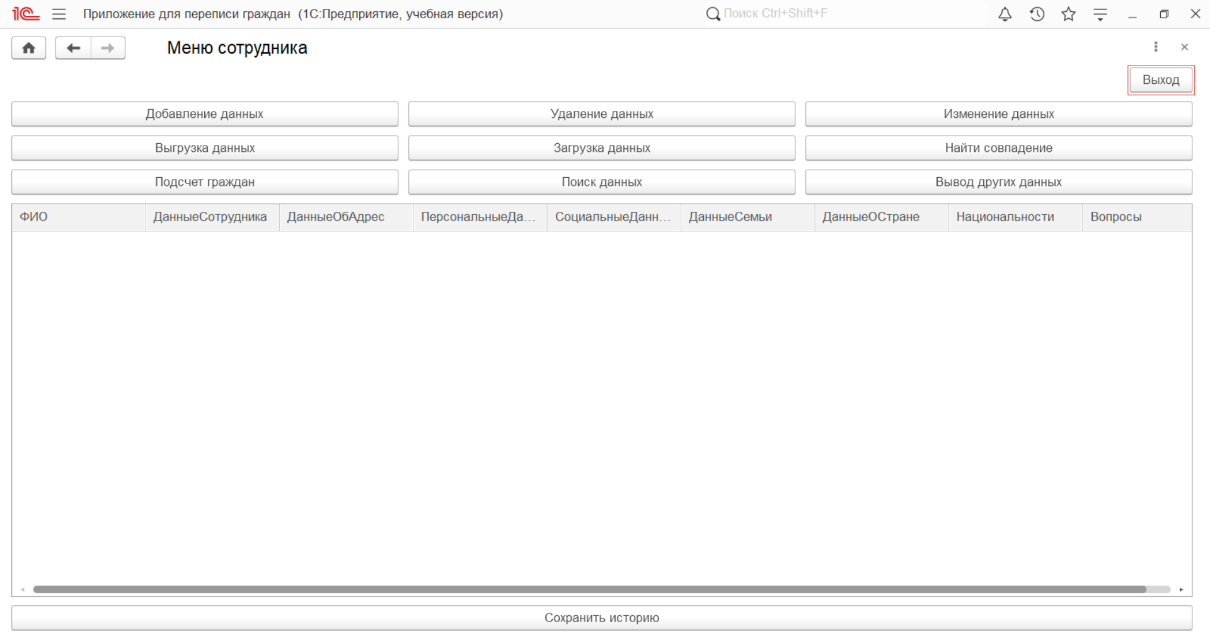
Пользователь-Сотрудник

Пользователь с ролью Сотрудник может авторизоваться в систему, если знает логин и пароль, введя логин и пароль в данные поля. Если запись о сотруднике существует, то откроется окно сотрудника с доступными для него функциями (Рисунок 32):

1. добавление данных гражданина;
2. удаление данных гражданина;
3. изменение данных гражданина;
4. выгрузка данных по области;
5. загрузка данных из текстового файла;

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

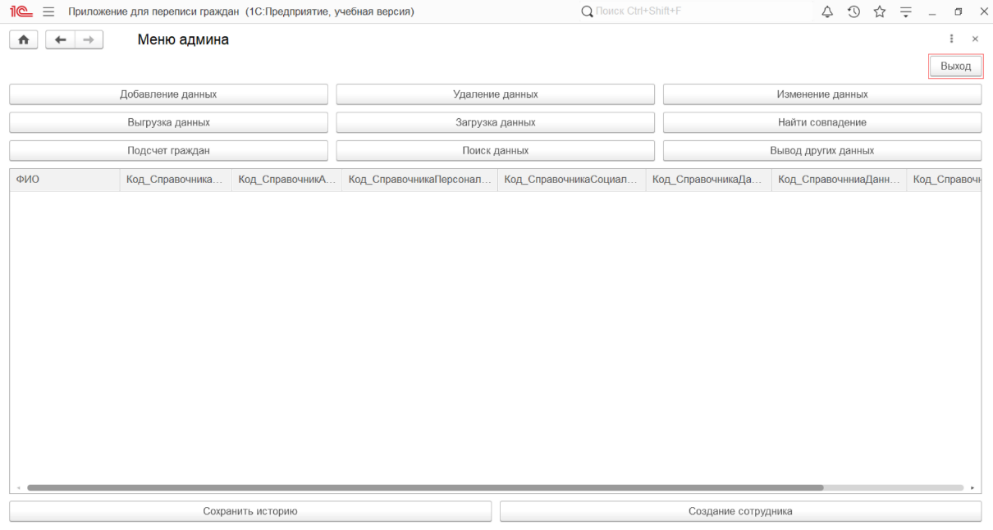
1. нахождение совпадения в таблицах;
2. подсчет граждан по определенному критерию;
3. поиск данных граждан;
4. вывод остальных данных по таблице;
5. сохранение истории действий.

  
 Рисунок 32 – Меню сотрудника

Пользователь-Админ

Пользователь с ролью Админ может также авторизироваться, как пользователь-сотрудник, после авторизации для него откроется схожее окно, как у пользователя-сотрудника, но с дополнительной функцией (Рисунок 33):

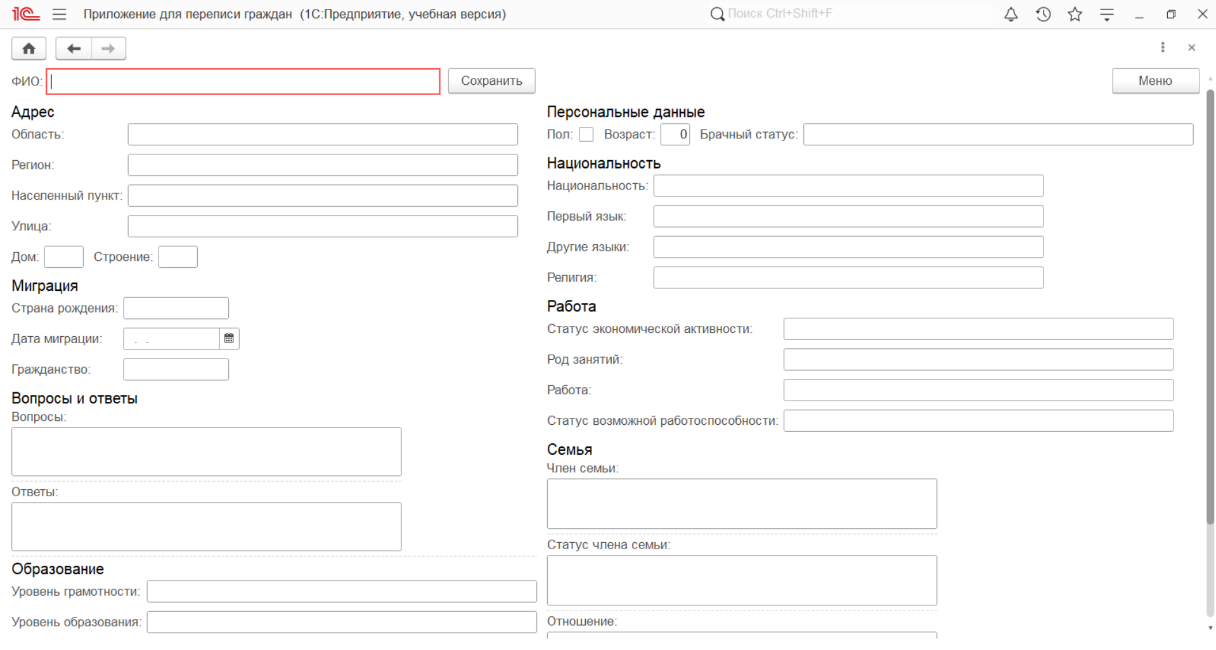
Создание учетной записи сотрудника

  
 Рисунок 33 – Меню админа

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

Пользователь-Гражданин

Пользователь-Гражданин являются пользователи без роли и не могут авторизоваться, но могут воспользоваться функцией «Ввод данных без авторизации» (Рисунок 34)

