СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc152768586)

[1 ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ 5](#_Toc152768587)

[1.1 Основные требования по технике безопасности во время прохождения учебной практики 5](#_Toc152768588)

[1.2 Нормативно-правовые документы 6](#_Toc152768589)

[2 ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ 8](#_Toc152768590)

[2.1 Описание предметной области 8](#_Toc152768591)

[2.2 Группы пользователей информационной системы 8](#_Toc152768592)

[2.3 Основные требования, предъявляемые к информационной системе 9](#_Toc152768593)

[3 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЕТ ЛИЧНОГО СОСТАВА» 11](#_Toc152768594)

[3.1 Моделирование некоторых динамических аспектов системы 11](#_Toc152768595)

[3.1.1 Создание диаграммы вариантов использования 11](#_Toc152768596)

[3.1.2 Создание диаграммы деятельности 11](#_Toc152768597)

[3.2 Разработка функциональной структуры информационной системы. Карта навигации по информационной системе 12](#_Toc152768598)

[3.3 Разработка макета информационной системы «Учет личного состава» 12](#_Toc152768599)

[4 РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЕТ ЛИЧНОГО СОСТАВА» 13](#_Toc152768600)

[4.1 Моделирование и разработка базы данных 13](#_Toc152768601)

[4.2 Разработка программных модулей информационной системы «Учет личного состава» 15](#_Toc152768602)

[5 РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ 56](#_Toc152768603)

[5.1 Назначение эксперимента. Выбор и обоснование методики проведения тестирования 56](#_Toc152768604)

[5.2 Технология тестирования 56](#_Toc152768605)

[5.3 Результаты проведения тестирования 58](#_Toc152768606)

[ЗАКЛЮЧЕНИЕ 60](#_Toc152768607)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 61](#_Toc152768608)

[ПРИЛОЖЕНИЯ А 63](#_Toc152768609)

# ВВЕДЕНИЕ

С развитием технологий и автоматизации военных структур становится необходимым создание эффективного программного обеспечения для учета личного состава и обеспечения деятельности воинских частей. В рамках данного проекта предлагается разработать прикладное программное обеспечение для отдела учета личного состава батальона железнодорожных войск.

Отдел учета личного состава батальона, аналогичный отделу кадров в мирное время, имеет свои специфические задачи и особенности. Батальон железнодорожных войск, состоит из нескольких рот, в каждой из которых действует несколько взводов, а каждый взвод, в свою очередь, подразделяется на три отделения.

В настоящее время различаются два вида службы: срочная и служба по контракту. Каждый офицер части обладает удостоверением личности, заменяющим паспорт, а военнослужащие срочной службы имеют военный билет. Учет и обработка данных о видах службы также представляют собой важную часть функционала предлагаемого программного обеспечения.

# ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ

Разработка программных модулей для системы учет личного состава батальона железнодорожных войскрассчитан на удобство работы отдела кадров. Программный продукт представляет собой универсальное программное обеспечение, предназначенное для формирования и отображения информации о военнослужащих железнодорожных войск, для удобной эксплуатации в целях предоставления информации.

Задачи:

1. Автоматизация Учета личного состава батальона железнодорожных войск: Разработка модулей для автоматизации мониторинга и учета военнослужащих.
2. Управление командованием батальона: Реализация функционала по управление Командованием войск, добавлять и снимать командиров, изменять данные удостоверения.
3. Обеспечение Безопасности Данных: Разработка модулей для защиты конфиденциальности данных, связанных с данными военнослужащих, чтобы предотвратить несанкционированный доступ.

Цель и задачи в совокупности направлены для формирования и отображения информации о военнослужащих железнодорожных войск, для удобной эксплуатации в целях предоставления информации.

## Основные требования по технике безопасности во время прохождения учебной практики

К работе в компьютерном классе допускаются лица, ознакомленные с данной инструкцией по технике безопасности и правилам поведения. Необходимо неукоснительно соблюдать правила по технике безопасности, т.к. нарушение этих правил может привести к поражению электрическим током, вызвать возгорание и навредить вашему здоровью.

* Работа обучающегося в компьютерном классе разрешается только в присутствии преподавателя (лаборанта).
* Во время занятий посторонние лица могут находиться в классе только с разрешения преподавателя.
* Во время перемен между уроками проводится обязательное проветривание компьютерного кабинета с обязательным выходом обучающегося из класса.

Каждый обучающийся в ответе за состояние своего рабочего места и сохранность размещенного на нем оборудования.

Обучающегося, допустившего нарушение по технике безопасности и правилам поведения в компьютерном классе, может быть привлечен к дисциплинарной ответственности. Если нарушение техники безопасности связано с причинением имущественного ущерба, он несет и материальную ответственность в установленном законом порядке.

## Нормативно-правовые документы

Оформление и содержание технического задания должно соответствовать требованиям стандарта «ГОСТ 19.201-78. ЕСПД. Общие положения.

Техническое задание оформляют в соответствии с ГОСТ 19.106-78 на листах формата 11 и 12 по ГОСТ 2.301-68, как правило, без заполнения полей листа. Номера листов (страниц) проставляются в верхней части листа над текстом.

Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 19.104-78.

Информационную часть (аннотацию и содержание), лист регистрации изменений допускается в документ не включать.

Для внесения изменений или дополнений в техническое задание на последующих стадиях разработки программы или программного изделия выпускают дополнение к нему.

Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.

Техническое задание должно содержать следующие разделы:

* введение;
* основания для разработки;
* назначение разработки;
* требования к программе или программному изделию;
* требования к программной документации;
* технико-экономические показатели;
* стадии и этапы разработки;
* порядок контроля и приемки;
* в техническое задание допускается включать приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

Вывод по главе: были описаны, техника безопасности во время прохождения учебной практики; нормативно-правовые документы, которые отвечают за стандарты разработки; цели и задачи учебной практики, включая вводную часть, где описана тематика отрасли и её тенденции развития.

# ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ

Предметная область информационной системы — это материальная система или система, характеризующая элементы материального мира, информация о которой хранится и обрабатывается. Предметная область рассматривается как некоторая совокупность реальных объектов и связей между ними.

## Описание предметной области

Разработать прикладное программное обеспечение деятельности отдела учета личного состава батальона железнодорожных войск. Это фактически отдел кадров воинской части. Учет личного состава: Регистрация и хранение информации о каждом военнослужащем, включая личные данные, военное звание, вид службы и принадлежность к определенному подразделению.

Батальон расквартирован на отдельной территории. В батальоне несколько рот, в каждой роте несколько взводов, каждый взвод состоит из трех отделений. Поддержание такой техники в отличном состоянии также входит в обязанности личного состава батальона. Иерархия подразделений: Возможность отображения и управления структурой батальона, рот, взводов и отделений.

В настоящее время существует два вида службы: срочная и по контракту. Каждый офицер части имеет удостоверение личности, которое заменяет паспорт, а военнослужащий срочной службы – военный билет.

Безопасность данных: Гарантированная безопасность конфиденциальных данных в соответствии с военными стандартами и требованиями.

## Группы пользователей информационной системы

Сотрудники, в зависимости от своих функциональных обязанностей, могут выполнять действия, направленные на мониторинг, управление, анализ данных и обеспечение безопасности в рамках проектируемой информационной системы "Ремонт Пассажирских Вагонов".

Роли и их Обязанности:

* + Администратор: управление системами аутентификации, авторизации и шифрования; оптимизация работы системы для обеспечения высокой производительности; полный доступ ко всей системе.
  + Рекрутер: получение информации о военнослужащих и их документов.
  + Начальник: получение информации о командирах батальонов, взводов, роты и отделения.

## Основные требования, предъявляемые к информационной системе

Данная информационная система должна позволять автоматически вести учет батальона.

Кроме того, должна иметь возможность просматривать каждого военнослужащего, который служил в данном батальоне за заданный период времени.

Перечень требований пользователя к программному продукту:

* + Гибкость: система должна обладать высокой степенью редактирования, чтобы пользователи могли адаптировать её под различные потребности в рамках своих обязанностей; возможность легкого масштабирования системы при увеличении объема данных или числа пользователей.
  + Надежность: обеспечение бесперебойной работы системы с минимальным количеством сбоев и перерывов; регулярное создание резервных копий данных для предотвращения потери информации в случае сбоев.
  + Эффективность: обеспечение высокой производительности при обработке данных, особенно в условиях одновременного использования множества пользователей; эффективное использование вычислительных ресурсов, чтобы система функционировала быстро и эффективно.
  + Безопасность: реализация надежной системы аутентификации для проверки личности пользователей и установление строгих прав доступа; и использование шифрования для защиты передаваемой и хранимой чувствительной информации.

