**Содержание**

[ВВЕДЕНИЕ 4](#_Toc130478188)

[1 ОЗНАКОМЛЕНИЕ С ПРЕДПРИЯТИЕМ 7](#_Toc130478189)

[1.1 Цели и задачи работы предприятия 7](#_Toc130478190)

[1.2 Аппаратные и технические средства, используемые на предприятие 7](#_Toc130478191)

[1.3 Программное обеспечение, используемое на предприятии 8](#_Toc130478192)

[1.4 Компьютерные сети и организация их работы на предприятии 9](#_Toc130478193)

[2 СОПРОВОЖДЕНИЕ И ОБСЛУЖИВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ 11](#_Toc130478194)

[2.1 Работа с операционной системой Windows 11](#_Toc130478195)

[2.1.1 Службы ОС 11](#_Toc130478196)

[2.1.2 Назначение, расположение и структура реестра 12](#_Toc130478197)

[2.1.3 Ключи реестра и их назначение 13](#_Toc130478198)

[2.1.4 Администрирование в 1С 15](#_Toc130478199)

[2.2 Работа с операционной системой Linux 17](#_Toc130478200)

[2.2.1 Пространство ядра и пространство пользователя 18](#_Toc130478201)

[2.2.2 Задачи администрирования в ОС Linux 19](#_Toc130478202)

[3 ПОДГОТОВИТЕЛЬНЫЙ ЭТАП РАЗРАБОТКИ ПРОГРАММНТОГО ПРОДУКТА 21](#_Toc130478203)

[3.1 Актуальность решаемой задачи 21](#_Toc130478204)

[3.2 Цели и задачи разработки 21](#_Toc130478205)

[3.3 Обзор существующих аналогов 22](#_Toc130478206)

[3.4 Функциональные требования к разрабатываемому программному продукту 22](#_Toc130478207)

[3.5 Постановка задачи на проектирование 23](#_Toc130478208)

[4 РАЗРАБОТКА ОТДЕЛЬНЫХ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА 24](#_Toc130478209)

[4.1 Построение инфологической модели 24](#_Toc130478210)

[4.2 Построение даталогической модели. 25](#_Toc130478211)

[4.3 Клиентское приложение 25](#_Toc130478212)

[ЗАКЛЮЧЕНИЕ 28](#_Toc130478213)

[СПИСОК ЛИТЕРАТУРЫ 29](#_Toc130478214)

[ПРИЛОЖЕНИЕ А 30](#_Toc130478215)

# **ВВЕДЕНИЕ**

Целью практики является овладение, общими и профессиональными компетенциями по специальности, видами профессиональной деятельности (ВПД).

ВПД «Сопровождение и обслуживание программного обеспечения компьютерных систем». В результате освоения программы практики обучающийся должен иметь практический опыт в настройке отдельных компонентов программного обеспечения компьютерных систем; выполнении отдельных видов работ на этапе поддержки программного обеспечения компьютерной системы.

ВПД «Проектирование и разработка информационных систем».   
В результате освоения программы практики обучающийся должен иметь практический опыт в: управлении процессом разработки приложений   
с использованием инструментальных средств; обеспечении сбора данных для анализа использования и функционирования информационной системы; программировании в соответствии с требованиями технического задания; использовании критериев оценки качества и надежности функционирования информационной системы; применении методики тестирования разрабатываемых приложений; определении состава оборудования   
и программных средств разработки информационной системы; разработке документации по эксплуатации информационной системы; проведении оценки качества и экономической эффективности информационной системы   
в рамках своей компетенции; модификации отдельных модулей информационной системы.

ВПД «Сопровождение информационных систем» В результате освоения программы практики обучающийся должен иметь практический опыт   
в инсталляции, настройке и сопровождении информационной системы; выполнении регламентов по обновлению, техническому сопровождению   
и восстановлению данных информационной системы.

Цель, которая ставится на практику, – приобретение реального опыта решения задач, увязав его с теоретическими знаниями, полученными   
в колледже.

Задачи производственной практики:

* формирование более полного и реального представления о своей профессии;
* наработка практических навыков в разных областях применения информационных технологий;
* изучение способов, используемых при организации технологического процесса обработки информации в сфере работы предприятия – базы практики.
* развитие профессионального мышления;
* изучение и освоение программных систем, пакетов прикладных программ, специализированных программных продуктов.
* проанализировать особенности социального обслуживания в Центрах социального обслуживания;
* рассмотреть виды, характер, цели деятельности и структуру Центров социального обслуживания.

Производственная практика проходила в «Центре социального обслуживания» (далее – Центр). Центр социального обслуживания является относительно самостоятельным элементом государственной системы социального обеспечения. В обобщенном виде он представляет собой комплекс разно - образных услуг и выдач, предоставляемых престарелым, инвалидам, нетрудоспособным, семьям с детьми в «натуральной» форме.

Актуальность данной темы возрастает с каждым годом,   
так как количество граждан, нуждающихся в социальной помощи, постоянно увеличивается.

Уровень жизни в России значительно ниже, чем в большинстве других развитых странах мира. Много людей становятся инвалидами в молодом возрасте, пожилые люди рано начинают испытывать нужду в посторонней помощи при самообслуживании. Центры социального обслуживания играют все большую роль в оказании адресной помощи пенсионерам и инвалидам, предоставлении им социальных услуг в соответствии с индивидуальной нуждаемостью

# **1 ОЗНАКОМЛЕНИЕ С ПРЕДПРИЯТИЕМ**

## **1.1 Цели и задачи работы предприятия**

Целью деятельности Центра является оказание социальных, социально педагогических, юридических, психологических, социально - медицинских, бытовых, консультативных и иных услуг гражданам пожилого возраста и инвалидам, частично утратившим способность к самообслуживанию, а также другим категориям населения, нуждающимся в социальной поддержке.

Центр осуществляет в соответствии с муниципальным заданием деятельность, связанную с выполнением работ, оказанием услуг, относящихся к его основным видам деятельности, определяемыми правовыми актами и уставом. Центр не вправе отказаться от выполнения муниципального задания.

Основными задачами деятельности Центра являются:

* социальное обслуживание населения на дому;
* полустационарное социальное обслуживание;
* срочное социальное обслуживание;
* оказание консультативной помощи;
* содействие в оказании материальной помощи;
* проведение социальной адаптации и реабилитации граждан, оказавшихся в трудной жизненной ситуации.