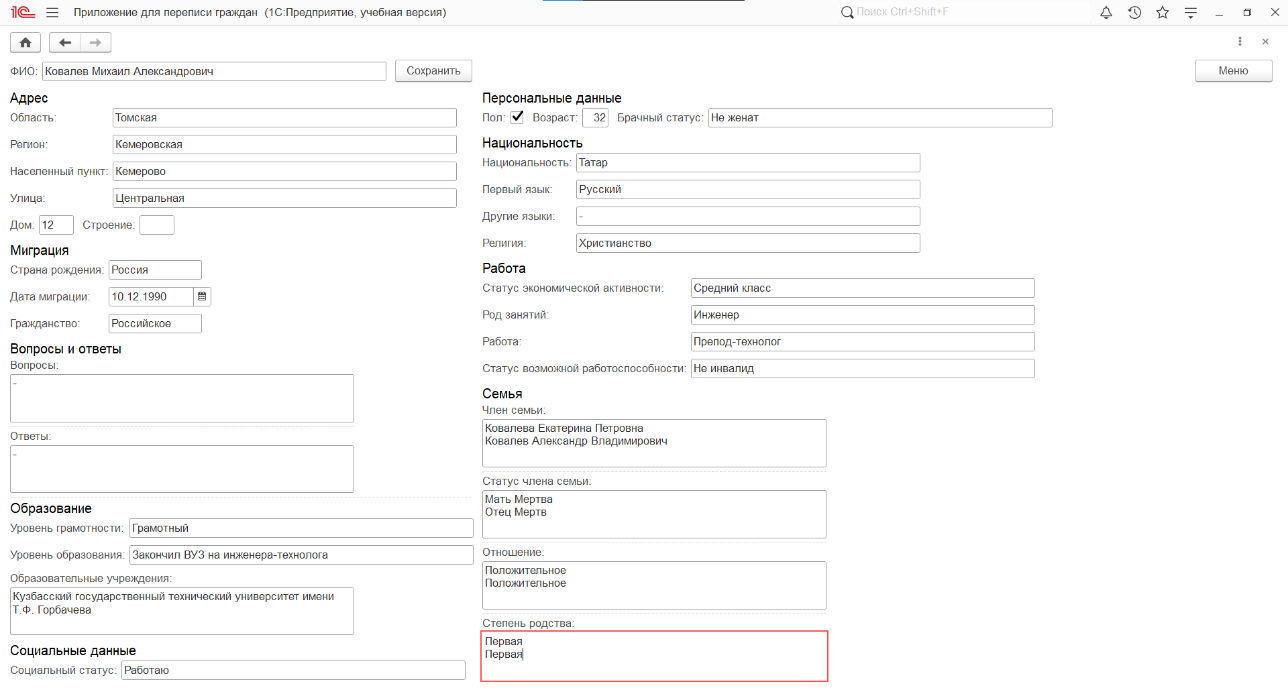
  
 Рисунок 34 – Окно ввода данных гражданина

Работа функций

Ввод данных без авторизации

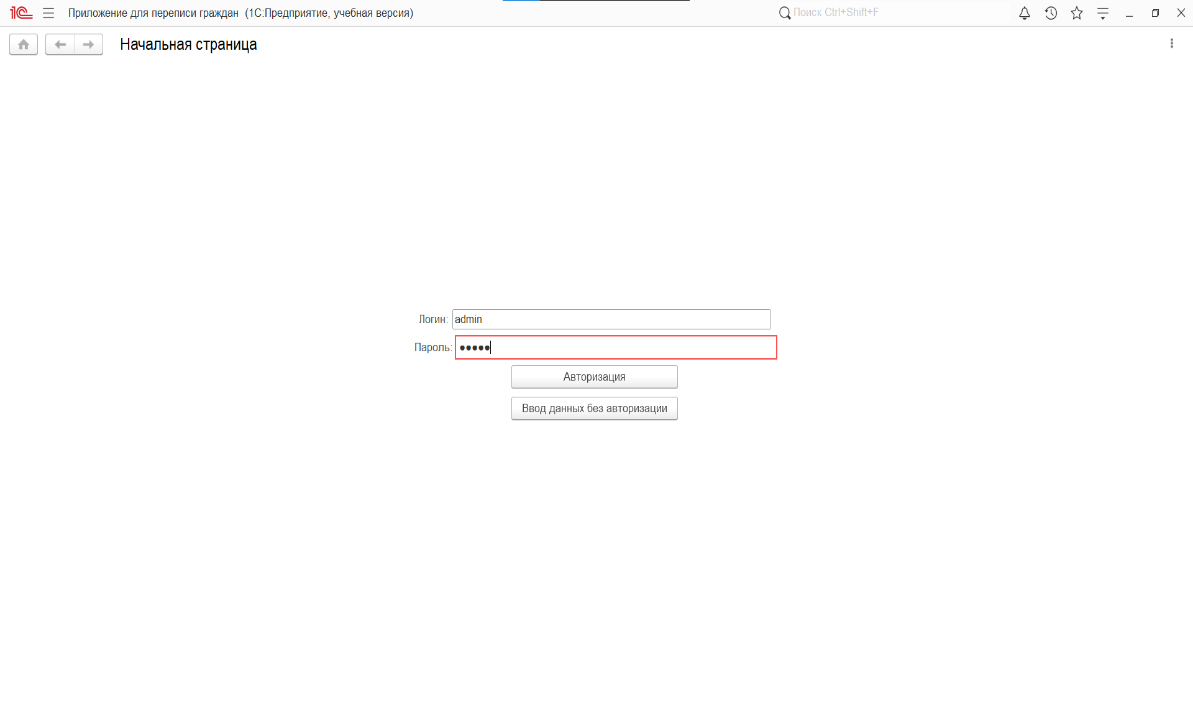
После нажатия на кнопку «Ввод данных без авторизации» на окне авторизации откроется окно с полями. Пользователь должен обязательно заполнить поле ФИО, но остальные поля не необходимы для создания записи о гражданине, поэтому заполнив поле ФИО пользователь может нажать кнопку «Сохранить» (Рисунок 35).

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

  
 Рисунок 35 – Заполненное окно для ввода данных

Авторизация

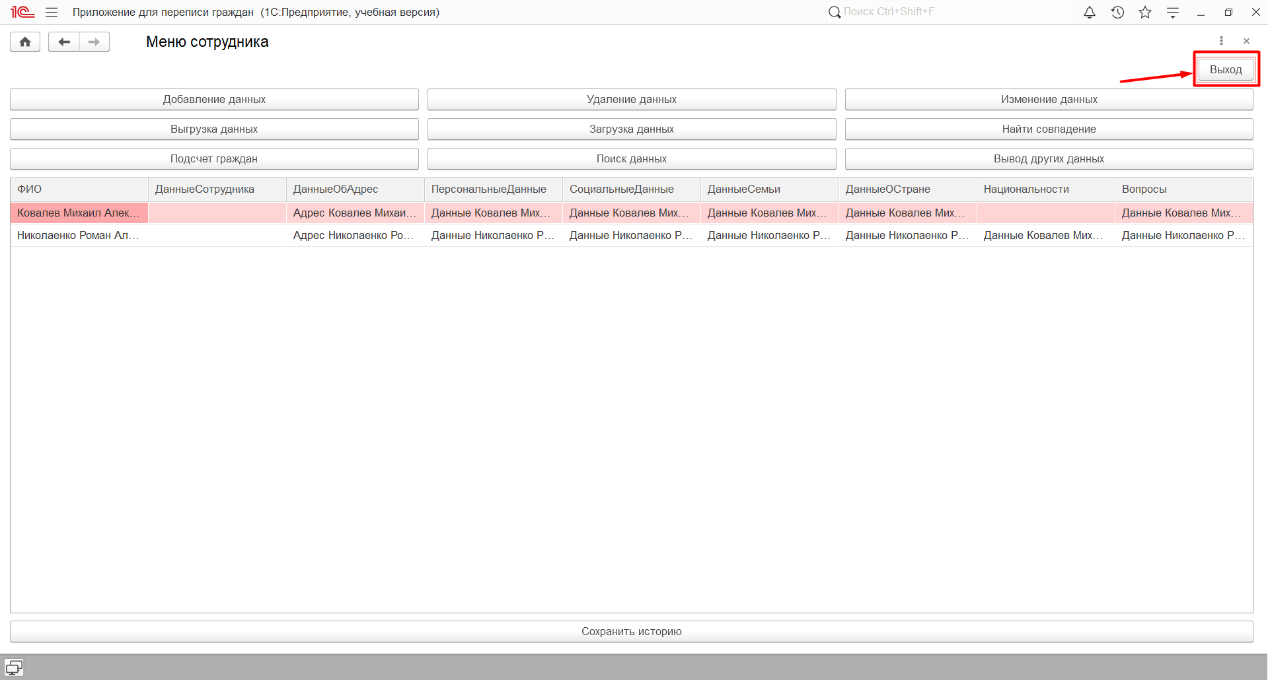
Пользователь после заполнения полей логина и пароля должен нажать кнопку «Авторизация» (Рисунок 36), после нажатия кнопка производиться поиск в справочнике по логину. Если запись находится в справочнике, то взависимости от значения роли в найденном записи откроется одно из двух окон: окно администратора или окно сотрудника.

  
 Рисунок 36 – Окно авторизации с заполненными полями

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

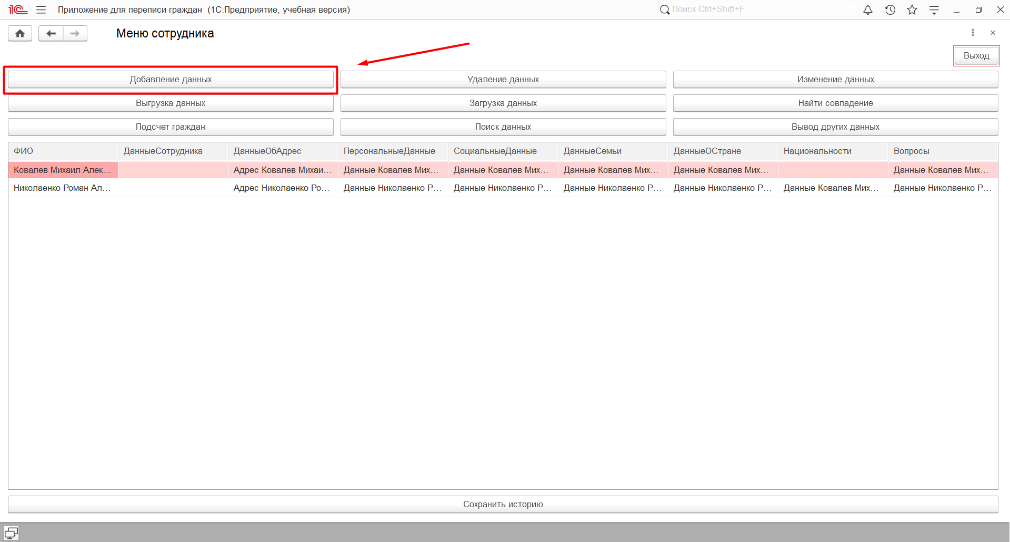
Выход из учетной записи пользователя

Чтобы воспользоваться данной функцией пользователю достаточно нажать кнопку «Выход». Данная кнопка присутствует на большинство окнах. После нажатия на эту кнопку закрываются все окна, кроме начальной, то есть окна авторизации (Рисунок 37).

  
 Рисунок 37 - Окно авторизации с отметкой кнопки

Добавление данных гражданина

После авторизации в системе пользователь может нажать на кнопку «Добавление данных» (Рисунок 38), которая есть в меню сотрудника и меню админа, откроется окно добавления данных без авторизации (Рисунок 33), но эта функция отличается тем, что после записи откроется окно не меню авторизации, а меню сотрудника или админа.

  
 Рисунок 38 – Кнопка «Добавления данных» в меню сотрудника

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

Удаление данных гражданина

После авторизации в системе пользователь может нажать на кнопку «Удаление данных» (Рисунок 39), которая есть в меню сотрудника и меню админа, откроется окно с функцией удаления данных (Рисунок 40). Чтобы удалить запись о гражданине, необходимо ввести в поле номер записи, который отображается в таблице с данными о гражданине. После ввода номера необходимо нажать кнопку «Удалить» и запись о гражданине удаляется из всех справочников, которые были связаны с записью, а так же высветится сообщение о удаленной записи.