Вывод по главе: было описано, что такое предметная область и её описание; группы пользователей, которые будут пользоваться информационной системой и основные системные требования к ней.

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЕТ ЛИЧНОГО СОСТАВА»

Проектированием информационных систем называется многоступенчатый процесс их создания и/или модернизации путём применения упорядоченной совокупности методологий и инструментария. Проектирование (в отличие от моделирования) предполагает работу с пока несуществующим объектом и направлено на создание информационной системы.

## Моделирование некоторых динамических аспектов системы

Диаграммы деятельности применяются для моделирования динамических аспектов поведения системы. Эти динамические аспекты могут включать деятельность на любом уровне абстракции в любом представлении системной архитектуры, включая классы, интерфейсы, компоненты и узлы.

При разработке программного обеспечения заблаговременное планирование и моделирование значительно упрощают программирование. На этапе создания концептуальной модели для описания функционального назначения используется диаграмма вариантов использования, а для описания логики процедур и бизнес-процессов используется диаграмма деятельности.

## Создание диаграммы вариантов использования

В приложении А на рисунке 1 представлен пример диаграммы вариантов использования информационной системы «Учет личного состава».

## Создание диаграммы деятельности

В приложении А на рисунках 2 – 4 представлена диаграмма деятельности (по ролям) информационной системы «Учет личного состава».

## Разработка функциональной структуры информационной системы. Карта навигации по информационной системе

Карта переходов по информационной системе «Учет личного состава» представлена в приложении А на рисунке 5.

## Разработка макета информационной системы «Учет личного состава»

В приложении А на рисунке 6представлено окно авторизации.

На этапе авторизации пользователь может ввести логин и пароль, чтобы продолжить работу в информационной системе.

После того, как пользователь успешно прошел авторизацию, он может продолжить работу с таблицами на разных Формах. В приложении А на рисунках 6 - 14 представлены формы, которые будут отображены, в зависимости от прав пользователя в системе. Например, Рекрут.имеет доступ к форме «Солдаты» и «Удостоверения личности»

Дизайн выполнен в простом стиле для того, чтобы пользователь с любым уровнем владения компьютером мог без препятствий работать с программным обеспечением.

Вывод по главе: была спроектирована диаграмма вариантов использования, диаграмма деятельности и макеты будущей информационной системы.

# РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЕТ ЛИЧНОГО СОСТАВА»

Прототип — это один из этапов разработки, который заключается в продумывании содержания и расположения важных элементов интерфейса. Прототип — это моделирование конечного продукта. Это интерактивный макет, который может иметь любую степень точности. Основная цель создания прототипов – проверить, насколько последователен путь пользователя, и выявить препятствия, которые могут возникнуть в процессе его взаимодействия с продуктом.

## Моделирование и разработка базы данных

Исходя из анализа предметной области, можно выделить восемь сущностей: «Военнослужащие», «Удостоверение личности», «Командиры», «Батальон», «Рота», «Взвод», «Отделение», «Пользователи».

В приложении А на рисунке 15 представлена логическая модель данных базы данных «Учет личного состава».

Создание базы данных происходит в MS SQL, в примере представлено создание базы данных в MS SQL Server.

С помощью раздела «Создание» и конструктора таблиц MS SQL созданы все таблицы и поля, представленные на логической модели данных.

Названия полей и их типы данных представлены в таблицах 1-8.

Таблица 1– Поля таблицы «Auth»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Login | Числовой |
| Password | Текстовый |
| Rule | Текстовый |

Таблица 2 – Поля таблицы «Батальон»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код Батальона | Числовой |
| Название | Текстовый |
| Код Командира | Числовой |
| Код Зам Командира | Числовой |

Таблица 3 – Поля таблицы «Взвод»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код взвода | Числовой |
| Название | Текстовый |
| Код роты | Числовой |
| Код командира | Числовой |
| Код Зам Командира | Числовой |

Таблица 4– Поля таблицы «Военнослужащие»

| Название поля | Тип данных |
| --- | --- |
| Код военнослужащего | Числовой |
| Фамилия | Текстовый |
| Имя | Текстовый |
| Отчество | Текстовый |
| Год Рождения | Дата |
| Домашний телефон | Текстовый |
| Адрес проживания | Текстовый |
| Картинка | Текстовый |

Таблица 5 – Поля таблицы «Командиры

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код Командира | Числовой |
| Код Удостоверения | Числовой |

Таблица 6 – Поля таблицы «Отделения»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код Отделения | Числовой |
| Код Удостоверения | Числовой |
| Код Взвода | Числовой |

Таблица 7 – Поля таблицы «Рота»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код роты | Числовой |
| Код батальона | Числовой |
| Код командира | Числовой |
| Название роты | Текстовый |
| Код Зам командира | Числовой |

Таблица 8 – Поля таблицы «Удостоверение личности»

| Название поля | Тип данных |
| --- | --- |
| Код Удостоверения | Числовой |
| Код военнослужащего | Числовой |
| Должность | Текстовый |
| Звание военнослужащего | Текстовый |
| Вид Службы | Текстовый |
| Примечание | Текстовый |
| Начало службы | Дата |
| Конец службы | Дата |

В приложении А на рисунке 16 представлены все созданные таблицы в обозревателе объектов базы данных.

После того, как таблицы созданы, необходимо создать между ними связи.

В приложении А на рисунке 17 представлена физическая модель данных, отражающая все таблицы и их поля, а также связи между таблицами.

## Разработка программных модулей информационной системы «Учет личного состава»

Разработанные в MS Visual Studio формы продемонстрированы в приложении А на рисунках 17-26.

Свойства элементов формы «Auth» представлены в таблице 9.

Таблица 9 – Свойство элементов формы «Auth»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Auth | Text | «Auth» |
| StartPosition | CenterScreen |
| Size | 407;311 |

Событийно-управляемые процедуры в форме «Auth» представлены в таблице 10

Таблица 10 – Событийно-управляемые процедуры в форме «Auth»

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Auth | public partial class Auth : Form  { int j = 0;  Boolean flag = false;  public Auth()  {  InitializeComponent();  }  } |
| Событие buttonLogin\_Click | private void buttonLogin\_Click(object sender, EventArgs e)  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();    SqlDataAdapter da = new SqlDataAdapter("select \* from Auth", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  Boolean flag=false;  //Проверка созданных пользователей  for (int i = 0; i < dt.Rows.Count; i++)  if ((dt.Rows[i]["login"].ToString() == textBoxLog.Text) && (dt.Rows[i]["password"].ToString() == textBoxPass.Text)) { j = i;flag = true; }    //Разграничение прав доступа  if (flag == true)  switch (dt.Rows[j]["Rule"].ToString())  {  case "Администратор":  Admin admin = new Admin();  admin.Show();  this.Hide();  break;  case "Начальник":  Commander commander = new Commander();  commander.Show();  this.Hide();  break;  case "Рекрутер":  Soldier soldier = new Soldier();  soldier.Show();  this.Hide();  break;  }  else MessageBox.Show("ТЕБЕ ТУТ НЕ МЕСТО");    } |
| Событие buttonLogOut\_Click | private void buttonLogOut\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Событие buttonVisPass\_Click | private void buttonVisPass\_Click(object sender, EventArgs e)  {  //показать или сокрыть видимость пароля    if (flag == false)  {  textBoxPass.PasswordChar ='\0';  flag = true;  }  else if (flag==true)  {  textBoxPass.PasswordChar = '\*';  flag = false;  }  } |

Свойства элементов формы «Admin» представлены в таблице 11.