Социальное обслуживание осуществляется структурными подразделениями Центра в соответствии с государственными стандартами социального обслуживания, утверждёнными законами и иными правовыми актами.

## **1.2 Аппаратные и технические средства, используемые на предприятие**

Компьютер — основное устройство для реализации информационных технологий, выполняющее совокупность заранее заданных операций с очень высокой скоростью. Он получает исходные данные, инструкции по их обработке, обрабатывает данные соответственно инструкциям и предоставляет результат в форме, которая может быть прочтена и использована человеком или другим компьютером.

Состав аппаратно-технических средств предприятия Центра социального обслуживания:

1. Системные блоки: Процессор Intel Core i5-3470; Видеокарта Intel HD Graphics 2500 ОЗУ 4 Гб; HDD 500 Гб
2. Мониторы: размер экрана – 27 дюймов; яркость: 300 кд/мІ; Соотношение сторон 16:9; Частота обновления кадров 75 Гц; Разрешение 1920×1080.
3. Wi-fi Роутер: Частоты 2.4 ГГц; Порты: 10/100BASE-TX х 4
4. Принтеры: Вид печати: черно-белая; Скорость ч/б печати (A4),   
   стр/мин: 22.

## **1.3 Программное обеспечение, используемое на предприятии**

Программное обеспечение (ПО) - совокупность программ, позволяющая организовать решение задач на компьютере. ПО и архитектура машины образуют комплекс взаимосвязанных и разнообразных функциональных средств, определяющих способность решения того или иного класса задач. Важнейшими классами ПО являются системное и специальное (прикладное), представленное пакетом прикладных программ (ППП).

Системное программное обеспечение - это совокупность программ, обеспечивающих: создание операционной среды функционирования других программ, надежную и эффективную работу компьютера и компьютерных сетей, проведение диагностики и профилактики аппаратуры компьютера и сетей.

Системное программное обеспечение организует процесс обработки информации в компьютере. Главную его часть составляет операционная система (ОС). На предприятии используется ОС Windows 10.

Windows 10 операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT.

Прикладное программное обеспечение - предназначено для решения конкретных задач пользователя и представляет уровень программного обеспечения, обращенный к человеку, который сам не разрабатывает программы, а лишь использует их в своей деятельности.

На каждом рабочем компьютере сотрудника компании установлены такие программы как:

* + Microsoft 365 офисный пакет приложений, созданных корпорацией Microsoft для операционных систем Windows, Windows Phone, Android, macOS, iOS.
  + Яндекс поисковая система и интернет-портал. Поиск по интернету, Алиса и другие сервисы: на ya.ru есть карты, транспорт, погода, музыка, такси, переводчик, электронная почта и облачное хранилище.
  + WinRAR архиватор файлов для 32- и 64-разрядных операционных систем Windows, позволяющий создавать, изменять и распаковывать архивы RAR и ZIP.

Данный набор прикладных программ позволяет решать все поставленные задачи предприятия.

## **1.4 Компьютерные сети и организация их работы на предприятии**

Локальная сеть - это компьютерная сеть, в которой ЭВМ расположены на небольшом расстоянии друг от друга, не использующая средства связи общего назначения типа телефонных сетей.

Компьютеры локальной сети используют единый комплект протоколов для всех участников. По территориальному признаку локальные сети отличаются компактностью. Они могут объединять компьютеры одного помещения, этажа, здания, компактно расположенных сооружений. Группы сотрудников, работающих над одним проектом в рамках локальной сети, называются рабочими группами. В рамках одной локальной сети могут работать несколько рабочих групп. У участников рабочих групп могут быть разные права для доступа к общим ресурсам сети. Совокупность приемов разделения и ограничения прав участников компьютерной сети называется политикой сети. Управление сетевыми политиками называется администрированием сети. Лицо, управляющее организацией работы участников локальной сети, называется системным администратором.

Простейшее устройство для соединения между собой двух локальных сетей, использующих одинаковые протоколы, называется мостом. Мост может быть аппаратным или программным. Цель моста - не выпускать за пределы локальной сети данные, предназначенные для внутреннего использования.

На предприятии «Центр социального обслуживания» используется локальная сеть. Особенностью является то, что сеть работает по принципу клиент-сервер. Сеть типа клиент-сервер — это вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.

# **2 СОПРОВОЖДЕНИЕ И ОБСЛУЖИВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ**

## **2.1 Работа с операционной системой Windows**

### **2.1.1 Службы ОС**

Службы ОС – это приложения, которые работают в фоновом режиме и предлагают свои «услуги». Эти службы выполняют большинство основных функций Windows, например подключение к внешним устройствам или создание стабильного подключения к Интернету. Как и любое другое приложение, эти службы также используют системные ресурсы.

Главные службы ОС:

* Центр обновления Windows. Данное средство обеспечивает обновление системы. Без его запуска невозможно будет обновить ОС ни автоматически, ни вручную, что, в свою очередь, приводит к её моральному устареванию, а также к образованию уязвимостей. Именно «Центр обновления Windows» ищет апдейты для операционной системы и установленных программ, а после этого производит их инсталляцию. Поэтому данная служба считается одной из важнейших. Ее системное имя — «Wuauserv».
* DHCP-клиент. Следующей важной службой является «DHCP-клиент». Её задачей является регистрация и обновление IP-адресов, а также DNS-записей. При отключении этого элемента системы компьютер не сможет выполнять указанные действия. Это означает, что серфинг по просторам интернета для пользователя станет недоступен, а также пропадет возможность производить другие сетевые подключения (например, по локальной сети). Системное наименование объекта предельно простое – «Dhcp».
* DNS-клиент. Ещё одна служба, от которой зависит работа ПК в сети, называется «DNS-клиент». В её задачу входит кэширование имен DNS. При её остановке получение DNS-имен будет продолжаться, но при этом результаты очередей в кэш не попадут, а это значит, что имя ПК зарегистрировано не будет, что опять же ведет к проблемам сетевого подключения. Кроме того, при отключении элемента «DNS-клиент» все связанные с ним службы включить также не получится. Системное наименование указанного объекта «Dnscache».
* Журнал событий Windows. Обеспечивает протоколирование событий в системе, производит их архивацию, обеспечивает хранение и доступ к ним. Отключение этого элемента повысит уровень уязвимости системы, так как значительно затруднит вычисление ошибок в ОС и определение их причин. Внутри системы идентифицируется под наименованием - «eventlog».