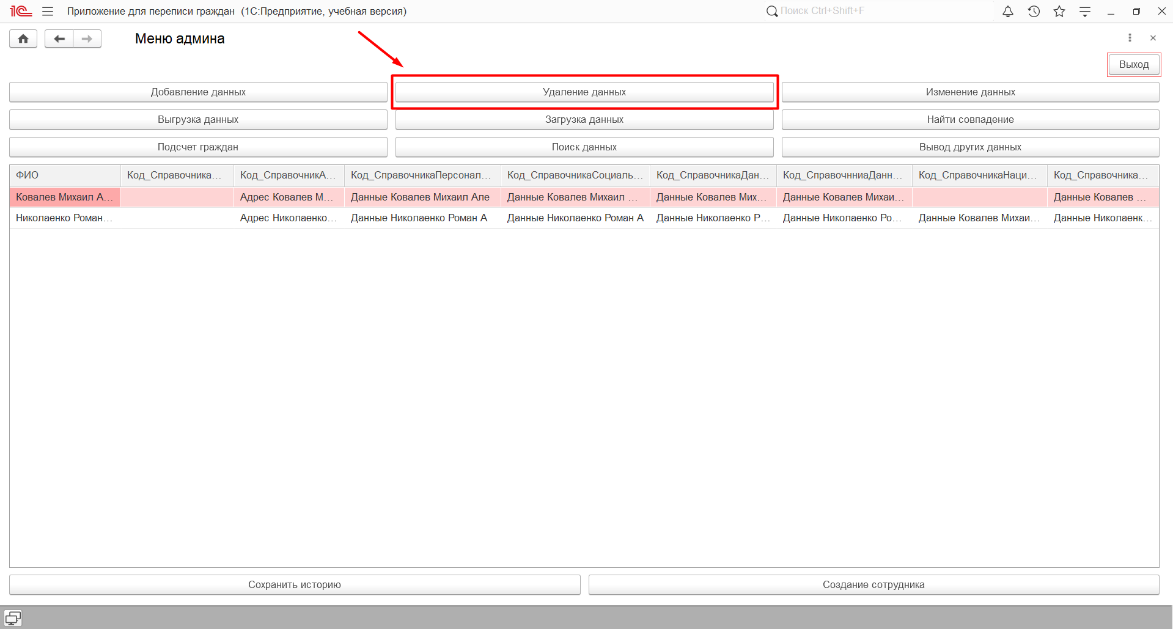
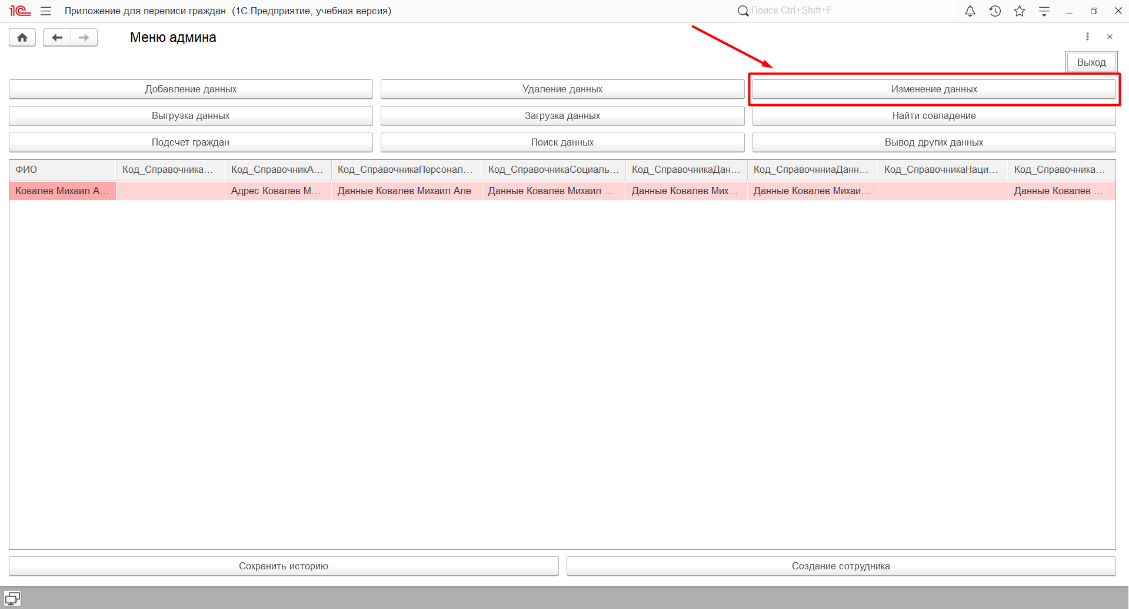
  
 Рисунок 39 – Кнопка «Удаления данных» в меню админа

  
 Рисунок 40 – Окно функции удаления данных с успешным удалением записи

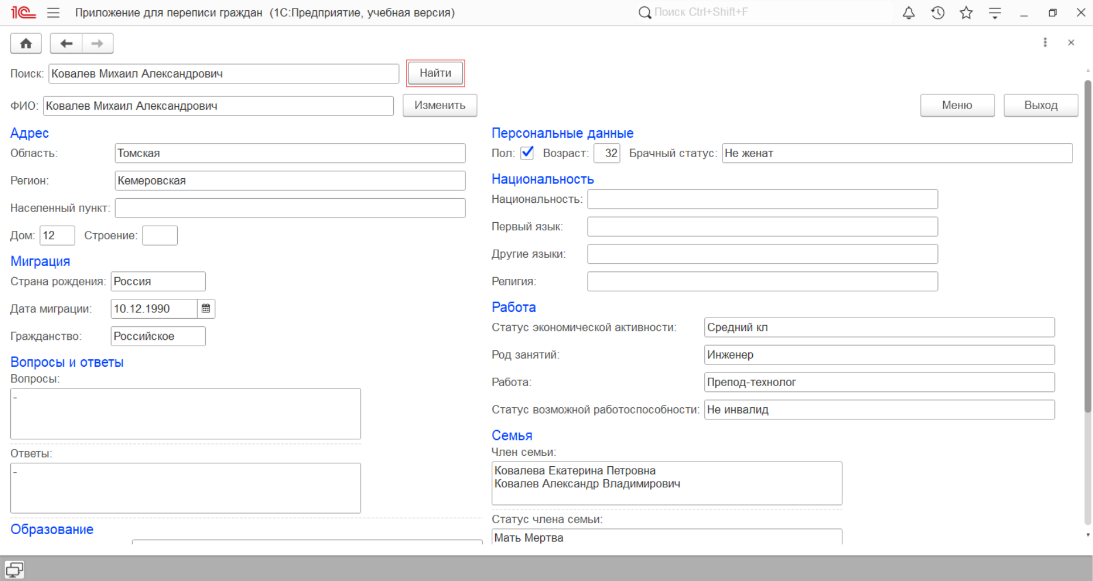
ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

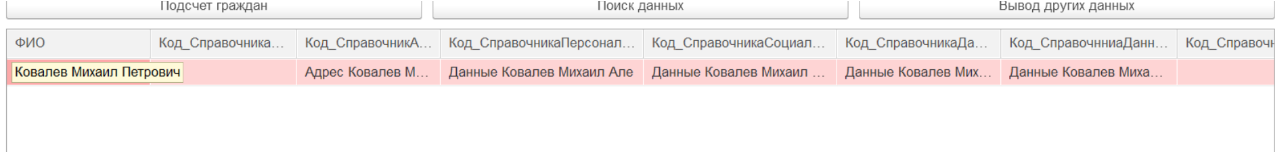
Изменение данных гражданина

После авторизации в системе пользователь может нажать на кнопку «Изменение данных» (Рисунок 41), которая есть в меню сотрудника и меню админа, откроется окно с функцией изменения данных (Рисунок 42). Чтобы изменить запись о гражданине пользователю необходимо заполнить поле поиск, в которое должно записываться ФИО гражданина. Если запись находиться, то все остальные поля заполняются данными из справочника найденного гражданина, если иначе, то выводится статус о том, что запись не была найдена. Изменив данные в полях, то данные перезаписываются во всех справочниках, где была упомянута данная запись. И в меню отобразится уже измененные данные этого гражданина (Рисунок 43).