Таблица 11 – Свойство элементов формы «Admin»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Admin | Text | «Admin» |
| StartPosition | CenterScreen |
| Size | 356;467 |

Таблица 12 – Событийно-управляемые процедуры в форме «Admin»

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Admin | public partial class Admin : Form  {  int k = 0;  public Admin()  {  InitializeComponent();  }  } |
| Событие Admin\_Load | private void Admin\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности". При необходимости она может быть перемещена или удалена.  this.удостверение\_личностиTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Рота". При необходимости она может быть перемещена или удалена.  this.ротаTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Рота);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Отделение". При необходимости она может быть перемещена или удалена.  this.отделениеTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Отделение);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры". При необходимости она может быть перемещена или удалена.  this.командирыTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие". При необходимости она может быть перемещена или удалена.  this.военнослужащиеTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Взвод". При необходимости она может быть перемещена или удалена.  this.взводTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Взвод);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Батальон". При необходимости она может быть перемещена или удалена.  this.батальонTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Батальон);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Auth". При необходимости она может быть перемещена или удалена.  this.authTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Auth);  } |
| События buttonPrev\_Click | private void buttonPrev\_Click(object sender, EventArgs e)  {  //Переход на следущую строку в datagridview  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  dataGridView1.Rows[index].Selected = true;  if (index != 0)  {  dataGridView1.CurrentCell = dataGridView1[0, index - 1];  }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, i - 1];  }  //Вывод права доступа в комбобокс  comboBoxRule.Text = label4.Text;  } |
|  | private void buttonAdd\_Click(object sender, EventArgs e)  {  //Добавление нового пользователя через запрос SQl  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Auth([Password],[Rule]) VALUES('{textBoxPass.Text}','{comboBoxRule.Text}');", sqlConnect); ;  DataTable dt = new DataTable();  da.Fill(dt);  MessageBox.Show("Успешно добавлен");  //Перерисовка таблицы в datagridview  authTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Auth);  } |
| Событие buttonDel\_Click | private void buttonDel\_Click(object sender, EventArgs e)  {  //Удаление пользователя через SQl запрос  DialogResult result = MessageBox.Show("Выберите один из вариантов", "Сообщение", MessageBoxButtons.OKCancel);  if (result == DialogResult.OK)  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"DELETE FROM Auth WHERE Login = {textBoxLog.Text}; ", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  }    //Перерисовка таблицы в datagridview  authTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Auth);  } |
| Событие buttonReturn\_Click | private void buttonReturn\_Click(object sender, EventArgs e)  {  //Возвражение к окну авторизации  this.Hide();  Auth auth = new Auth();  auth.Show();  authTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  authTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Auth);  } |
| Событие buttonUpdate\_Click | private void buttonUpdate\_Click(object sender, EventArgs e)  {  //Заполнение данными из комбобокса в Таблицу  label4.Text = comboBoxRule.Text;  //Обновление таблиц  authTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  военнослужащиеTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  командирыTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  взводTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  батальонTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  ротаTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  отделениеTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  удостверение\_личностиTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);    } |
| Событие buttonOpen\_Click | private void buttonOpen\_Click(object sender, EventArgs e)  {  //открытие формы и появление таблицы  if (k == 0)  {  dataGridView1.Visible = true;  panel1.Location = new Point(383, 11);  this.Width = 733;  buttonOpen.Text = "Закрыть";  k = 1;  }  else  {  dataGridView1.Visible = false;  panel1.Location = new Point(11, 11);  this.Width = 354;  buttonOpen.Text = "Открыть";  k = 0;  }  } |
| Событие buttonChange\_Click | private void buttonChange\_Click(object sender, EventArgs e)  {  //Переход на выбранную таблицу в комбобоксе  switch (comboBoxChange.Text)  {  case "Солдаты":  dataGridView1.DataSource = военнослужащиеBindingSource;  textBoxPass.Visible = false;  textBoxLog.Visible = false;  Login.Visible = false;  PasswordLab.Visible = false;  RuleLab.Visible = false;  label4.Visible = false;  comboBoxRule.Visible = false;  break;  case "Паспорт":  dataGridView1.DataSource = удостверениеличностиBindingSource;  textBoxPass.Visible = false;  textBoxLog.Visible = false;  Login.Visible = false;  PasswordLab.Visible = false;  RuleLab.Visible = false;  label4.Visible = false;  comboBoxRule.Visible = false;  break;  case "Командиры":  dataGridView1.DataSource = командирыBindingSource;  textBoxPass.Visible = false;  textBoxLog.Visible = false;  Login.Visible = false;  PasswordLab.Visible = false;  RuleLab.Visible = false;  label4.Visible = false;  comboBoxRule.Visible = false;  break;  case "Батальон":  dataGridView1.DataSource = батальонBindingSource;  textBoxPass.Visible = false;  textBoxLog.Visible = false;  Login.Visible = false;  PasswordLab.Visible = false;  RuleLab.Visible = false;  label4.Visible = false;  comboBoxRule.Visible = false;  break;  case "Рота":  dataGridView1.DataSource = ротаBindingSource;  textBoxPass.Visible = false;  textBoxLog.Visible = false;  Login.Visible = false;  PasswordLab.Visible = false;  RuleLab.Visible = false;  label4.Visible = false;  comboBoxRule.Visible = false;  break;  case "Взвод":  dataGridView1.DataSource = взводBindingSource;  textBoxPass.Visible = false;  textBoxLog.Visible = false;  Login.Visible = false;  PasswordLab.Visible = false;  RuleLab.Visible = false;  label4.Visible = false;  comboBoxRule.Visible = false;  break;  case "Отделение":  dataGridView1.DataSource = отделениеBindingSource;  textBoxPass.Visible = false;  textBoxLog.Visible = false;  Login.Visible = false;  PasswordLab.Visible = false;  RuleLab.Visible = false;  label4.Visible = false;  comboBoxRule.Visible = false;  break;  case "Авторизация":  textBoxPass.Visible = true;  textBoxLog.Visible = true;  Login.Visible = true;  PasswordLab.Visible = true;  comboBoxRule.Visible = true;  RuleLab.Visible = true;  label4.Visible = true;  dataGridView1.DataSource = authBindingSource;  break;  }  } |
| Событие buttonsearch\_Click | private void buttonsearch\_Click(object sender, EventArgs e)  {  //Поиск по Таблице    for (int j = 0; j < dataGridView1.RowCount; j++)  {  dataGridView1.Rows[j].Selected = false;  if (dataGridView1.Rows[j].Cells[0].Value != null)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(textBoxSearch.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  break;  }  }  }  } |

Свойства элементов формы «Battalion» представлены в таблице 13.

Таблица 13 – Свойства элементов формы «Battalion»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Battalion | Text | «Battalion» |
| StartPosition | CenterScreen |
| Size | 356;467 |

Событийно-управляемые процедуры в форме «Battalion» представлены в таблице 14

Таблица 14 – Событийно-управляемые процедуры в форме «Battalion»

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Battalion | public partial class Battalion : Form  {  int k = 0;  public Battalion()  {  InitializeComponent();  }  } |
| Событие Battalion\_Load | private void Battalion\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие". При необходимости она может быть перемещена или удалена.  this.военнослужащиеTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности". При необходимости она может быть перемещена или удалена.  this.удостверение\_личностиTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры". При необходимости она может быть перемещена или удалена.  this.командирыTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Батальон". При необходимости она может быть перемещена или удалена.  this.батальонTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Батальон);  } |
| Метод searchInDataGridView | public void searchInDataGridView(string k)  {  /\*Метод для поиска данных на datagridView\*/  for (int j = 0; j < dataGridView2.RowCount; j++)  {  if (dataGridView2.Rows[j].Cells[0].Value.ToString().Contains(k))  {  dataGridView2.Rows[j].Selected = true;  dataGridView2.CurrentCell = dataGridView2[0, j];  break;  }  }  for (int j = 0; j < dataGridView3.RowCount; j++)  {  if (dataGridView3.Rows[j].Cells[0].Value.ToString().Contains(label11.Text))  {  dataGridView3.Rows[j].Selected = true;  dataGridView3.CurrentCell = dataGridView3[0, j];  break;  }    }  for (int j = 0; j < dataGridView4.RowCount; j++)  {  if (dataGridView4.Rows[j].Cells[0].Value.ToString().Contains(label12.Text))  {  dataGridView4.Rows[j].Selected = true;  dataGridView4.CurrentCell = dataGridView4[0, j];  break;  }  }  } |
| Событие buttonCom\_Click | private void buttonCom\_Click(object sender, EventArgs e)  {  //Вызов метода для поиска данных  searchInDataGridView(textBoxCom.Text);    } |
|  | private void buttonCom2\_Click(object sender, EventArgs e)  {  //Вызов метода для поиска данных  searchInDataGridView(textBoxCom2.Text);  } |
| Событие buttonAdd\_Click | private void buttonAdd\_Click(object sender, EventArgs e)  {  //Добавление нового пользователя через запрос SQl  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Батальон(Название,Код\_командира,[Код\_ Зам\_Командира]) VALUES('{textBoxNameBat.Text}','{textBoxCom2.Text}','{textBoxCom.Text}');", sqlConnect); ;  DataTable dt = new DataTable();  da.Fill(dt);  //Перерисовка таблицы в datagridview  батальонTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Батальон);  MessageBox.Show("Успешно добавлен");  } |
| Событие buttonDel\_Click | private void buttonDel\_Click(object sender, EventArgs e)  {  //Удаление пользователя  DialogResult result = MessageBox.Show("Выберите один из вариантов", "Сообщение", MessageBoxButtons.OKCancel);  if (result == DialogResult.OK)  {  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(label7.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  dataGridView1.Rows.RemoveAt(j);  break;  }  }  }  } |
| Событие buttonPrev\_Click | private void buttonPrev\_Click(object sender, EventArgs e)  {  //переход на предыдущую строку в datagridView  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  dataGridView1.Rows[index].Selected = true;  if (index != 0)  {  dataGridView1.CurrentCell = dataGridView1[0, index - 1];  }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, i - 1];  }  } |
| Событие buttonNext\_Click | private void buttonNext\_Click(object sender, EventArgs e)  {  //переход на следущую строку в datagridView  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  if ((i - 1) <= index) { dataGridView1.CurrentCell = dataGridView1[0, 0]; }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, index + 1];  }  } |
| Событие buttonCompany\_Click | private void buttonCompany\_Click(object sender, EventArgs e)  {  //переход на форму роты  this.Hide();  Company company = new Company();  company.Show();  } |
| Событие buttonOpen\_Click | private void buttonOpen\_Click(object sender, EventArgs e)  {  //открытие формы и появление таблицы  if (k == 0)  {  dataGridView1.Visible = true;  panel1.Location = new Point(471, 12);  this.Width = 863;  buttonOpen.Text = "Закрыть";  k = 1;  }  else  {  dataGridView1.Visible = false;  panel1.Location = new Point(12, 12);  this.Width = 386;  k = 0;  buttonOpen.Text = "Открыть";  }  } |
| Событие buttonReturn\_Click | private void buttonReturn\_Click(object sender, EventArgs e)  {  //вернуться на форму командиры  this.Hide();  Commander commander = new Commander();  commander.Show();  } |
| Событие buttonsearch\_Click | private void buttonsearch\_Click(object sender, EventArgs e)  {  //Поиск по таблице  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(textBoxSearch.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  break;  }    }  } |
| Событие buttonUpdate\_Click | private void buttonUpdate\_Click(object sender, EventArgs e)  {  батальонTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  dataGridView1.CurrentCell = dataGridView1[0, 0];  } |