### **2.1.2 Назначение, расположение и структура реестра**

Реестр можно рассматривать как записную книжку Windows - как только системе нужна какая-то информация, то она ищет ее в реестре.

Реестр выполняет следующие основные функции:

* отслеживает все системные устройства и их установки, включая такие ресурсы, как запросы на прерывания (IRQ) и номера каналов прямого доступа памяти (DMA);
* работает как база данных, которая унифицирует функционирование приложений;
* проверяет наличие необходимых драйверов для инсталлируемого оборудования. При добавлении нового периферийного устройства Диспетчер конфигурации (Configuration manager) операционной системы помещает конфигурационные данные устройства в реестр;
* предоставляет системные сервисы, которые необходимы для работы многих приложений;
* обеспечивает запуск необходимого приложения при щелчке мышью;
* сохраняет информацию, относящуюся к системным правилам, профилям пользователей и средствам администрирования.

Реестр представляет собой иерархическую структуру, состоящую из поддеревьев, разделов, подразделов и параметров. Верхний уровень иерархии системного реестра составляют поддеревья (ветви). Поддеревья являются пятью важнейшими частями реестра.

Второй ступенью в иерархической системе реестра являются так называемые разделы, или ключи (Keys). Ключи отображаются в программе редактор реестра (RegEdit) в виде подпапок ветвей HKEY\_.

Ступенькой ниже в структурной иерархии реестра расположены подразделы реестра (subkeys). Подразделы прямо не связаны с какими-либо типами данных и не используются в рамках каких-либо соглашений, которые ограничивают присвоение им названий.

### **2.1.3 Ключи реестра и их назначение**

Ключи в иерархии реестра служат исключительно для облегчения доступа к информации и являются одним из средств ее упорядочивания. По своему функциональному предназначению ключи реестра разделяются на две следующие категории:

1) Указываются системой. Имена ключей выбираются ОС, их изменение может сделать Windows полностью неработоспособной.

2)Указываются пользователем. Имена ключей может изменять администратор компьютера, и такие модификации не станут причиной каких-либо фатальных проблем.

Всего существует пять ключей, название каждого из которых начинается с английской аббревиатуры HKEY\_. За символом подчеркивания следует название ветви:

* HKEY\_CLASSES\_ROOT (HKCR). Этот раздел включает в себя ряд подразделов, в которых содержатся сведения о расширениях всех зарегистрированных в системе типов файлов и данные о COM-серверах, зарегистрированных на компьютере. Данные этого раздела нужны при открытии файлов по двойному щелчку мыши или операций drag-and-drop. Кроме того, раздел HKEY\_CLASSES\_ROOT предоставляет объединенные данные программам, написанным под ранние версии Windows.
* HKEY\_CURRENT\_USER (HKCU). Эта ветвь реестра хранит настройки персональной оболочки пользователя, совершающего вход в операционную систему (меню «Пуск», рабочий стол и т. д.). В ее подразделах находится информация о переменных окружения, группах программ данного пользователя, настройках Рабочего стола, цветах экрана, сетевых соединениях, принтерах и дополнительных настройках приложений. Эта информация берется из подраздела Security ID (SID) ветви HKEY\_USERS для текущего пользователя. Фактически, в данной ветви собраны все сведения, относящиеся к профилю пользователя, работающего с Windows в настоящий момент.
* HKEY\_LOCAL\_MACHINE (HKLM). Раздел содержит информацию об установленном программном обеспечении, его настройках, драйверах. Здесь же – информация, относящаяся к операционной системе и оборудованию, например тип шины компьютера, общий объем доступной памяти, список загруженных в данный момент времени драйверов устройств, а также сведения о загрузке Windows. Данная ветвь включает наибольшее количество информации в системном реестре и нередко используется для тонкой настройки аппаратной конфигурации компьютера. Хранящиеся в этой ветви данные справедливы для всех профилей, зарегистрированных в системе пользователей
* HKEY\_USERS (HKU). Если раздел HKEY\_CURRENT\_USER содержит настройки текущего пользователя, то данный раздел хранит настройки Windows для всех пользователей. В его подразделах содержится информация обо всех профилях пользователей, а один из подразделов всегда соотносится с разделом HKEY\_CURRENT\_USER (через параметр Security ID (SID) пользователя). Другой подраздел, HKEY\_USERS\DEFAULT, содержит информацию о настройках системы в момент времени, предшествующий началу сеанса текущего пользователя.
* HKEY\_CURRENT\_CONFIG (HKCC). В этой ветви содержится информация о профиле оборудования, используемом локальным компьютером при запуске системы. Профили оборудования позволяют выбрать драйверы поддерживаемых устройств для заданного сеанса работы.

### **2.1.4 Администрирование в 1С**

Конфигуратор — один из двух режимов работы системы 1C:Предприятие. В этом режиме разрабатываются прикладные решения и выполняется администрирование информационных баз. Фактически, администрирование — это вторая профессия Конфигуратора, входящего в стандартную поставку. Которое включает в себя ведение списка пользователей, настройку прав доступа, резервное копирование информационной базы, мониторинг системных событий (журнал регистрации настраивается в конфигураторе; просматривается как в конфигураторе, так и в предприятии) и выполнение других действий для поддержания работоспособности системы. Основные инструменты администратора включены в подменю Администрирование.

Система 1С:Предприятие значительно облегчает администрирование информационной базы, предоставляя для этого все необходимое. К задачам по администрированию можно отнести следующие:

* установку и обновление системы,
* ведение списка пользователей,
* настройку прав доступа на основе механизма ролей,
* мониторинг действий пользователей и системных событий,
* выгрузка и загрузка информационной базы в файл,
* резервное копирование,
* тестирование и исправление информационной базы,
* установку региональных настроек,
* обновление конфигураций.

Основные средства администрирования системы 1С:Предприятие реализованы в составе конфигуратора. Однако есть ряд механизмов и утилит, которые не входят в состав конфигуратора, хотя также имеют отношение к администрированию системы 1С:Предприятие.

Административная установка системы предназначена для установки системы 1С:Предприятие в локальной сети. При использовании такого режима администратор локальной сети получает возможность устанавливать 1С:Предприятие с единого источника в сети, а затем обновлять версию 1С:Предприятия только в месте административной установки. Дальнейшее обновление версий на локальных рабочих местах смогут произвести сами пользователи.