  
 Рисунок 41 – Кнопка «Изменение данных» в меню админа

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

  
 Рисунок 42 – Окно изменения данных

  
 Рисунок 43 – Измененная запись в меню

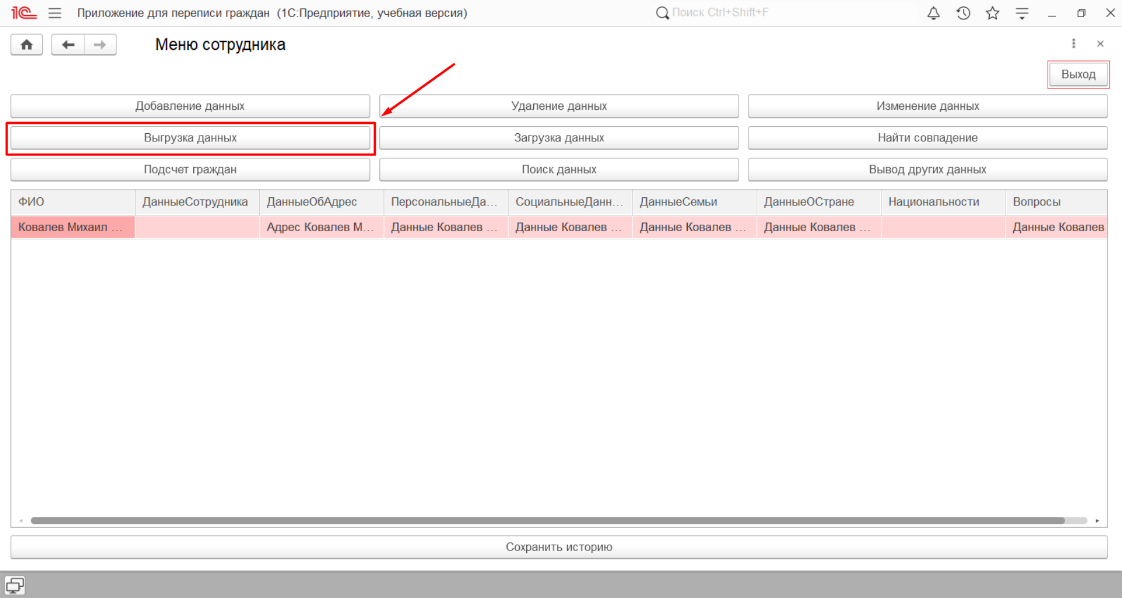
Сохранение данных гражданина

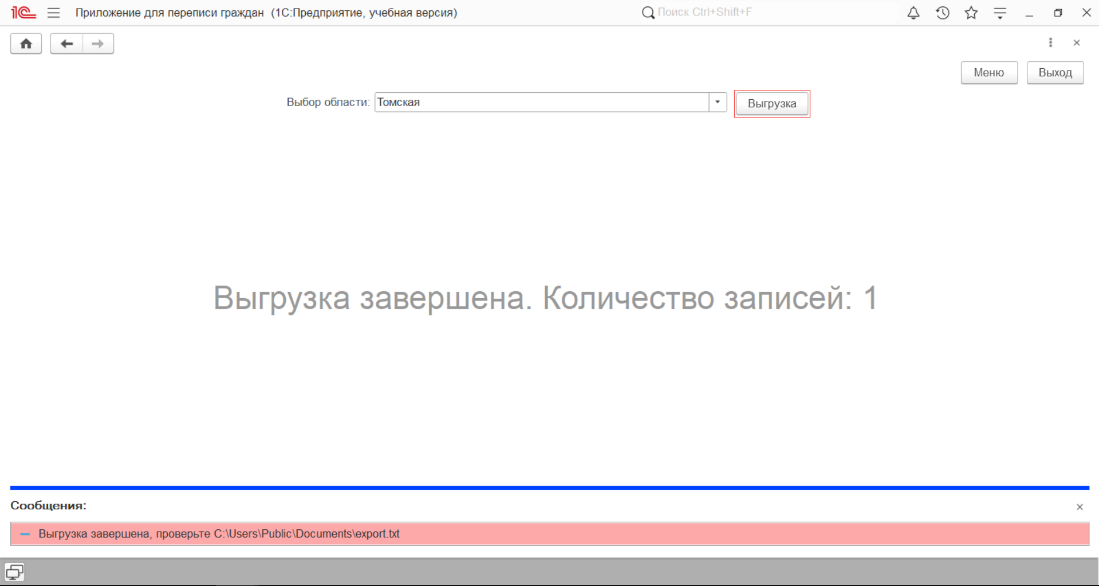
Данная функция встроена в другие функции, где производится запись данных в справочник или документ.

Выгрузка данных по области

После авторизации в системе пользователь может нажать на кнопку «Выгрузка данных» (Рисунок 44), которая есть в меню сотрудника и меню админа, откроется окно с функцией выгрузки данных в текстовый файл. Чтобы выгрузить данные необходимо в окне этой функции выбрать область, по которой будет делаться запрос по выборке данных. После этой выборки будет производиться создание текстового файла в общую папку документов, путь которой будет выведен в сообщении после выгрузки. (Рисунок 45)

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

  
 Рисунок 44 – Кнопка «Выгрузка данных» в меню сотрудника

  
 Рисунок 45 – Окно функции «Выгрузки данных» с результатом выгрузки

Загрузка данных из файла

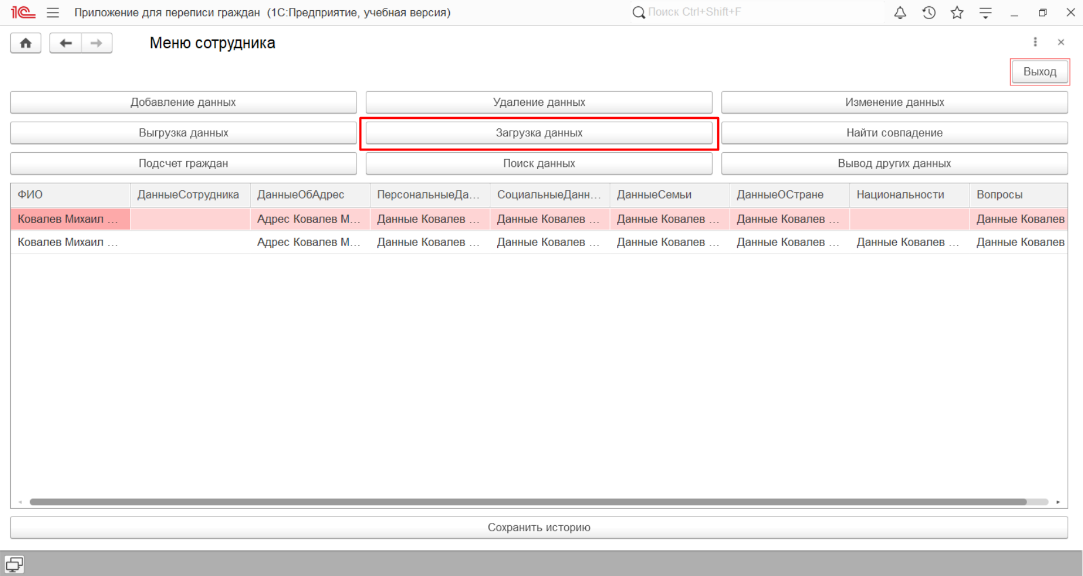
После авторизации в системе пользователь может нажать на кнопку «Загрузка данных» (Рисунок 46), которая есть в меню сотрудника и меню админа, откроется окно с функцией загрузки данных из текстового файла (Рисунок 47). Чтобы загрузить данные из файла запись в файле должно соблюдаться правило для прочтения строки выбранного файла:

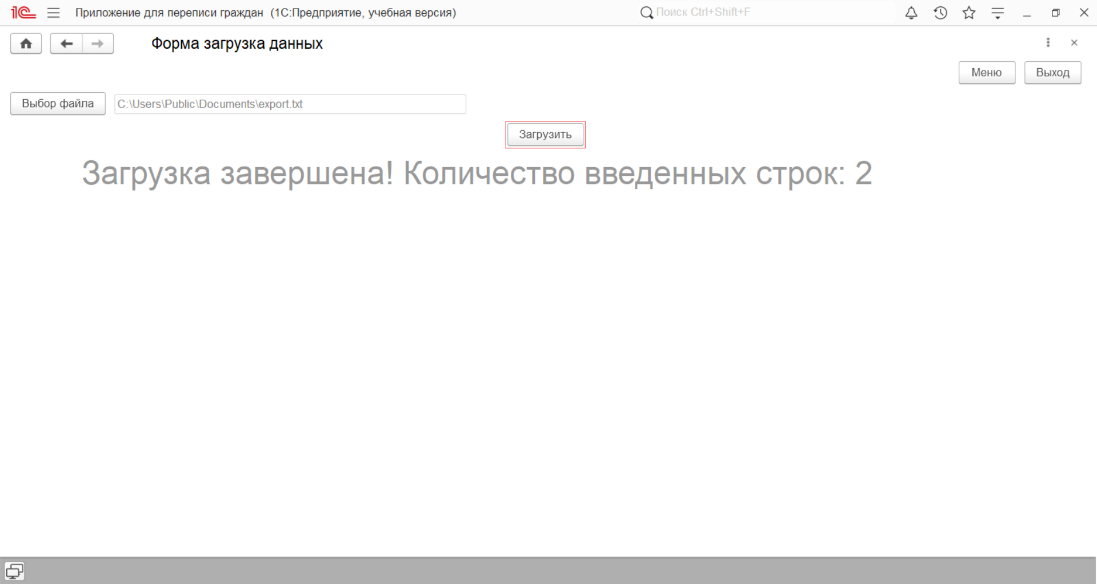
<ФИО>,<Пол>,<Возраст>,<БрачныйСтатус>,<Логин>,<Пароль>,<Роль>,<СоциальныйСтатус>,<УровеньГрамотности>,<УровеньОбразования>,<ОбразовательныеУчреждения>,<СтатусЭкономическойАктичности>,<РодЗанятий>,<Работа>,<СтатусВозможнойРаботоспособности>,<СтранаРождения>,<ДатаМ

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

играции>,<Гражданство>,<ЧленСемьи>.<СтатусЧленаСемьи>,<Отношение>,<СтепеньРодства>,<Национальность>,<ПервыйЯзык>,<Религия>,<ДругиеЯзыки>,<Область>,<Регион>,<НаселенныйПункт>,<Улица>,<Дом>,<Строение>,<Вопросы>,<Ответы>

Каждая строка прочитывается системой, создаются записи в справочниках для каждой строки и потом записи сохраняются.

  
 Рисунок 46 – Кнопка «Загрузка данных» в меню сотрудника

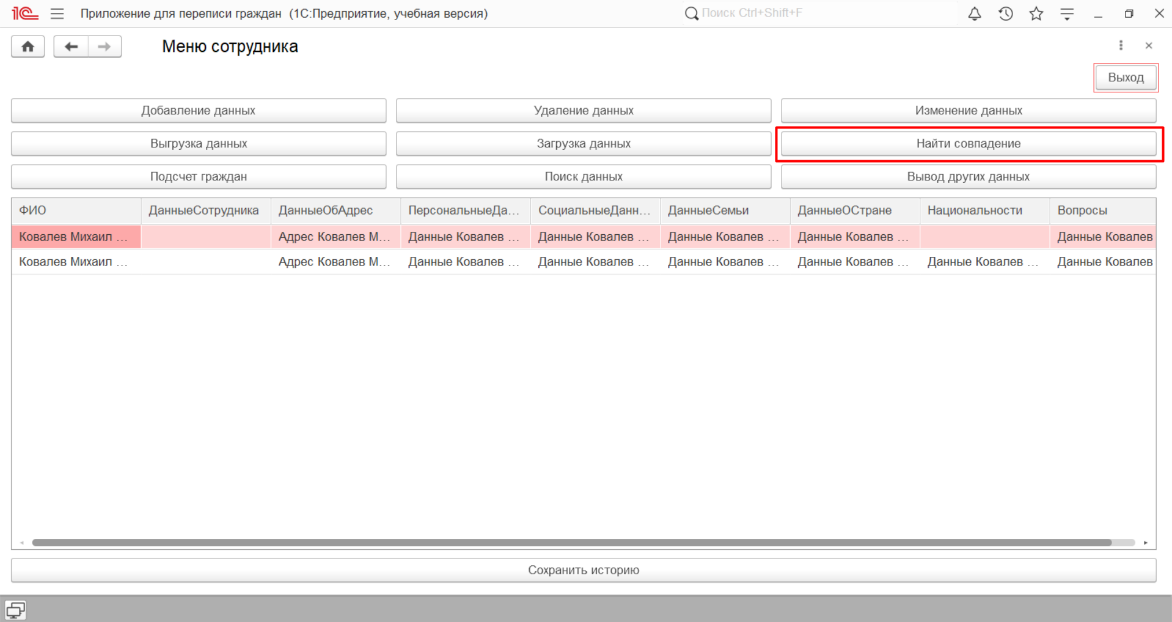
  
 Рисунок 47 – Окно функции «Загрузка данных»