Свойства элементов формы «Commander» представлены в таблице 15

Таблица 15 – Свойства элементов формы «Commander»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Commander | Text | «Commander» |
| StartPosition | CenterScreen |
| Size | 383; 439 |

Событийно-управляемые процедуры в форме «Commander» представлены в таблице 16

Таблица 16 – Событийно-управляемые процедуры в форме «Commander»

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Commander | public partial class Commander : Form  {  int k=0;  public Commander()  {  InitializeComponent();  }  } |
| Событие Commander\_Load | private void Commander\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие". При необходимости она может быть перемещена или удалена.  this.военнослужащиеTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности". При необходимости она может быть перемещена или удалена.  this.удостверение\_личностиTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры". При необходимости она может быть перемещена или удалена.  this.командирыTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры);  dataGridView1.AllowUserToAddRows = false;  pictureBoxPhoto.Image = Image.FromFile(label9.Text);  } |
| Метод searchInDataGridView | public void searchInDataGridView()  {  /\*Метод для поиска данных на datagridView\*/  for (int j = 0; j < dataGridView2.RowCount; j++)  {  if (dataGridView2.Rows[j].Cells[0].Value.ToString().Contains(textBoxPass.Text))  {  dataGridView2.Rows[j].Selected = true;  dataGridView2.CurrentCell = dataGridView2[0, j];  break;  }  }  for (int j = 0; j < dataGridView2.RowCount; j++)  {  if (dataGridView3.Rows[j].Cells[0].Value.ToString().Contains(label5.Text))  {  dataGridView3.Rows[j].Selected = true;  dataGridView3.CurrentCell = dataGridView3[0, j];  break;  }  }    //Появление фото из таблицы  pictureBoxPhoto.Image = Image.FromFile(label9.Text);  } |
| Событие buttonCheck\_Click | private void buttonCheck\_Click(object sender, EventArgs e)  {  //Вызов метода для поиска данных  searchInDataGridView();  //Обновление Данных в таблице  командирыTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  } |
| Событие buttonNext\_Click | private void buttonNext\_Click(object sender, EventArgs e)  {  //переход на следущую строку в datagridView  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  if ((i - 1) <= index) { dataGridView1.CurrentCell = dataGridView1[0, 0]; }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, index + 1];  }  //Вызов метода для поиска данных  searchInDataGridView();    } |
| Событие buttonPrev\_Click | private void buttonPrev\_Click(object sender, EventArgs e)  {  //переход на предыдущую строку в datagridView  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  dataGridView1.Rows[index].Selected = true;  if (index != 0)  {  dataGridView1.CurrentCell = dataGridView1[0, index - 1];  }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, i - 1];  }  //Вызов метода для поиска данных |
| Событие buttonDel\_Click | private void buttonDel\_Click(object sender, EventArgs e)  {  //Удаление пользователя  DialogResult result = MessageBox.Show("Выберите один из вариантов", "Сообщение", MessageBoxButtons.OKCancel);  if (result == DialogResult.OK)  {  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(label4.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  dataGridView1.Rows.RemoveAt(j);  break;  }  }  }  } |
| Событие buttonAdd\_Click | private void buttonAdd\_Click(object sender, EventArgs e)  {  //Добавление нового пользователя через запрос SQl  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Командиры(Код\_удовстверения) VALUES({textBoxPass.Text});", sqlConnect); ;  DataTable dt = new DataTable();  da.Fill(dt);  //Перерисовка таблицы в datagridview  командирыTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры);  //Уведомление пользователя о выполнении команды  MessageBox.Show("Успешно добавлен");  } |
| Событие buttonPass\_Click | private void buttonPass\_Click(object sender, EventArgs e)  {  //Открытие формы удостверение  Passport passport = new Passport();  passport.Show();  } |
| Событие buttonBat\_Click | private void buttonBat\_Click(object sender, EventArgs e)  {  //открытие формы Батальон  Battalion battalion = new Battalion();  battalion.Show();  this.Hide();  } |
| Событие buttonOpen\_Click | private void buttonOpen\_Click(object sender, EventArgs e)  {  //открытие формы и появление таблицы  if (k == 0)  {  dataGridView1.Visible = true;  panel1.Location = new Point(278, 13);  this.Width = 656;  buttonOpen.Text = "Закрыть";  k = 1;  }  else  {  dataGridView1.Visible = false;  panel1.Location = new Point(12, 12);  this.Width = 383;  buttonOpen.Text = "Открыть";  k = 0;  }  } |
| Событие buttonReturn\_Click | private void buttonReturn\_Click(object sender, EventArgs e)  {  //Вернуться на форму авторизации  Auth auth = new Auth();  auth.Show();  this.Hide();  } |
| Событие buttonIN\_Click | private void buttonIN\_Click(object sender, EventArgs e)  {  //Выбор и переход по другим формам  switch (comboBoxTable.Text)  {  case "Батальон":  Battalion battalion = new Battalion();  battalion.Show();  this.Hide();  break;  case "Рота":  Company company = new Company();  company.Show();  company.Com(1);  this.Hide();  break;  case "Взвод":  Platoon platoon = new Platoon();  platoon.Show();  platoon.Com(1);  this.Hide();  break;  case "Отделение":  Department department = new Department();  department.Show();  department.Com(1);  this.Hide();  break;  }  } |
| Событие buttonsearch\_Click | private void buttonsearch\_Click(object sender, EventArgs e)  {    for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(textBoxSearch.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  break;  }  }  } |
| Событие buttonUpdate\_Click | private void buttonUpdate\_Click(object sender, EventArgs e)  {  //Обновление таблицы  командирыTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  //Возвращение к 1 строке  dataGridView1.CurrentCell = dataGridView1[0, 0];  } |

Свойства элементов формы «Company» представлены в таблице 17.