Для выполнения административной установки необходимо запустить программу установки setup.exe с ключом /а. После запуска на экран выводится диалог, в котором необходимо указать каталог административной установки. В указанном каталоге будет создан набор файлов, необходимый для установки на локальных рабочих местах. Для установки системы 1С:Предприятие на локальных рабочих местах необходимо запустить программу setup.exe из каталога административной установки.

## **2.2 Работа с операционной системой Linux**

Linux — это семейство операционных систем (ОС), работающих на основе одноименного ядра. Нет одной операционной системы Linux, как, например, Windows или MacOS. Есть множество дистрибутивов (набор файлов, необходимых для установки ПО), выполняющих конкретные задачи.

Достоинства:

* Open Source — это свобода программного обеспечения, то есть, никаких платных программ нет. Дело в том, что ядро Linux и любого его дистрибутива имеет открытый исходный код. Благодаря этому, для каждого пользователя становится возможным самостоятельно улучшить и модернизировать работу любой программы.
* Безопасность. Пользоваться Linux — значит заботиться о безопасности своего компьютера. Дело в том, что на него практически нет вирусов, а значит и угрозы со стороны вредоносного и опасного ПО.
* Созданная специально для программистов. Встроенные компиляторы, консоли, возможность создавать собственные скрипты и многое другое.
* Значительно меньше различных тормозов и ошибок.
* Простая и понятная установка. Говорить о том, что установка Linux легче, чем установка Windows нельзя, но если это будет выполнять уверенный пользователь ПК, то проблем не возникнет.
* Требует значительно меньших системных ресурсов.
* Поддержка драйверов для большинства видов устройств: модемов, USB-устройств, периферии.
* Недостатки, которые определяют слабые места Linux:
* Освоения. Новичку будет очень трудно освоить совсем другую и неизвестную для него систему.
* Консоль. К сожалению, без нее невозможно обойтись. Любые действия, связанные с запуском, установкой или удалением программ нужно приписывать определенный код в консоли.
* Настройка принтеров. Если все остальные внешние устройства настраиваются без проблем и корректно работают, то с принтерами возникает большая проблема. Конечно, уже сегодня существует множество решений такой проблемы, и все же она остается очень актуальной.
* Права доступа. Для большинства программ и утилит каждый раз придется сталкиваться с правами доступа. Все сделано с целью безопасности.
* Очень много отличий по сравнению с Windows. Тому, кто перешел на Linux, сначала будет трудно привыкнуть к новому интерфейсу и визуальной оболочки.

### **2.2.1 Пространство ядра и пространство пользователя**

Для 32-разрядной операционной системы ее адресное пространство (виртуальное адресное пространство или линейное адресное пространство) составляет 4G (2 в 32-й степени). Другими словами, максимальное адресное пространство процесса составляет 4G. Ядром операционной системы является ядро, которое не зависит от обычных приложений и может обращаться к защищенному пространству памяти, а также ко всем разрешениям на доступ к базовым аппаратным устройствам. Чтобы обеспечить безопасность ядра, современные операционные системы обычно вынуждают пользовательские процессы не работать с ядром напрямую. Конкретная реализация заключается в том, что операционная система делит виртуальное адресное пространство на две части, одна часть — это пространство ядра, а другая часть - пользовательское пространство.

В адресном пространстве 4G каждого процесса самый высокий 1G совпадает, то есть пространство ядра. Только оставшийся 3G используется самим процессом.

### **2.2.2 Задачи администрирования в ОС Linux**

Кратко рассмотрим задачи администрирования в Linux:

* Удаленный доступ;
* Диагностика сети;
* Мониторинг ресурсов системы;
* Проверка работоспособности сервисов;
* Просмотр логов;
* Установка программного обеспечения.

Удаленный доступ к серверу Linux. Чаще всего веб-мастера и администраторы используют для удаленного доступа и загрузки файлов на сервер протокол SSH и FTP. По SSH вы не только передавать файлы, но и выполнять на сервере различные команды Linux.

Диагностика сети Linux. Этот момент администрирования Linux серверов подходит больше для компьютеров, к которым у вас есть физический доступ, но может в некоторых случаях полезным и на сервере. Самый простой способ проверить есть ли доступ к сети на компьютере, это выполнить команду ping.

Мониторинг ресурсов системы. Часто может случиться, что сервер начинает работать очень медленно, веб-службы начинают очень долго отвечать на запросы и даже соединение по SSH работает медленно. Скорее всего, причиной этому может стать перегрузка ресурсов процессора или памяти. Если вся память будет занята, система будет сбрасывать данные на диск, в раздел подкачки, что тоже сильно замедляет работу сервера. Чтобы посмотреть сколько памяти осталось доступно используйте команду free.

Проверка работоспособности сервисов. В системное администрирование linux также входит управление сервисами. Сейчас в большинстве дистрибутивов, в качестве системы инициализации используется systemd. Соответственно, управление службами linux выполняется с помощью нее.

Просмотр логов. Если какой-либо сервис или системный компонент не работает, то первое что нужно сделать — это смотреть логи. Если не помогает - включить режим отладки и смотреть логи. В 90% вы найдете ответ почему ничего не работает в логах программы. Логи всех служб и системные логи находятся в папке /var/log/.

Установка программ. Обычно применяют два основных пакетных менеджера, которые позволяют установить софт из депозитариев. В CentOS используют yum, в Ubuntu - apt. Чтобы установить программу, нужно набрать install. Чтобы удалить - remove. И не стоит забывать про автоматическое обновление программ. Ведь со временем находятся уязвимости, которые срочно нужно устранить.

# **3 ПОДГОТОВИТЕЛЬНЫЙ ЭТАП РАЗРАБОТКИ ПРОГРАММНТОГО ПРОДУКТА**

## **3.1 Актуальность решаемой задачи**

Актуальность темы исследования данной работы определяется тем, что в обществе существуют определенные категория граждан, которые в силу ряда объективных причин нуждаются в помощи со стороны других лиц. В любом социальном государстве руководство страны заботится о таких людях, организует систему социального обслуживания для таких категорий граждан.

Социальное обслуживание незащищенных категорий граждан представляет собой одну из самых основных функций и задач любого государства. Во много развитие страны определяется именно уровнем того, насколько полноценно в нем организована система социального обслуживания таких категорий граждан как инвалиды, пенсионеры и иные нуждающиеся в заботе и внимании со стороны категории людей.