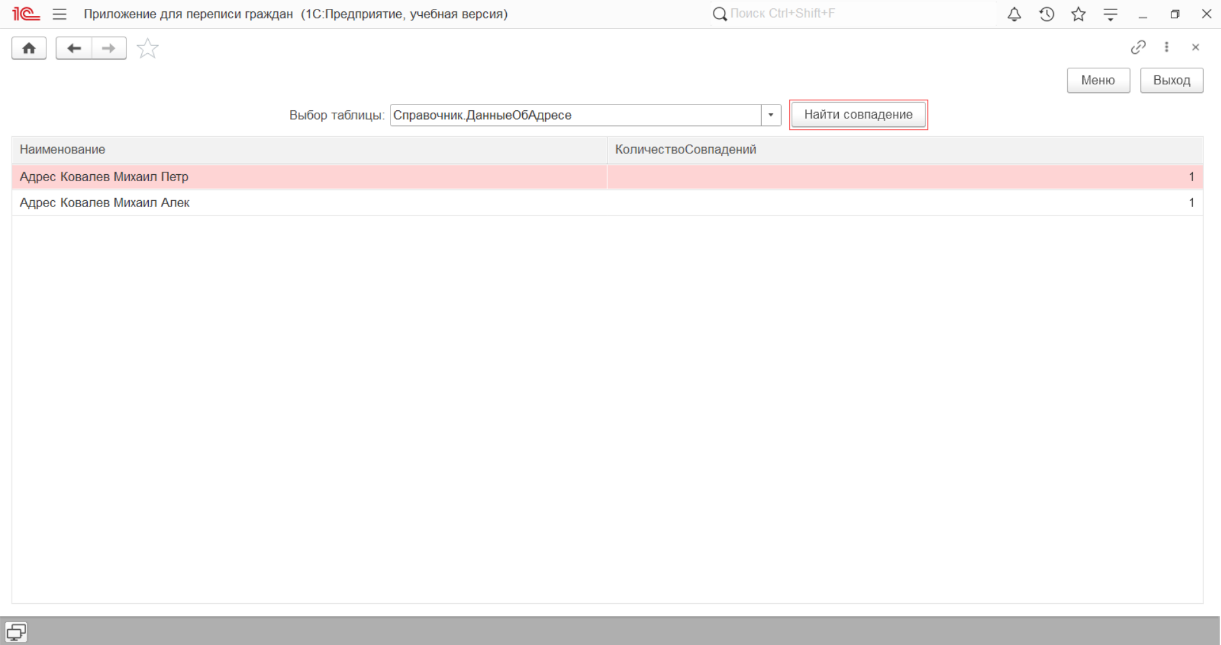
Найти совпадение в таблицах

После авторизации в системе пользователь может нажать на кнопку «Найти совпадение» (Рисунок 48), которая есть в меню сотрудника и меню админа, откроется окно с функцией нахождения идентичных данных по имени в выбранном справочнике.

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

Чтобы проверить, пользователю необходимо выбрать имя справочника, в котором будет производиться поиск идентичных данных (Рисунок 49). После нажатия кнопки «Найти совпадение» на окне функции выполнится запрос с выводом данных в таблицу с количеством одинаковых имен в справочнике.

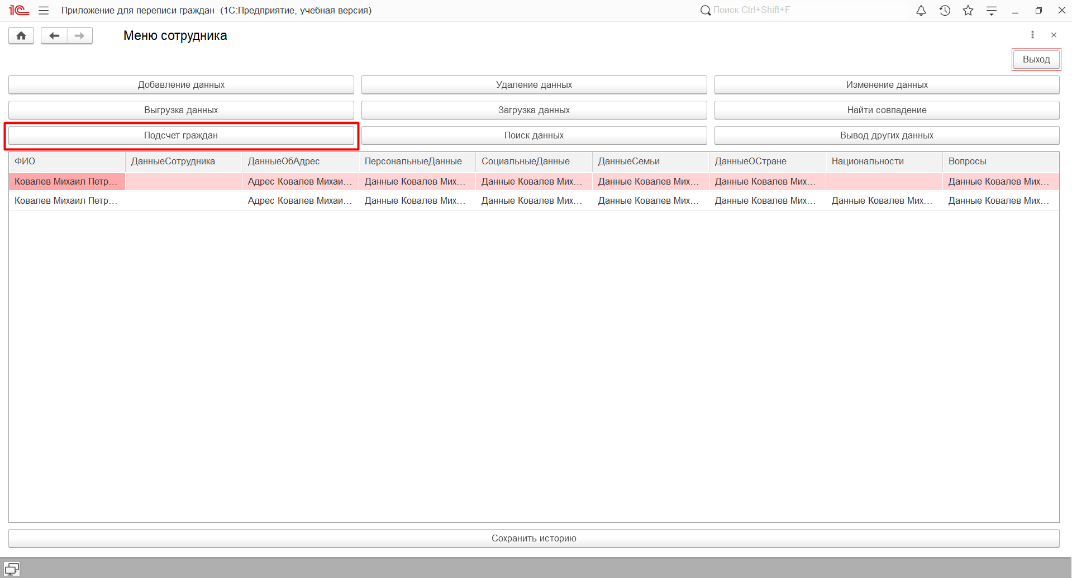
  
 Рисунок 48 – Кнопка «Найти совпадение» в меню сотрудника

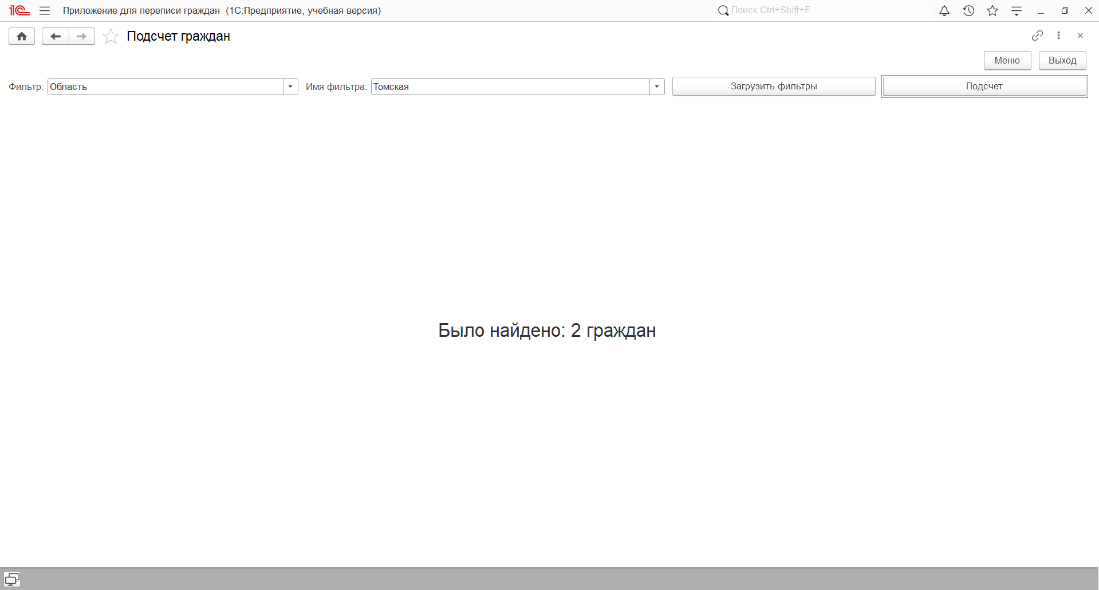
  
 Рисунок 49 - Окно функции «Найти совпадение» по выбранной таблице

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

Подсчет граждан по заданным фильтрам

После авторизации в системе пользователь может нажать на кнопку «Подсчет граждан» (Рисунок 50), которая есть в меню сотрудника и меню админа, откроется окно с функцией подсчет граждан по выбранным пользователем данным. Чтобы подсчитать граждан, необходимо выбрать фильтр для подсчета, но нужно загрузить значения, чтобы выбрать значение для поиска, нажав на кнопку «Загрузить фильтры», а после нажать кнопку «Подсчет». Результатом функции будет вывод сообщения о количестве найденных граждан по фильтру (Рисунок 51).