Таблица 17 – Событийно-управляемые процедуры в форме «Company»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Company | Text | «Company» |
| StartPosition | CenterScreen |
| Size | 458; 571 |

Событийно-управляемые процедуры в форме «Company» представлены в таблице 18

Таблица 18 – Событийно-управляемые процедуры в форме «Company»

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Company | public partial class Company : Form  {  public Company()  {  InitializeComponent();  }  } |
| Событие Company\_Load | private void Company\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие". При необходимости она может быть перемещена или удалена.  this.военнослужащиеTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности". При необходимости она может быть перемещена или удалена.  this.удостверение\_личностиTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры". При необходимости она может быть перемещена или удалена.  this.командирыTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Рота". При необходимости она может быть перемещена или удалена.  this.ротаTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Рота);  } |
| Метод searchInDataGridView | public void searchInDataGridView(string k)  {  /\*Метод для поиска данных на datagridView\*/  for (int j = 0; j < dataGridView2.RowCount; j++)  {  if (dataGridView2.Rows[j].Cells[0].Value.ToString().Contains(k))  {  dataGridView2.Rows[j].Selected = true;  dataGridView2.CurrentCell = dataGridView2[0, j];  break;  }  }  for (int j = 0; j < dataGridView3.RowCount; j++)  {  if (dataGridView3.Rows[j].Cells[0].Value.ToString().Contains(labelPass2.Text))  {  dataGridView3.Rows[j].Selected = true;  dataGridView3.CurrentCell = dataGridView3[0, j];  break;  }  }  for (int j = 0; j < dataGridView4.RowCount; j++)  {  if (dataGridView4.Rows[j].Cells[0].Value.ToString().Contains(labelSoldier2.Text))  {  dataGridView4.Rows[j].Selected = true;  dataGridView4.CurrentCell = dataGridView4[0, j];  break;  }  }  } |
| Событие buttonCom\_Click | private void buttonCom\_Click(object sender, EventArgs e)  {  //Вызов метода для поиска данных  searchInDataGridView(textBoxCom.Text);  } |
| Событие buttonCom2\_Click | private void buttonCom2\_Click(object sender, EventArgs e)  {  //Вызов метода для поиска данных  searchInDataGridView(textBoxCom2.Text);  } |
| Событие buttonNext\_Click | private void buttonNext\_Click(object sender, EventArgs e)  {  //переход на следущую строку в datagridView  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  if ((i - 1) <= index) { dataGridView1.CurrentCell = dataGridView1[0, 0]; }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, index + 1];  }  } |
| Событие buttonPrev\_Click | private void buttonPrev\_Click(object sender, EventArgs e)  {  //переход на предыдущую строку в datagridView  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  dataGridView1.Rows[index].Selected = true;  if (index != 0)  {  dataGridView1.CurrentCell = dataGridView1[0, index - 1];  }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, i - 1];  }  } |
| Событие buttonAdd\_Click( | private void buttonAdd\_Click(object sender, EventArgs e)  {  //Добавление нового пользователя через запрос SQl  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Рота(Код\_командира,Название\_роты,Код\_Зам\_командира) VALUES('{textBoxCom.Text}','{textBoxName.Text}','{textBoxCom2.Text}');", sqlConnect); ;  DataTable dt = new DataTable();  da.Fill(dt);  //Перерисовка таблицы в datagridview  ротаTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Рота);  MessageBox.Show("Успешно добавлен");  } |
| Событие buttonDel\_Click | private void buttonDel\_Click(object sender, EventArgs e)  {  //Удаление пользователя  DialogResult result = MessageBox.Show("Выберите один из вариантов", "Сообщение", MessageBoxButtons.OKCancel);  if (result == DialogResult.OK)  {  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(label6.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  dataGridView1.Rows.RemoveAt(j);  break;  }  }  }  } |
| Событие buttonPlat\_Click | private void buttonPlat\_Click(object sender, EventArgs e)  {  //Переход на форму звода  this.Hide();  Platoon platoon = new Platoon();  platoon.Show();  } |
| Событие buttonOpen\_Click | private void buttonOpen\_Click(object sender, EventArgs e)  {  //открытие формы и появление таблицы  if (k == 0)  {  dataGridView1.Visible = true;  panel1.Location = new Point(573, 13);  this.Width = 1016;  buttonOpen.Text = "Закрыть";  k = 1;  }  else  {  dataGridView1.Visible = false;  panel1.Location = new Point(12, 12);  this.Width = 458;  buttonOpen.Text = "Открыть";  k = 0;  }  } |
| Метод Com | public void Com(int i)  {  l = i;  } |
| Событие buttonReturn\_Click | private void buttonReturn\_Click(object sender, EventArgs e)  {  //Вернуться на форму Батальона или Командира  if (l == 1)  {  this.Hide();  Commander commander = new Commander();  commander.Show();  }  else  {  this.Hide();  Battalion battalion = new Battalion();  battalion.Show();  }    } |
| Событие buttonsearch\_Click | private void buttonsearch\_Click(object sender, EventArgs e)  {  //Поиск по таблице  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(textBoxSearch.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  break;  }  }  } |
| Событие buttonUpdate\_Click | private void buttonUpdate\_Click(object sender, EventArgs e)  {  //Обновление данных  ротаTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  dataGridView1.CurrentCell = dataGridView1[0, 0];  } |

Свойства элементов формы «Department» представлены в таблице 19.

Таблица 19 – Свойства элементов формы «Department»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Department | Text | «Department» |
| StartPosition | CenterScreen |
| Size | 389; 489 |

Событийно-управляемые процедуры в форме «Department» представлены в таблице 20

Таблица 20 – Событийно-управляемые процедуры в форме «Department»

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Department | public partial class Department : Form  {  public Department()  {  InitializeComponent();  }  } |
| Событие Department\_Load | private void Department\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие". При необходимости она может быть перемещена или удалена.  this.военнослужащиеTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности". При необходимости она может быть перемещена или удалена.  this.удостверение\_личностиTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Отделение". При необходимости она может быть перемещена или удалена.  this.отделениеTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Отделение);  } |
| Событие buttonNext\_Click | private void buttonNext\_Click(object sender, EventArgs e)  {  //переход на следущую строку в datagridView  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  if ((i - 1) <= index) { dataGridView1.CurrentCell = dataGridView1[0, 0]; }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, index + 1];  }  } |
| Событие buttonPrev\_Click | private void buttonPrev\_Click(object sender, EventArgs e)  {  //переход на следущую строку в datagridView  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  dataGridView1.Rows[index].Selected = true;  if (index != 0)  {  dataGridView1.CurrentCell = dataGridView1[0, index - 1];  }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, i - 1];  }  } |
| Событие buttonAdd\_Click | private void buttonAdd\_Click(object sender, EventArgs e)  {  //Добавление нового пользователя через запрос SQl  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Отделение(Код\_удостверения,Код\_взвода) VALUES('{textBoxPass.Text}','{textBoxPlat.Text}');", sqlConnect); ;  DataTable dt = new DataTable();  da.Fill(dt);  //Перерисовка таблицы в datagridview  отделениеTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Отделение);  MessageBox.Show("Успешно добавлен");  } |
| Событие buttonDel\_Click | private void buttonDel\_Click(object sender, EventArgs e)  {  //Удаление пользователя  DialogResult result = MessageBox.Show("Выберите один из вариантов", "Сообщение", MessageBoxButtons.OKCancel);  if (result == DialogResult.OK)  {  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(label2.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  dataGridView1.Rows.RemoveAt(j);  break;  }  }  }  } |
| Событие searchInDataGridView | public void searchInDataGridView(string k)  {  /\*Метод для поиска данных на datagridView\*/  for (int j = 0; j < dataGridView2.RowCount; j++)  {  if (dataGridView2.Rows[j].Cells[0].Value.ToString().Contains(k))  {  dataGridView2.Rows[j].Selected = true;  dataGridView2.CurrentCell = dataGridView2[0, j];  break;  }  }  for (int j = 0; j < dataGridView3.RowCount; j++)  {  if (dataGridView3.Rows[j].Cells[0].Value.ToString().Contains(labelSoldier2.Text))  {  dataGridView3.Rows[j].Selected = true;  dataGridView3.CurrentCell = dataGridView3[0, j];  break;  }  }  } |
| Событие buttonCheck\_Click | private void buttonCheck\_Click(object sender, EventArgs e)  {  //Вызов метода для поиска данных  searchInDataGridView(textBoxPass.Text);  } |
| Событие buttonOpen\_Click | private void buttonOpen\_Click(object sender, EventArgs e)  {  //открытие формы и появление таблицы  if (k == 0)  {  dataGridView1.Visible = true;  panel1.Location = new Point(389, 13);  this.Width = 712;  buttonOpen.Text = "Закрыть";  k = 1;  }  else  {  dataGridView1.Visible = false;  panel1.Location = new Point(12, 12);  this.Width = 354;  buttonOpen.Text = "Закрыть";  k = 0;  }  } |
| Событие Com | public void Com(int i)  {  l = i;  } |
| Событие buttonReturn\_Click | private void buttonReturn\_Click(object sender, EventArgs e)  {  //Вернуться на форму Взвод или Командира  if (l==1)  {  Commander commander = new Commander();  commander.Show();  this.Hide();  }  else  {  this.Hide();  Platoon platoon = new Platoon();  platoon.Show();  }  } |
| Событие buttonsearch\_Click | private void buttonsearch\_Click(object sender, EventArgs e)  {  //Поиск по таблице  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(textBoxSearch.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  break;  }  }  } |
| Событие buttonUpdate\_Click | private void buttonUpdate\_Click(object sender, EventArgs e)  {  /\*Обновление таблицы данных\*/    отделениеTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  dataGridView1.CurrentCell = dataGridView1[0, 0];  } |
| Событие buttonPassport\_Click | private void buttonPassport\_Click(object sender, EventArgs e)  {  //Переход на форму паспорт  Passport passport = new Passport();  passport.Show();  } |

Свойства элементов формы «Passport» представлены в таблице 21.