## **3.2 Цели и задачи разработки**

Целью практической работы является создания информационной системы для предприятия «Центр социального обслуживания». Информационная система должна обладать следующими возможностями:

* + улучшение эффективность и производительность предприятия
  + улучшить коммуникации на предприятии
  + упростить и ускорить работу сотрудников
  + обработка текстов на компьютерах с помощью различных текстовых процессоров
  + оказывать сбор, хранение, обработку и предоставление информацию о подопечных и услугах

## **3.3 Обзор существующих аналогов**

В настоящее время существует не большое количество аналогов информационных систем в этой сфере, это связанно с тем, что данная тема в информационных технологиях находиться на стадии формирования и не приобрело достаточной популяции. Существующие аналоги:

* «УЧЁТ КЛИЕНТОВ ЦСО» - комплексное решение для управления деятельностью ЦСО. Программа позволяет облегчить работу социальным и медицинским работникам, заведующим отделениями ЦСО, специалистам по социальной работе, отвечающим за ведение личных дел клиентов и подготовку отчётности.
* Мобильное приложение «Мобильный центр социальных услуг». Установив приложение, гражданин сможет выбрать и оперативно заказать необходимые социальные услуги в ближайшем социальном учреждении по всей Московской области. Приложение «МЦСУ» позволяет оперативно отыскать нужную услугу в соответствии с индивидуальной потребностью пользователя и способствует повышению информированности граждан старшего возраста и инвалидов о социальных учреждениях и предоставляемых ими социальных услугах, преодолению информационно-коммуникативных барьеров, вызванных нарушениями функций здоровья, созданию комфортной информационной среды, а также позволит расширить аудиторию получателей социальных услуг.

## **3.4 Функциональные требования к разрабатываемому программному продукту**

Функциональные требования определяют, каким должно быть поведение продукта в тех или иных условиях. Они определяют, что разработчики должны создать, чтобы пользователи смогли выполнить свои задачи

В информационной системе должны быть следующие функциональные требования:

* надежное хранение и доступ к информации базы данных
* ввод и редактирование данных о подопечных
* ввод и редактирование записей на оказание услуг
* ввод и редактирование данных об услугах
* ввод и редактирование данных о сотрудниках
* поиск по записям, услугам, сотрудникам
* учет по оказанию услуг
* информационная система должна иметь удобный и понятный пользователю интерфейс

## **3.5 Постановка задачи на проектирование**

Для создания базы данных необходимо проанализировать данную информацию, создать на ее основе необходимые таблицы, установить определенные связи между ними. Поэтому задачи, которые нужно будет решить в процессе анализа, заключаются в разработке связей между таблицами базы данных, устранении избыточности в таблицах, их обработке и редактировании. Требуется сформировать инфологическую модель и даталогическую модель базы данных, а также создать клиентское приложение.

Задачи проектирования:

* выявить сущности и описать их
* сформировать даталогическую и инфологическую модель базы данных
* сформировать таблицы и связи между ними
* сформировать хранимые процедуры, запросы

# **4 РАЗРАБОТКА ОТДЕЛЬНЫХ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА**

## **4.1 Построение инфологической модели**

Инфологическая модель данных - обобщенное неформальное описание создаваемой базы данных, выполненное с использованием естественного языка, математических формул, таблиц, графиков и других средств, понятных всем людям, работающих над проектированием базы данных.

Инфологическая модель имеет следующий вид:

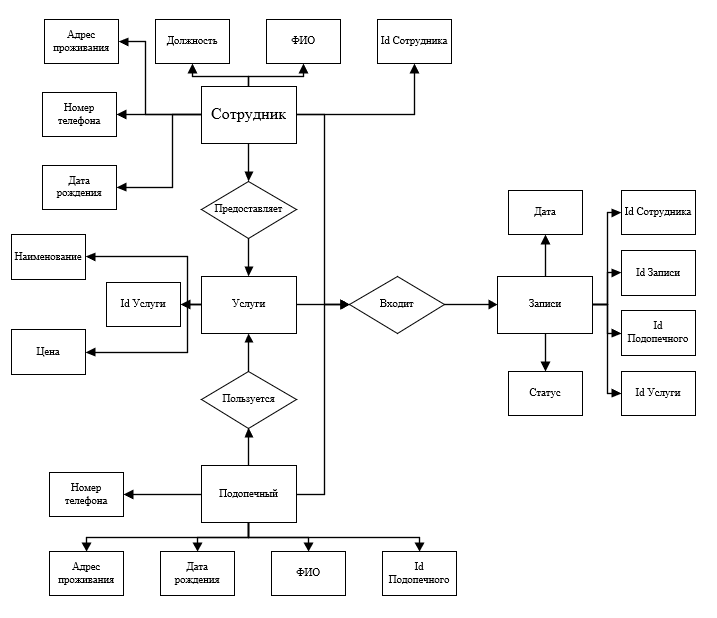


Рисунок 4.2.1 – Инфологическая модель

## **4.2 Построение даталогической модели.**

Даталогическая модель — это модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации.

Даталогическая модель имеет следующий вид:

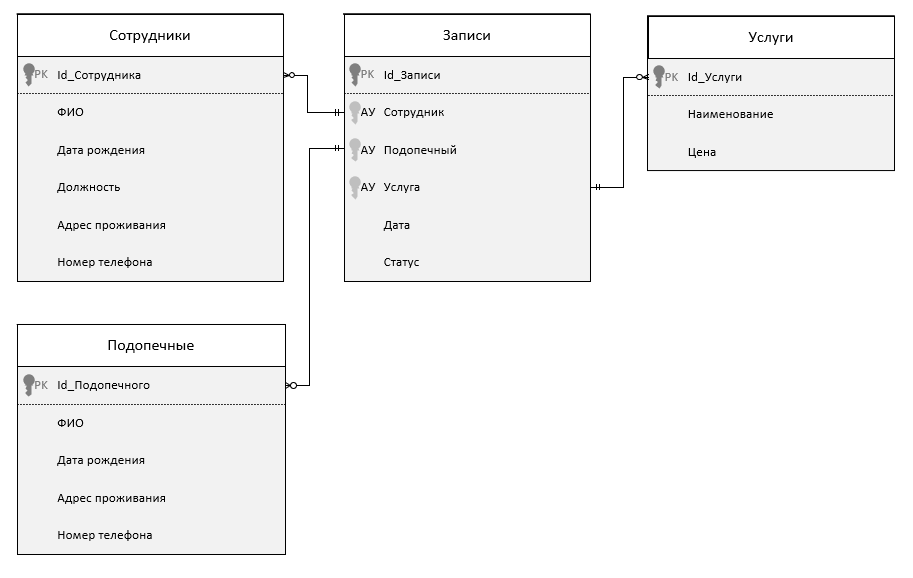


Рисунок 4.2.2 – Даталогическая модель

## **4.3 Клиентское приложение**

В качестве инструмента для реализации клиентского приложения была выбрана интегрированная среда разработки Microsoft Visual Studio и интерфейс программирования приложений Windows Forms.