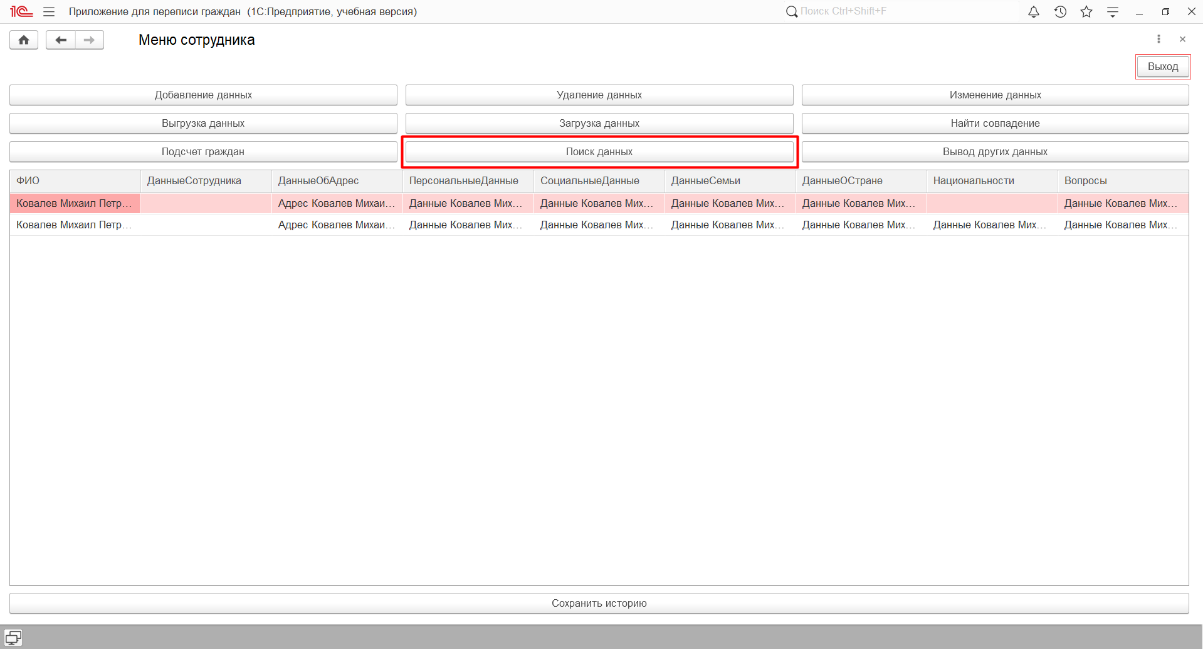
  
 Рисунок 50 – Кнопка «Подсчет граждан» в меню сотрудника

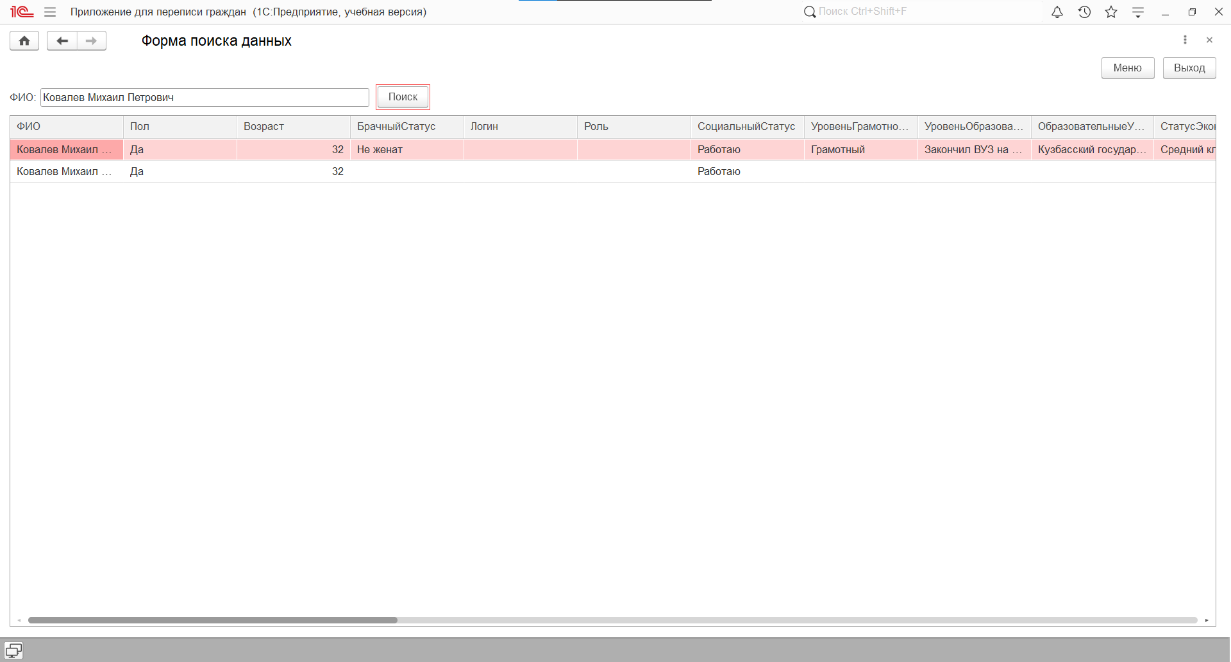
  
 Рисунок 51 – Окно функции «Подсчет граждан» с результатом

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

Поиск данных гражданина

После авторизации в системе пользователь может нажать на кнопку «Поиск граждан» (Рисунок 52), которая есть в меню сотрудника и меню админа, откроется окно с функцией поиска граждан по ФИО гражданина. Чтобы найти записи о гражданине, пользователю необходимо ввести полное имя гражданина, который записан в справочниках. Результатом функции будет список найденных данных граждан, которые совпадают по введенному ФИО (Рисунок 53).

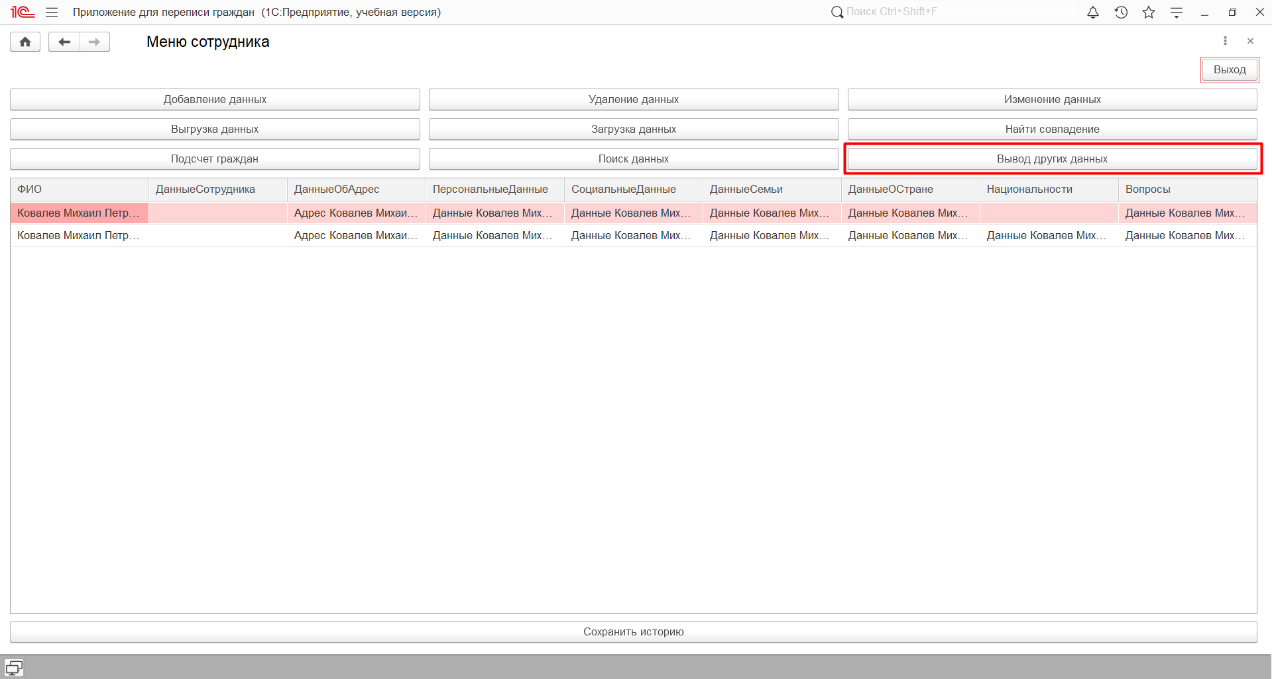
  
 Рисунок 52 – Кнопка «Поиск данных» в меню сотрудника

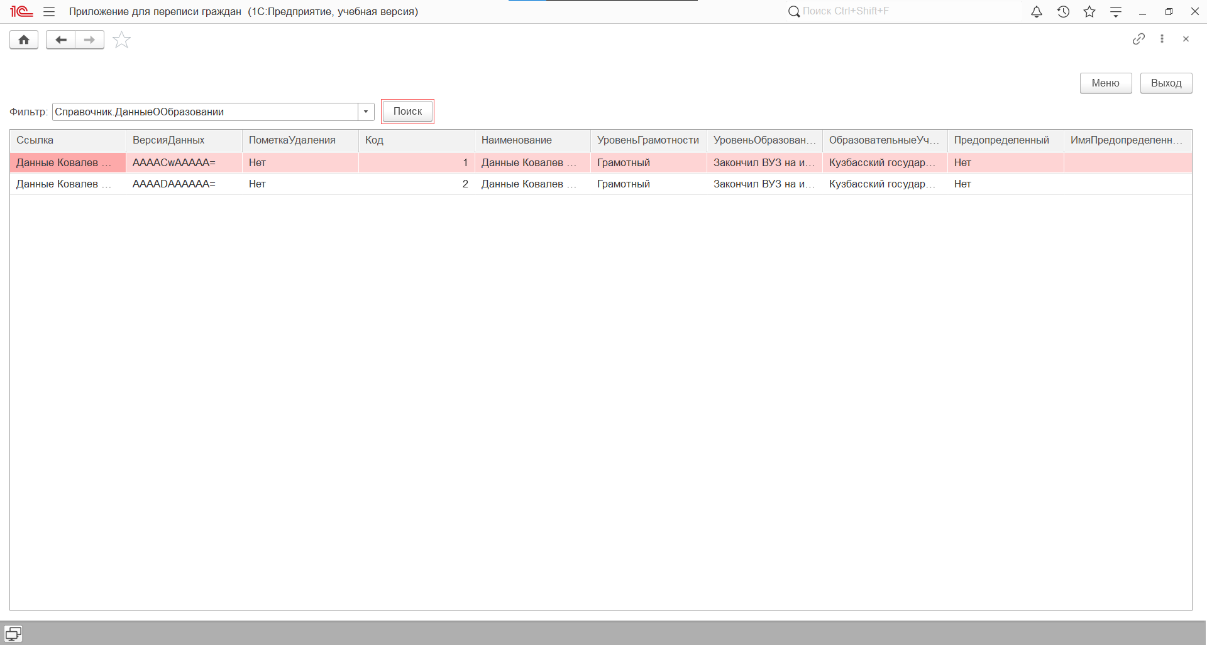
  
 Рисунок 53 – Окно функции «Поиск данных» с результатом

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

Вывод других данных

После авторизации в системе пользователь может нажать на кнопку «Вывод других данных» (Рисунок 54), которая есть в меню сотрудника и меню админа, откроется окно с функцией вывода данных справочников по выбранному фильтру пользователем. Чтобы вывести данные других справочников, пользователь должен выбрать в фильтре справочник для вывода всех значений этого справочника и нажать кнопку «Поиск» (Рисунок 55).

  
 Рисунок 54 – Кнопка «Вывод других данных» в меню сотрудника

  
 Рисунок 55 – Окно функции «Вывод других данных» с результатом

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

Сохранение истории действий

После авторизации в системе пользователь может нажать на кнопку «Сохранение истории» (Рисунок 56), которая есть в меню сотрудника и меню админа, будет совершена выгрузка записей действий в текстовый файл в той папке, где хранится файл с выгрузкой данных и выведется сообщение о выполненном сохранении действий (Рисунок 57-58).