Таблица 21 – Свойства элементов формы «Passport»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Passport | Text | «Passport» |
| StartPosition | CenterScreen |
| Size | 483; 716 |

Событийно-управляемые процедуры в форме «Passport» представлены в таблице 22

Таблица 22 – Событийно-управляемые процедуры в форме «Passport»

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Passport | public partial class Passport : Form  {  public Passport()  {  InitializeComponent();  }  } |
| Событие Passport\_Load | private void Passport\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие". При необходимости она может быть перемещена или удалена.  this.военнослужащиеTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности". При необходимости она может быть перемещена или удалена.  this.удостверение\_личностиTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности);  dataGridView1.AllowUserToAddRows = false;  } |
| Событие buttonUpdate\_Click | private void buttonUpdate\_Click(object sender, EventArgs e)  { /\*Обновление таблицы\*/  label13.Text=comboBoxTypServ.Text;  label14.Text=comboBoxTitle.Text;  удостверение\_личностиTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  dataGridView1.CurrentCell = dataGridView1[0, 0];  } |
| Метод searchInDataGridView | public void searchInDataGridView()  {  //Метод для поиска в datagridview  for (int j = 0; j < dataGridView2.RowCount; j++)  {  if (dataGridView2.Rows[j].Cells[0].Value.ToString().Contains(textBoxSoldier.Text))  {  dataGridView2.Rows[j].Selected = true;  dataGridView2.CurrentCell = dataGridView2[0, j];  break;  }  }  } |
| Событие buttonReturn\_Click | private void buttonReturn\_Click(object sender, EventArgs e)  {  //обновление таблицы  удостверение\_личностиTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  this.Hide();  } |
| Событие buttonNext\_Click | private void buttonNext\_Click(object sender, EventArgs e)  {  //Переход на следующую строку  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  if ((i - 1) <= index) { dataGridView1.CurrentCell = dataGridView1[0, 0]; }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, index + 1];  }  //Вызов метода  searchInDataGridView();  comboBoxTypServ.Text = label13.Text;  comboBoxTitle.Text = label14.Text;  } |
| Событие buttonPrev\_Click | private void buttonPrev\_Click(object sender, EventArgs e)  {  /\*переход на предыдущую строку\*/  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  dataGridView1.Rows[index].Selected = true;  if (index != 0)  {  dataGridView1.CurrentCell = dataGridView1[0, index - 1];  }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, i - 1];  }  //Вызов метода для поиска  searchInDataGridView();  //Присваивание значений для комбобокса  comboBoxTypServ.Text = label13.Text;  comboBoxTitle.Text = label14.Text;  } |
| Событие buttonDel\_Click | private void buttonDel\_Click(object sender, EventArgs e)  {  //Удаление пользователя  DialogResult result = MessageBox.Show("Выберите один из вариантов", "Сообщение", MessageBoxButtons.OKCancel);  if (result == DialogResult.OK)  {  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(labelPass2.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  dataGridView1.Rows.RemoveAt(j);  break;  }  }  }  } |
| Событие buttonAdd\_Click | private void buttonAdd\_Click(object sender, EventArgs e)  {  //Добавление пользователя через SQl запрос  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Удостверение\_личности(Должность,Звание\_военнослужащего,Вид\_службы) VALUES('{textBoxPost.Text} ','{comboBoxTitle.Text }','{ comboBoxTypServ.Text}');", sqlConnect); ;  DataTable dt = new DataTable();  da.Fill(dt);  //Перерисовка таблицы в datagridview  удостверение\_личностиTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности);  MessageBox.Show("Успешно добавлен");  } |
| Событие buttonOpen\_Click | private void buttonOpen\_Click(object sender, EventArgs e)  {  //открытие формы и появление таблицы  if (k == 0)  {  dataGridView1.Visible = true;  panel1.Location = new Point(910, 13);  this.Width = 1390;  buttonOpen.Text = "Закрыть";  k = 1;  }  else  {  dataGridView1.Visible = false;  panel1.Location = new Point(13, 13);  this.Width = 483;  buttonOpen.Text = "Открыть";  k = 0;  }  } |
| Событие buttonsearch\_Click | private void buttonsearch\_Click(object sender, EventArgs e)  { /\* Реализация поиска\*/  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(textBoxSearch.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  break;  }  }  } |

Свойства элементов формы «Platoon» представлены в таблице 23

Таблица 23 – Свойства элементов формы «Platoon»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Platoon | Text | «Platoon» |
| StartPosition | CenterScreen |
| Size | 370; 482 |

Событийно-управляемые процедуры в форме «Platoon» представлены в таблице 24

Таблица 24 – Событийно-управляемые процедуры в форме «Platoon»

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Platoon | public partial class Platoon : Form  {  public Platoon()  {  InitializeComponent();  }  } |
| Событие Platoon\_Load | private void Platoon\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие". При необходимости она может быть перемещена или удалена.  this.военнослужащиеTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности". При необходимости она может быть перемещена или удалена.  this.удостверение\_личностиTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Удостверение\_личности);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры". При необходимости она может быть перемещена или удалена.  this.командирыTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Командиры);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Взвод". При необходимости она может быть перемещена или удалена.  this.взводTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Взвод);      } |
| Метод searchInDataGridView | public void searchInDataGridView(string k)  {  //Метод для поиска в datagridview  for (int j = 0; j < dataGridView2.RowCount; j++)  {  if (dataGridView2.Rows[j].Cells[0].Value.ToString().Contains(k))  {  dataGridView2.Rows[j].Selected = true;  dataGridView2.CurrentCell = dataGridView2[0, j];  break;    }  }  for (int j = 0; j < dataGridView3.RowCount; j++)  {  if (dataGridView3.Rows[j].Cells[0].Value.ToString().Contains(labelPass2.Text))  {  dataGridView3.Rows[j].Selected = true;  dataGridView3.CurrentCell = dataGridView3[0, j];  break;    }  }  for (int j = 0; j < dataGridView4.RowCount; j++)  {  if (dataGridView4.Rows[j].Cells[0].Value.ToString().Contains(labelSold2.Text))  {  dataGridView4.Rows[j].Selected = true;  dataGridView4.CurrentCell = dataGridView4[0, j];  break;    }  }  } |
| Событие buttonCheck\_Click | private void buttonCheck\_Click(object sender, EventArgs e)  {  //Вызов метода для поиска  searchInDataGridView(textBoxCom.Text);  } |
| Событие buttonCheck2\_Click | private void buttonCheck2\_Click(object sender, EventArgs e)  {  //Вызов метода для поиска  searchInDataGridView(textBoxCom2.Text);  } |
| Событие buttonNext\_Click | private void buttonNext\_Click(object sender, EventArgs e)  {  //Переход на следующую строку  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  if ((i - 1) <= index) { dataGridView1.CurrentCell = dataGridView1[0, 0]; }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, index + 1];  }  } |
| Событие buttonPrev\_Click | private void buttonPrev\_Click(object sender, EventArgs e)  {  /\*переход на прошлую строку\*/  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  dataGridView1.Rows[index].Selected = true;  if (index != 0)  {  dataGridView1.CurrentCell = dataGridView1[0, index - 1];  }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, i - 1];  }  } |
| Событие buttonAdd\_Click | private void buttonAdd\_Click(object sender, EventArgs e)  {  //Добавление новой строки в базу данных  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Взвод(Код\_роты,Код\_командира,Название,Код\_Зам\_командира) VALUES('{textBoxComp.Text}','{textBoxCom.Text}','{textBoxName.Text}','{textBoxCom2.Text}');", sqlConnect); ;  DataTable dt = new DataTable();  da.Fill(dt);  взводTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Взвод);  } |
| Событие buttonDel\_Click | private void buttonDel\_Click(object sender, EventArgs e)  {  //Удаление пользователя  DialogResult result = MessageBox.Show("Выберите один из вариантов", "Сообщение", MessageBoxButtons.OKCancel);  if (result == DialogResult.OK)  {  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(labelPlat2.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  dataGridView1.Rows.RemoveAt(j);  break;  }  }  }  } |
| Событие buttonDepart\_Click | private void buttonDepart\_Click(object sender, EventArgs e)  {  // Переход на форму отделения  Department department = new Department();  this.Hide();  department.Show();  } |
| Событие buttonOpen\_Click | private void buttonOpen\_Click(object sender, EventArgs e)  { /\*Открыть или закрыть на форме таблицу\*/  if (k == 0)  {  dataGridView1.Visible = true;  panel1.Location = new Point(584, 2);  this.Width = 950;  buttonOpen.Text = "Закрыть";  k = 1;  }  else  {  dataGridView1.Visible = false;  panel1.Location = new Point(2, 2);  this.Width =370;  buttonOpen.Text = "Открыть";  k = 0;  }  } |
| Событие Com | public void Com(int i)  {  l = i;  } |
| Событие buttonReturn\_Click | private void buttonReturn\_Click(object sender, EventArgs e)  {  /\*переход на форму командира или Роты\*/  if (l == 1)  {  Commander commander = new Commander();  commander.Show();  this.Hide();  }  else  {  this.Hide();  Company company = new Company();  company.Show();  }  } |
| Событие buttonsearch\_Click | private void buttonsearch\_Click(object sender, EventArgs e)  {/\* поиск по таблице\*/  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(textBoxSearch.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  break;  }  }  } |
| Событие buttonUpdate\_Click | private void buttonUpdate\_Click(object sender, EventArgs e)  {  /\*Обновление таблицы\*/  взводTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  dataGridView1.CurrentCell = dataGridView1[0, 0];  } |