Microsoft Visual Studio – это линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов.

Windows Forms – это интерфейс программирования приложений, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework.

Главная форма клиентского приложения. На главной форме расположены несколько вкладок с информацией. Каждая вкладка имеет табличную область со структурированной информацией, тип информации в табличной области зависит от выбранной вкладки. Так же на форме расположены кнопки редактирования табличной части, обновления формы, поле для поиска информации, панель ввода данных. На рисунке (Рис 4.3.1) отображена вкладка с записями, остальные вкладки выполнены в аналогичной структуре.

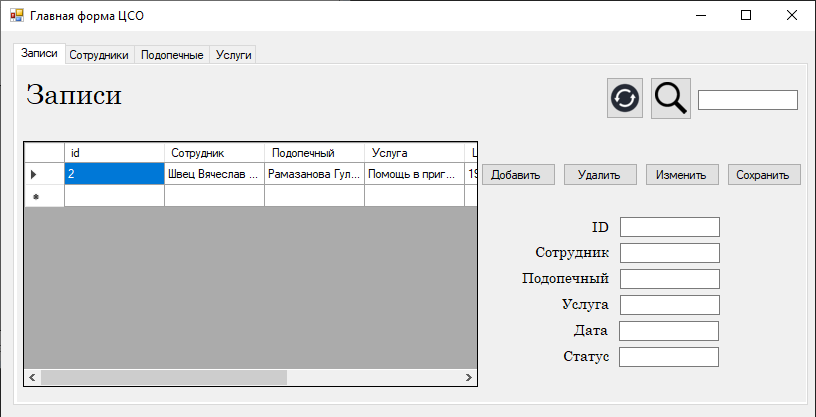


Рисунок 4.3.1 – Главная форма ЦСО, вкладка «Записи»

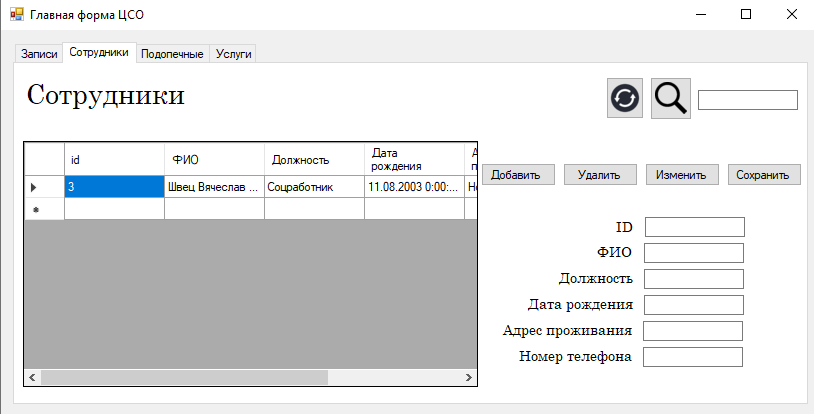


Рисунок 4.3.1 – Главная форма ЦСО, вкладка «Сотрудники»

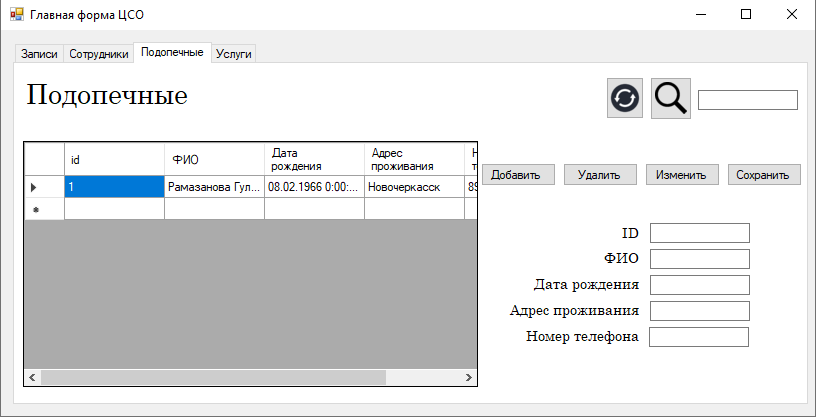


Рисунок 4.3.1 – Главная форма ЦСО, вкладка «Подопечные»

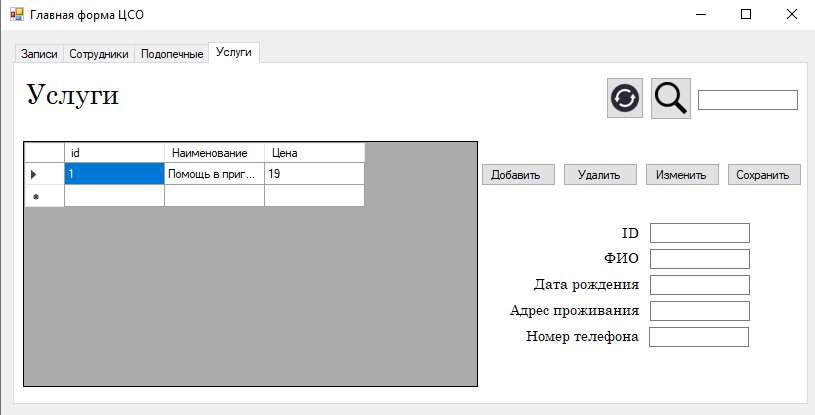


Рисунок 4.3.1 – Главная форма ЦСО, вкладка «Услуги»

# **ЗАКЛЮЧЕНИЕ**

Анализируя знания и опыт, полученные в период прохождения производственной практики, учитывая непростую специфику нашей работы, полагаю, что результат ее должно видеть в оказании помощи, поддержки и защите всех людей и особенно «слабых» слоев населения.