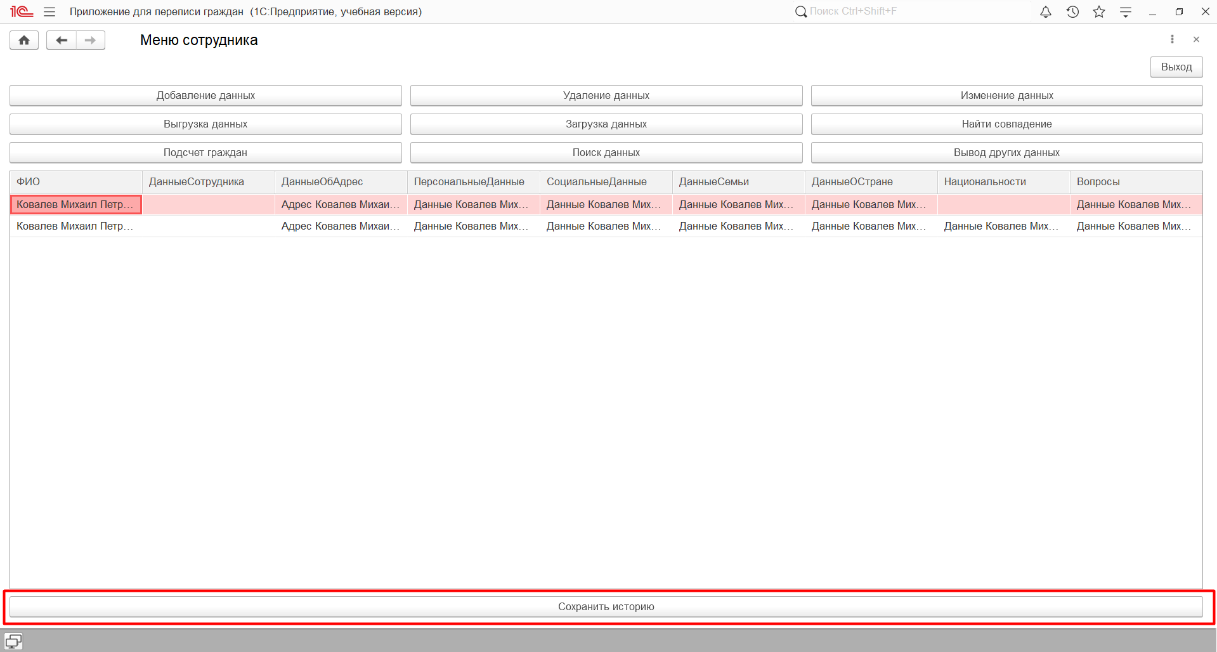
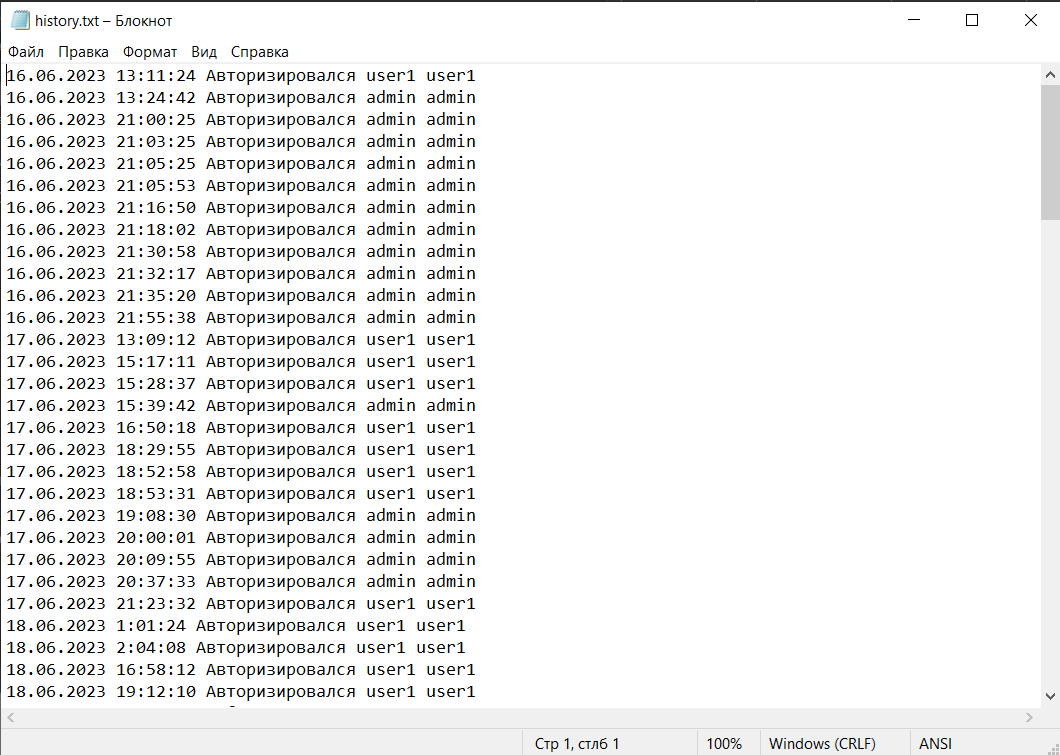
  
 Рисунок 56 – Кнопка «Сохранить историю» в меню сотрудника

  
 Рисунок 57 – Выполнение функции

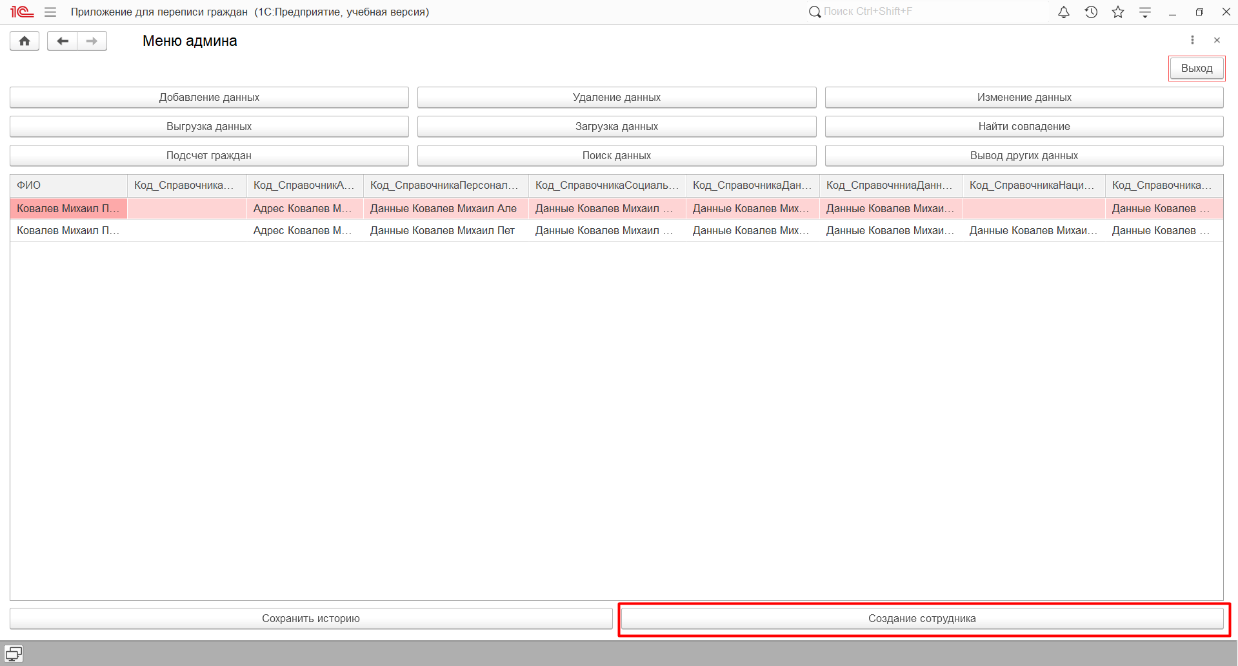
ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

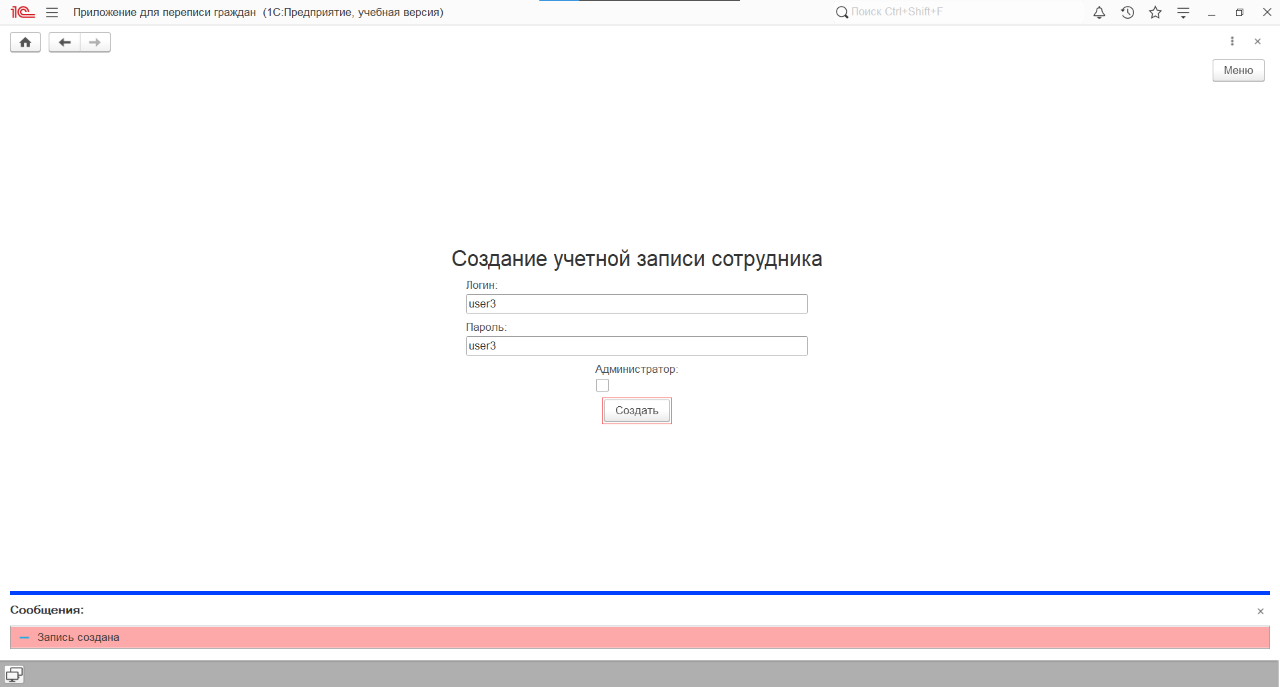
  
 Рисунок 58 – Результат функции

Создание учетных записей сотрудников

После авторизации в системе пользователь c ролью администратора может нажать на кнопку «Создание сотрудника» (Рисунок 59), которая есть только в меню админа, откроется окно с функцией создание учетной записи для сотрудника. Чтобы создать запись для сотрудника необходимо заполнить поле логина и пароля данными, которые не записаны в справочнике. После ввода данных пользователь-админ должен подтвердить запись, нажав кнопку «Создать» (Рисунок 60).