Свойства элементов формы «Soldier» представлены в таблице 25

Таблица 25 – Свойства элементов формы «Soldier»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Soldier | Text | «Soldier» |
| StartPosition | CenterScreen |
| Size | 360; 575 |

Событийно-управляемые процедуры в форме «Soldier» представлены в таблице 26

Таблица 26 – Событийно-управляемые процедуры в форме «Soldier»

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Soldier | public partial class Soldier : Form  {  public Soldier()  {  InitializeComponent();  }  } |
| Событие soldier\_Load | private void soldier\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие". При необходимости она может быть перемещена или удалена.  this.военнослужащиеTableAdapter.Fill(this.уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие);  dataGridView1.AllowUserToAddRows = false;  } |
| Событие buttonPass\_Click | private void buttonPass\_Click(object sender, EventArgs e)  { /\*открытие формы паспорт\*/  Passport passport = new Passport();  passport.Show();  военнослужащиеTableAdapter.Update(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet);  } |
| Событие buttonReturn\_Click | private void buttonReturn\_Click(object sender, EventArgs e)  {  /\*Переход на форму авторизации\*/  Auth auth = new Auth();  auth.Show();  this.Hide();  } |
| Событие buttonNext\_Click | private void buttonNext\_Click(object sender, EventArgs e)  {  /\*Переход на следующую строку \*/  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  if ((i - 1) <= index) { dataGridView1.CurrentCell = dataGridView1[0, 0]; }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, index + 1];  }  pictureBoxPhoto.Image = Image.FromFile(label9.Text);  } |
| Событие buttonPrev\_Click | private void buttonPrev\_Click(object sender, EventArgs e)  {  //переход на предыдущую строку  int i = dataGridView1.RowCount;  int index = dataGridView1.CurrentRow.Index;  dataGridView1.Rows[index].Selected = true;  if (index != 0)  {  dataGridView1.CurrentCell = dataGridView1[0, index - 1];  }  else  {  dataGridView1.Rows[index].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, i - 1];  }  pictureBoxPhoto.Image = Image.FromFile(label9.Text);  } |
| Событие buttonAdd\_Click | private void buttonAdd\_Click(object sender, EventArgs e)  {  //Добавление новой строки  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Военнослужащие(Фамилия,Имя,Отчество,Адрес\_Проживания,картинка) VALUES('{textBoxFam.Text} ','{textBoxName.Text}','{textBoxName2.Text}','{textBoxAddress.Text}','{label9.Text}');", sqlConnect); ;  DataTable dt = new DataTable();  da.Fill(dt);  военнослужащиеTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие);  //Уведомление о выполнении  MessageBox.Show("Успешно добавлен");  } |
| Событие buttonDel\_Click | private void buttonDel\_Click(object sender, EventArgs e)  {  /\*Удаление пользователя с возможность отмены и предупреждения\*/  DialogResult result= MessageBox.Show("Выберите один из вариантов","Сообщение",MessageBoxButtons.OKCancel);  if (result == DialogResult.OK)  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП\_ПМ01\_ИСПП\_5\_Степурин\_НС; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"DELETE FROM Военнослужащие WHERE Код\_военнослужащего = {labelSold2.Text}; ", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  }    военнослужащиеTableAdapter.Fill(уП\_ПМ01\_ИСПП\_5\_Степурин\_НСDataSet.Военнослужащие);  } |
| Событие buttonOpen\_Click | private void buttonOpen\_Click(object sender, EventArgs e)  {  /\*открытие и закрытие таблиц\*/  if (k == 0)  {  dataGridView1.Visible = true;  panel1.Location = new Point(890, 0);  this.Width = 1233;  buttonOpen.Text = "Закрыть";  buttonPrev.Location = new Point(569, 409);  buttonOpen.Location = new Point(684, 409);  buttonNext.Location = new Point(842, 409);  k = 1;  }  else  {  dataGridView1.Visible = false;  panel1.Location = new Point(15, 15);  buttonPrev.Location = new Point(10, 495);  buttonOpen.Location = new Point(123, 495);  buttonNext.Location = new Point(283, 495);  this.Width = 360;  buttonOpen.Text = "Открыть";  k = 0;  }  } |
| Событие pictureBoxAdd\_Click | private void pictureBoxAdd\_Click(object sender, EventArgs e)  {  /\*Добавление картинки к пользователю\*/  var fileContent = string.Empty;  var filePath = string.Empty;  using (OpenFileDialog openFileDialog = new OpenFileDialog())  {  openFileDialog.InitialDirectory = @"\\apetfs\ИСПП-5\Степурин\_НС\Практика 3 курс\Проект\Новая папка";  openFileDialog.Filter = "All files (\*.\*)|\*.\*";  openFileDialog.FilterIndex = 2;  openFileDialog.RestoreDirectory = true;  if (openFileDialog.ShowDialog() == DialogResult.OK)  {  filePath = openFileDialog.FileName;  var fileStream = openFileDialog.OpenFile();  using (StreamReader reader = new StreamReader(fileStream))  {  fileContent = reader.ReadToEnd();  }  pictureBoxPhoto.Image = Image.FromFile(filePath);  label9.Text = filePath;  }  }  } |
| Событие buttonsearch\_Click | private void buttonsearch\_Click(object sender, EventArgs e)  {  for (int j = 0; j < dataGridView1.RowCount; j++)  {  if (dataGridView1.Rows[j].Cells[0].Value.ToString().Contains(textBoxSearch.Text))  {  dataGridView1.Rows[j].Selected = true;  dataGridView1.CurrentCell = dataGridView1[0, j];  break;    }  }  } |

# РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ

Тестирование программного обеспечения – процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Тестовый сценарий – это неавтоматизированный или автоматизированный сценарий, содержащий инструкции по реализации тестового набора. Тестовый сценарий может быть написан вручную (для выполнения человеком) либо полностью или частично автоматизирован.

## Назначение эксперимента. Выбор и обоснование методики проведения тестирования

В данном разделе проведены два вида тестирования. Провести анализ тестирования и выбрать, как будет проходить тестирование ИС.

Модульное тестирование. Цель: проверить, что код работает именно так, как должен (при заданных входных параметрах выдает предполагаемый результат).

Анализ покрытия кода тестами. Цель: проверить, что весь наш код отрабатывает при модульном тестировании, что нет, не участвующих в тестировании участков кода.

Анализ стилистики кода. Цель: проверка кода на соблюдение стандартов разработки (отраслевых, корпоративных, принятых на проекте).

Анализ производительности. Цель: проанализировать производительность ключевых операций, потребление памяти, утечку памяти.