Профессия социального работника у нас в России одна из самых молодых. Отсутствуют традиции, практически нет опыта, как положительного, так и отрицательного. Начинает складываться система специального образования, но, к сожалению, по большей части, в отрыве от реально практикующих социальных служб, научные разработки зачастую выполняются людьми, не работавшими в социальной сфере.

Сейчас можно утверждать, что социальная работа в России завершила этап становления и активно развивается, опираясь на достижение естественных, социальных и общественных наук. Созданная сеть социальных учреждений призвана решать актуальные личностные проблемы на всех уровнях общественной жизни. Но неоспоримой является и необходимость повышения качества социального обслуживания населения, и совершенствование механизмов предоставления социальных услуг.

Важно налаживать обмен опытом, накапливать  базу данных, создавать условия для плодотворных творческих дискуссий, изучать и распространять любой полезный и интересный опыт.

# 

# **СПИСОК ЛИТЕРАТУРЫ**

1. Джеймс, Р. Грофф SQL. Полное руководство / Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. - М.: Вильямс, 2014. [Текст] - 960 c. Режим доступа: https://codernet.ru/books/sql/sql\_polnoe\_rukovodstvo \_trete\_izdanie/ (дата обращения 22.03.2023)
2. Брайан Нойес Привязка данных с помощью Windows Forms 2.0: Программирование интеллектуальных клиентских приложений для обработки данных с помощью .NET (серия разработки Microsoft .Net); СПб. [и др.] : Питер - Москва, 2006. - 736 c
3. Федеральный закон Российской Федерации "О социальном обслуживании граждан пожилого возраста и инвалидов" от 2 августа 1995 г. N 122-ФЗ (ред. от 25.11.2013 с изменениями, вступившими в силу с 25.11.2013) // Российская газета. - 1995. - 04 августа. - С. 2. [Текст]
4. Режим доступа: [https://www.consultant.ru/document/ cons\_doc\_LAW\_7370/](https://www.consultant.ru/document/%20cons_doc_LAW_7370/)(дата обращения 22.03.2023)
5. Избачков Ю., Петров В. Информационные системы; СИНТЕГ - Москва, 2020. - 656 c.
6. Илюшечкин, В. М. Основы использования и проектирования баз данных. Учебник / В.М. Илюшечкин. - М.: Юрайт, 2015. [Текст] - 214 c.   
   Режим доступа: https://urss.ru/images/add\_ru/187807-1.pdf (дата обращения 22.03.2023)

# **ПРИЛОЖЕНИЕ А**

(справочное)

Листинги информационной системы

Листинг 1.1 - Создание базы данных «STCO»

CREATE DATABASE STCO

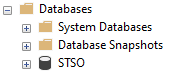


Рисунок 1.1 - Создание БД

Листинг 1.2 - Создание таблиц «Записи», «Сотрудники», «Услуги», «Подопечные»

CREATE TABLE employee

(

e\_id int NOT NULL IDENTITY (1,1),

e\_fio nvarchar(150) NOT NULL,

e\_post nvarchar(150) NOT NULL,

e\_date\_of\_birth date NOT NULL,

e\_address nvarchar(200) NOT NULL,

e\_phone char(15) NOT NULL

)

GO

CREATE TABLE records

(

r\_id int NOT NULL IDENTITY (1,1),

r\_employee int NOT NULL,

r\_wards int NOT NULL,

r\_service int NOT NULL,

r\_date date NOT NULL,

r\_status nvarchar(150) NOT NULL

)

GO

CREATE TABLE wards

(

w\_id int NOT NULL IDENTITY (1,1),

w\_fio nvarchar(150) NOT NULL,

w\_date\_of\_birth date NOT NULL,

w\_address nvarchar(200) NOT NULL,

w\_phone char(15) NOT NULL

)

GO

CREATE TABLE services\_

(

s\_service int NOT NULL IDENTITY (1,1),

s\_name nvarchar(150) NOT NULL,

s\_price int NOT NULL

)

GO

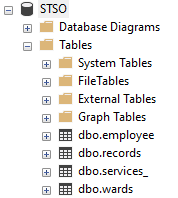


Рисунок 1.2 - Создание таблиц

Листинг 1.3 - Создание первичных ключей для таблиц «Записи», «Сотрудники», «Услуги», «Подопечные»

ALTER TABLE employee

ADD CONSTRAINT PK\_employee

PRIMARY KEY (e\_id ASC)

ALTER TABLE records

ADD CONSTRAINT PK\_records

PRIMARY KEY (r\_id ASC)

ALTER TABLE services\_

ADD CONSTRAINT PK\_services\_

PRIMARY KEY (s\_id ASC)

ALTER TABLE wards

ADD CONSTRAINT PK\_wards

PRIMARY KEY (w\_id ASC)

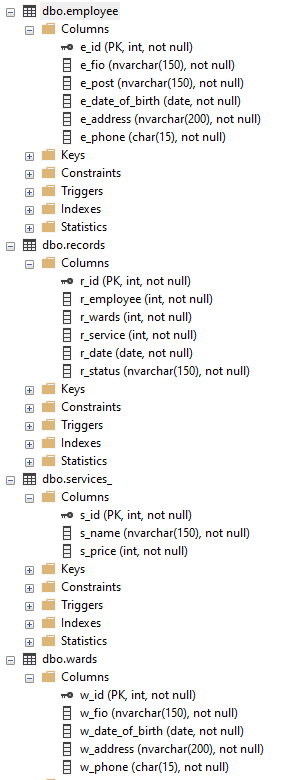


Рисунок 1.3 - Создание первичных ключей

Листинг 1.4 – Создание связей между таблицами «Записи»-«Сотрудники», «Записи»-«Услуги», «Записи»-«Подопечные»

ALTER TABLE records

ADD CONSTRAINT REF\_records\_services\_

FOREIGN KEY (r\_service) REFERENCES services\_ (s\_id)

ON DELETE Cascade ON UPDATE Cascade

ALTER TABLE records

ADD CONSTRAINT REF\_records\_employee

FOREIGN KEY (r\_employee) REFERENCES employee (e\_id)

ON DELETE Cascade ON UPDATE Cascade

ALTER TABLE records

ADD CONSTRAINT REF\_records\_wards

FOREIGN KEY (r\_wards) REFERENCES wards (w\_id)