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

  
 Рисунок 59 – Кнопка функции «Создание сотрудника»

  
 Рисунок 60 – Окно функции «Создание учетной записи сотрудника» с результатом

ПРИЛОЖЕНИЕ В

Тесты

Таблица 12 – Тест функции «Авторизации»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_1 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Авторизация пользователя в систему. |
| Резюме испытания | Пользователь должен авторизоваться в систему при правильном логине и пароле. |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Ввод логина и пароля в соответствующие поля. 4. Нажатие на кнопку «Авторизация» |
| Тестовые данные | Логин: admin  Пароль: admin |
| Ожидаемый результат | Откроется окно меню администратора. |
| Фактический результат | Авторизация прошла успешно и открылось окно меню администратора. |
| Предпосылки | Поля не должны быть пустыми. |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | 1. Протестирована авторизация с неправильными данными. В результате программа выводит оповещение об ошибке авторизации. 2. Протестирована авторизация с пустыми полями или с вводом только пробелов. Система выводит ошибку авторизации. |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 13 – Тест функции «Добавление данных»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_2 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Добавление данных гражданином. |
| Резюме испытания | Данные, введенные пользователем в поля на окне функции, должны будут добавиться в базу данных. |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Нажать на кнопку «Ввод данных без авторизации» 4. Заполнение полей данными о пользователе 5. Нажать на кнопку «Сохранить» |
| Тестовые данные | ФИО: Токарев Михаил Сергеевич  Пол: Верно  Возраст: 25  Область: Московская  Регион: Московский  Населенный пункт: Москва  Улица: Центральная  Дом: 12 |
| Ожидаемый результат | Окно добавления данных гражданина закроется и откроется начальное окно |
| Фактический результат | Окно добавления данных гражданина закрылось и открылось окно авторизации |
| Предпосылки | Поле ФИО не должно быть пустое. |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | 1. Протестировано добавление с пустыми полями, результатом теста было закрытия окна и открытия окна авторизации и добавления записи в справочник. |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 14 – Тест функции «Удаление данных»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_3 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Удаление данных гражданина |
| Резюме испытания | Данные, которые были выбраны для удаления из базы данных, должны будут удалены из базы данных. |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Авторизоваться 4. Нажать на кнопку «Удаления данных» 5. Выбор данных для удалений, введя номер данных 6. Нажать на кнопку «Удалить» |
| Тестовые данные | Номер кода: 4 |
| Ожидаемый результат | Данные удалятся из базы данных и выведется сообщение об удаленных данных |
| Фактический результат | Данные удалились и вывелось сообщение об удалении данных |
| Предпосылки | Поле номер кода должен быть не пустой |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | Чтобы проверить удалились ли данные необходимо перезайти на окно функции, даже если номер кода этой записи не существует, то запись всё равно удалится. |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 15 – Тест функции «Изменение данных»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_4 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Изменение данных гражданина |
| Резюме испытания | Данные должны будут измениться, если данные были перезаписаны |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Авторизоваться 4. Нажать на кнопку «Изменение данных» 5. Ввод ФИО гражданина в поле поиска 6. Нажать кнопку «Найти» 7. Изменить данные, которые были найдены и введены в поля 8. Нажать на кнопку «Изменить» |
| Тестовые данные | ФИО для поиска: Ковалев Михаил Петрович |
| Ожидаемый результат | Данные перезапишутся на те данные, которые были записаны в поля после введения во время поиска данных и окно закроется |
| Фактический результат | Окно закрылось и в таблице граждан можно заметить перезаписанные данные |
| Предпосылки | Поле ФИО для поиска не должно быть пустым |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | При попытке изменить данные с пустым полем ФИО, выводится сообщение, чтобы пользователь сначала нашел данные гражданина. При поиске данных по ФИО выберутся первые встреченные данные. |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 16 – Тест функции «Выгрузка данных»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_5 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Выгрузка данных граждан по выбранному субъекту |
| Резюме испытания | Данные выбранного субъекта должны будут записаться в текстовый файл |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Авторизоваться 4. Нажать на кнопку «Выгрузка данных» 5. Выбрать область из выпадающего списка 6. Нажать на кнопку «Выгрузка» |
| Тестовые данные | Область: Томская |
| Ожидаемый результат | Выведется сообщение о выгрузке данных в текстовый файл и о количестве записей, создастся текстовый файл по пути, который будет записан в сообщении |
| Фактический результат | Вывелось сообщение и создался файл с данными |
| Предпосылки | Поле область не должно быть пустой |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | При попытке выгрузке данных по субъекту, когда поле область пустое, вывелось ошибка об невыбранной области. |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 17 – Тест функции «Загрузка данных»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_5 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Загрузка данных граждан файла |
| Резюме испытания | Данные файла, которые записаны по правилу, должны будут записаться в базу данных |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Авторизоваться 4. Нажать на кнопку «Загрузка данных» 5. Выбрать файл для загрузки данных 6. Нажать на кнопку «Загрузка» |
| Тестовые данные | Файл: C:\Users\Public\Documents\export.txt |
| Ожидаемый результат | Выведется сообщение о загруженных данных из текстового файла и о количестве записей и данные запишутся в базу данных |
| Фактический результат | Вывелось сообщение и записались данные в базу данных |
| Предпосылки | Поле файл не должно быть пустой |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | При попытке загрузке данных, когда поле файл пустое, вывелось ошибка о невыбранном файле. |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 18 – Тест функции «Найти совпадение»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_6 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Нахождение идентичных данных гражданина в разных населённых пунктах |
| Резюме испытания | Поиск совпадений данных граждан по выбранной таблице в базе данных |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Авторизоваться 4. Нажать на кнопку «Найти совпадение» 5. Выбрать таблицу для поиска 6. Нажать на кнопку «Найти» |
| Тестовые данные | Таблица: Справочник.ДанныеОбАдресе |
| Ожидаемый результат | По найденным данным создаться таблица с указанием идентичных данных и количество совпадений |
| Фактический результат | Появилась список с записями о совпадении |
| Предпосылки | Поле таблица не должно быть пустой |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | При попытке поиска данных, когда поле таблица пустое, вывелось ошибка о невыбранной таблицы. |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 19 – Тест функции «Подсчет граждан»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_7 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Подсчет граждан по заданным критериям сотрудника |
| Резюме испытания | Поиск совпадений данных граждан по выбранной таблице в базе данных |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Авторизоваться 4. Нажать на кнопку «Подсчет граждан» 5. Выбрать фильтр 6. Нажать на кнопку «Загрузить фильтры» 7. Выбрать имя фильтра 8. Нажать на кнопку «Подсчет» |
| Тестовые данные | Фильтр: Область  Имя фильтра: Томская |
| Ожидаемый результат | Выведется надпись о количестве найденных по фильтру данных граждан |
| Фактический результат | Отобразилась надпись с количеством найденных записей по данному фильтру |
| Предпосылки | Поле фильтра и имя фильтра не должны быть пустыми |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | При попытке подсчета граждан с пустым полем фильтра и имя фильтра вывелась ошибка о невыбранном фильтре и имени фильтре.  При попытке подсчета граждан только с пустым именем фильтра вывелась ошибка о невыбранном имени фильтра. |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 20 – Тест функции «Поиск данных»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_8 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Поиск данных гражданина |
| Резюме испытания | Поиск по ФИО и вывод полных данных гражданина из базы данных |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Авторизоваться 4. Нажать на кнопку «Поиск данных» 5. Ввести ФИО гражданина в поле ФИО 6. Нажать на кнопку «Поиск» |
| Тестовые данные | ФИО: Ковалев Владимир Петрович |
| Ожидаемый результат | Выведется список с данными об гражданине |
| Фактический результат | Отобразилась таблица с данными, которые связанны с введенным ФИО |
| Предпосылки | Поле ФИО не должны быть пустыми |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | При попытке поиска данных с пустым полем ФИО вывелось ошибка о пустом поле ФИО.  Если есть несколько записей с этим ФИО, то выведутся все записи. |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 21 – Тест функции «Вывод других данных»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_9 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Вывод остальной информации |
| Резюме испытания | Вывод всех данных выбранной таблицы базы данных |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Авторизоваться 4. Нажать на кнопку «Вывод других данных» 5. Выбрать таблицу в поле фильтр 6. Нажать на кнопку «Поиск» |
| Тестовые данные | Таблица: Справочник.ДанныеОбАдресе |
| Ожидаемый результат | Выведется список данных таблицы |
| Фактический результат | Отобразилась таблица данных |
| Предпосылки | Поле фильтр не должно быть пустым |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | При попытке поиска данных с пустым полем фильтра вывелось ошибка о пустом поле фильтра.  Если есть несколько записей с этим ФИО, то выведутся все записи. |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 22 – Тест функции «Сохранение истории»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_10 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Сохранение истории выполненных действий |
| Резюме испытания | Сохранение записей о совершенных записях в текстовый файл |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Авторизоваться 4. Нажать на кнопку «Сохранение истории» |
| Тестовые данные | Тестовых данных нет. |
| Ожидаемый результат | Создастся файл с данными о совершенных действиях в системе |
| Фактический результат | Вывело сообщение о завершении сохранения в файл, который был создан |
| Предпосылки | - |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | - |

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

Таблица 22 – Тест функции «Создание сотрудника»

|  |  |
| --- | --- |
| Номер тестирования | Тест\_11 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Создание новых учетных записей сотрудников |
| Резюме испытания | Создание новых учетных записей сотрудников и сохранение данных записей в базе данных |
| Шаги тестирования | 1. Запуск платформы «1С Предприятие 8» 2. Запуск нужной конфигурации 3. Авторизоваться под учетной записью админа 4. Нажать на кнопку «Создание сотрудника» |
| Тестовые данные | Логин: test1  Пароль: test1  Роль: Ложь |
| Ожидаемый результат | Данные данные запишутся в таблицу, где хранятся в базе данных данные о учетных записях сотрудников, и высветится сообщение о созданной записи |
| Фактический результат | Высветилось сообщение и запись была добавлена в базу данных |
| Предпосылки | Поле логин и пароль не должны быть пустыми |
| Статус (Зачет/Незачет) | Зачет |
| Комментарии | При попытке создать учетную запись с пустыми полями логина и пароля высветилось сообщении о пустых полях |