ON DELETE Cascade ON UPDATE Cascade

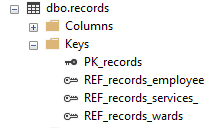


Рисунок 1.4 – Создание связей между таблицами

Листинг 1.5 – Создание хранимой процедуры вывода записей

CREATE PROCEDURE records\_db as

SELECT r\_id,e\_fio,w\_fio,s\_name

FROM records

INNER JOIN employee

ON r\_employee = e\_id

INNER JOIN wards

ON r\_wards = w\_id

INNER JOIN services\_

ON r\_service = s\_id

ORDER BY r\_id

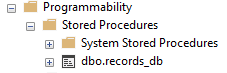


Рисунок 1.5 – Создание хранимой процедуры

Листинг 1.6 – Главная форма «STSO»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace STSO

{

enum RowState

{

Exsited,

New,

Modified,

ModifiedNew,

Deleted

}

public partial class Main : Form

{

DataBaseConnection database = new DataBaseConnection();

int SelectedRow;

public Main()

{

InitializeComponent();

}

private void CreateColumnsRecords()

{

DGW1.Columns.Add("r\_id", "id");

DGW1.Columns.Add("e\_fio", "Сотрудник");

DGW1.Columns.Add("w\_fio", "Подопечный");

DGW1.Columns.Add("s\_name", "Услуга");

DGW1.Columns.Add("s\_price", "Цена");

DGW1.Columns.Add("r\_date", "Дата");

DGW1.Columns.Add("r\_status", "Статус");

}

private void ReadSinglRowRecords(DataGridView dgw, IDataRecord record)

{

dgw.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetString(2), record.GetString(3), record.GetInt32(4), record.GetDateTime(5), record.GetString(6),RowState.ModifiedNew);

}

private void RefreshDataGridRecords(DataGridView dgw)

{

dgw.Rows.Clear();

string queryString = $"exec records\_db";

SqlCommand command = new SqlCommand(queryString,database.GetConnection());

database.openConnection();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

ReadSinglRowRecords(dgw,reader);

}

reader.Close();

}

private void CreateColumnsEmployee()

{

DGW2.Columns.Add("e\_id", "id");

DGW2.Columns.Add("e\_fio", "ФИО");

DGW2.Columns.Add("e\_post", "Должность");

DGW2.Columns.Add("e\_date\_of\_birth", "Дата рождения");

DGW2.Columns.Add("e\_address", "Адрес проживания");

DGW2.Columns.Add("e\_phone", "Номер телефона");

}

private void ReadSinglRowEmployee(DataGridView dgw2, IDataRecord record)

{

dgw2.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetString(2), record.GetDateTime(3), record.GetString(4), record.GetString(5), RowState.ModifiedNew);

}

private void RefreshDataGridEmployee(DataGridView dgw2)

{

dgw2.Rows.Clear();

string queryString = $"select\*from employee";

SqlCommand command = new SqlCommand(queryString, database.GetConnection());

database.openConnection();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

ReadSinglRowEmployee(dgw2, reader);

}

reader.Close();

}

private void CreateColumnsWards()

{

DGW3.Columns.Add("w\_id", "id");

DGW3.Columns.Add("w\_fio", "ФИО");

DGW3.Columns.Add("w\_date\_of\_birth", "Дата рождения");

DGW3.Columns.Add("e\_address", "Адрес проживания");

DGW3.Columns.Add("e\_phone", "Номер телефона");

}

private void ReadSinglRowWards(DataGridView dgw3, IDataRecord record)

{

dgw3.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetDateTime(2), record.GetString(3), record.GetString(4), RowState.ModifiedNew);

}

private void RefreshDataGridWards(DataGridView dgw3)

{

dgw3.Rows.Clear();

string queryString = $"select\*from wards";

SqlCommand command = new SqlCommand(queryString, database.GetConnection());

database.openConnection();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

ReadSinglRowWards(dgw3, reader);

}

reader.Close();

}

private void CreateColumnsServices()

{

DGW4.Columns.Add("s\_id", "id");

DGW4.Columns.Add("s\_name", "Наименование");

DGW4.Columns.Add("s\_price", "Цена");

}

private void ReadSinglRowServices(DataGridView dgw4, IDataRecord record)

{

dgw4.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetInt32(2), RowState.ModifiedNew);

}

private void RefreshDataGridServices(DataGridView dgw4)

{

dgw4.Rows.Clear();

string queryString = $"select\*from services\_";

SqlCommand command = new SqlCommand(queryString, database.GetConnection());

database.openConnection();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

ReadSinglRowServices(dgw4, reader);

}

reader.Close();

}

private void Form1\_Load(object sender, EventArgs e)

{

CreateColumnsRecords();

RefreshDataGridRecords(DGW1);

CreateColumnsEmployee();

RefreshDataGridEmployee(DGW2);

CreateColumnsWards();

RefreshDataGridWards(DGW3);

CreateColumnsServices();

RefreshDataGridServices(DGW4);

}

private void DGW1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

SelectedRow = e.RowIndex;

if(e.RowIndex >= 0)

{

DataGridViewRow row = DGW1.Rows[SelectedRow];

textBoxID.Text = row.Cells[0].Value.ToString();

textBoxEmployee.Text = row.Cells[1].Value.ToString();

textBoxWards.Text = row.Cells[2].Value.ToString();

textBoxServices.Text = row.Cells[3].Value.ToString();

textBoxDateRecords.Text = row.Cells[4].Value.ToString();

textBoxStatus.Text = row.Cells[5].Value.ToString();

}

}

private void button2\_Click(object sender, EventArgs e)

{

RefreshDataGridRecords(DGW1);

}

private void button4\_Click(object sender, EventArgs e)

{

AddForm1 addform = new AddForm1();

addform.Show();

}

private void button12\_Click(object sender, EventArgs e)

{

RefreshDataGridRecords(DGW2);

}

private void button18\_Click(object sender, EventArgs e)

{

RefreshDataGridRecords(DGW3);

}

}

}

Листинг 1.6 – Класс соединения с сервером «DataBaseConnection»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.Sql;

using System.Data.SqlClient;

namespace STSO

{

class DataBaseConnection

{

SqlConnection sqlconnection = new SqlConnection(@"Data Source=DESKTOP-PJ5KOD6;Initial Catalog=STSO;Integrated Security=True;");

public void openConnection()

{

if (sqlconnection.State == System.Data.ConnectionState.Closed)

{

sqlconnection.Open();

}

}

public void closeConnection()

{

if (sqlconnection.State == System.Data.ConnectionState.Open)

{

sqlconnection.Close();

}

}

public SqlConnection GetConnection() {return sqlconnection;}

}

}