## Технология тестирования

Рассмотрим, какие технологии можно применить для модульного тестирования и покрытия кода тестами.

Как в ходе выполнения проекта, так и при развитии системы, изменения в ее алгоритмы вносятся в соответствии с техническими заданиями, хотя данный документ может иметь различные названия. В технических заданиях, как правило, описываются критерии приемки: как мы и заказчик убедимся в том, что заказанный функционал работает верно.

В качестве примера можно привести такой упрощенный критерий:

* Исходные данные: в систему внесен военнослужащий, которому, например, нужно изменить вид службы.
* Действия пользователя: пользователь ищет в таблице, какой это военнослужащий.
* В результате в результате отображены все данные о военнослужащем.
* Вручную такое тестирование, как правило, выполняется единожды, поскольку выполнять каждый раз полный комплект тестов очень трудоемко.
* Нет гарантии того, что тесты, описанные в техническом задании, покрывают функциональность полностью. Т.е. даже если тесты описаны и проводятся регулярно, вероятно будет существовать наличие не протестированного функционала.

Автоматизированное тестирование лишено данных недостатков. Конечно, возникает отдельная задача кодирования данного теста на основании критериев приемки, но это скорее дисциплинирующий фактор…

Прежде чем начинать юзабилити-тестирование необходимо собрать группу пользователей, которые будут тестировать данную информационную систему. Количество привлеченных пользователей должно быть не менее пяти человек. Пользователям будет представлен список некоторых вопросов, относящихся к графическому интерфейсу.

Таблица 27 **–** Информация о пользователях

|  |  |  |  |
| --- | --- | --- | --- |
| ФИО | Пол | Возраст | Образование |
| Неверов Дмитрий Сергеевич | Мужской | 18 | Общее образование |
| Макуров Владимир Алексеевич | Мужской | 18 | Общее образование |
| Швидко Ивана Анатолевич | Мужской | 18 | Общее образование |
| Буйлов Михаил Алексеевич | Мужской | 19 | Общее образование |
| Казанин Алексей Андреевич | Мужской | 18 | Общее образование |

## Результаты проведения тестирования

В ходе тестирования был применен метод нагрузочного тестирования и юзабилити-тестирования, которые проверяют работу при нагрузке системы и сделать простым и удобным в пользовании информационной системы.

Суть тестирования заключается в проверке работы всех обязательных функций.

Результаты проведения нагрузочного тестирования продемонстрированы в приложении А на рисунке 27

Исходя из результатов нагрузочного тестирования, можно прийти к выводу, что нагрузка, оказываемая на центральный процессор и память, не имеет критичных значений. Это значит, что показатель производительности информационной системы на приемлемом уровне, то есть сбои и длительная загрузка не должны помешать пользователю при работе с данным программным продуктом.

Таблица 28 **–** Результаты проведения юзабилити-тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вопрос | Неверов Дмитрий Сергеевич | Макуров Владимир Алексеевич | Швидко Ивана Анатолевич | Буйлов Михаил Алексеевич | Казанин Алексей Андреевич |
| Присутствуют ли такие дизайн-элементы, которые неприятны для зрения? | Да | Да | Нет | Да | Да |
| Присутствует ли отвлекающая информация? | Нет | Нет | Нет | Нет | Нет |
| Понятна ли структура приложения на интуитивном уровне? | Да | Да | Да | Да | Да |
| Смогли ли вы найти интересующую вас информацию без затруднений? | Да | Да | Да | Да | Да |
| Понятен ли смысл приложения? | Да | Да | Да | Да | Да |
| Общая оценка | 7/10 | 7/10 | 8/10 | 8/10 | 8/10 |

Исходя из средней оценки информационной системы, рассчитанной на основе общих оценок всех пользователей с учетом ответов на все поставленные вопросы, можно сделать вывод о том, что внешние характеристики графического интерфейса находятся на довольно высоком уровне. Также в информационной системе нет помех для пользователя в виде отвлекающей информации, так как все отображаемые данные находятся в рамках предметной области.

Вывод по главе: были проведены тесты путём проведения модульного тестирования, анализа покрытия кода тестами, анализа стилистики кода и анализом производительности. Также было проведено нагрузочное тестирование, в результате чего нагрузка на процессор и память не имела критичных значений.

# ЗАКЛЮЧЕНИЕ

В результате учебной практики был проведен анализ предметной области, спроектирована и разработана информационная система «Учет личного состава», серверная часть которой была реализована в MS SQL, а клиентская часть в MS Visual Studio.

В MS SQL была создана база данных для хранения информации о военнослужащих, их удостоверениях и какому отделению принадлежат.

Клиентская часть, реализованная в MS Visual Studio, отражает все хранящиеся в базе данных таблицы. Также добавлять новые и осуществлять поиск по таблицам.

Цели учебной практики были достигнуты путем проведения анализа предметной области, проектирования информационной системы и разработки программных модулей.

Это позволяет сделать вывод, что поставленные цели и задачи выполнены в полном объеме.

В дальнейшем данная информационная система может быть модернизирована путем улучшения и изменения дизайна, улучшения оптимизации кода.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 20886-85. Организация данных в системах обработки данных. Термины и определения [Электронный доступ]. — Режим доступа: [ГОСТ 20886-85. Организация данных в системах обработки данных. Термины и определения (rostest.info)](https://rostest.info/gost/001.001.040.035/gost-20886-85/#:~:text=%D0%93%D0%9E%D0%A1%D0%A2%2020886%2D85.%20%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B2,%D0%BD%D0%B0%D1%83%D1%87%D0%BD%D0%BE%2D%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%) (Дата обращения: 22.05.2013 г.).
2. ГОСТ Р. 59793–2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания [Электронный доступ]. — Режим доступа: [ГОСТ Р. 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания (internet-law.ru)](https://internet-law.ru/gosts/gost/77858/?ysclid=lifap277mg519765664) (Дата обращения: 30.04.2022 г.).
3. ГОСТ 19.201-78. ЕСПД. Общие положения. [Электронный доступ]. — Режим доступа: [ГОСТ 19.201-78.ЕСПД. Техническое задание. Требования к содержанию и оформлению (swrit.ru)](https://www.swrit.ru/doc/espd/19.201-78.pdf?ysclid=lpi5u6naom706376064) (Дата обращения: 01.10.1981 г.).
4. Microsoft Visual Studio [Электронный доступ]. — Режим доступа: [Руководство. Создание приложения — Microsoft (microsoft.com)](https://learn.microsoft.com/ru-ru/visualstudio/ide/walkthrough-building-an-application?view=vs-2022) (Дата обращения: 02.11.2023 г.).
5. Пауэрс Л., Снелл М. Microsoft Visual Studio 2008 = Microsoft Visual Studio 2008 Unleashed by Lars Powers and Mike Snell. — C: [«БХВ-Петербург»](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%A5%D0%92-%D0%9F%D0%B5%D1%82%D0%B5%D1%80%D0%B1%D1%83%D1%80%D0%B3_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2008. — С. 1200.
6. Основы проектирования баз данных: учебник для студ. Учреждений сред. проф. образования / Г.Н. Федорова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 224 с.
7. Федорова Г.Н. Разработка модуля программного обеспечения для компьютерных систем: учебник для студ. Учреждений сред. Проф. Образования / Г.Н.Федорова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 384 с.
8. Мартишин, С. А. Базы данных практическое применение СУБД SQL и NOSQL-типа для проектирования информационных систем: учеб. пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко.-М.: ИД "ФОРУМ": ИНФРА-М, 2
9. Электронно-библиотечная система «Лань» [Электронный доступ]. — Режим доступа: [https://e.lanbook.com/](https://e.lanbook.com/%20) (Дата обращения: 22.05.2013 г.)
10. Microsoft [Электронный доступ]. — Режим доступа: [Документация по интегрированной среде разработки Visual Studio](https://learn.microsoft.com/ru-ru/visualstudio/ide/?view=vs-2022) (Дата обращения: 30.05.2023 г.)
11. Microsoft [Электронный доступ]. — Режим доступа: [SQL Server Guides](https://learn.microsoft.com/en-us/sql/relational-databases/sql-server-guides?view=sql-server-ver16%20) (Дата обращения: 30.05.2023 г.)
12. Справочник по кадровой работе в военной организации [Электронный доступ]. — Режим доступа: [Справочник по кадровой работе в военной](http://www.voennoepravo.ru/files/kadri.pdf) (Дата обращения: 30.05.2009 г.)

# ПРИЛОЖЕНИЯ А

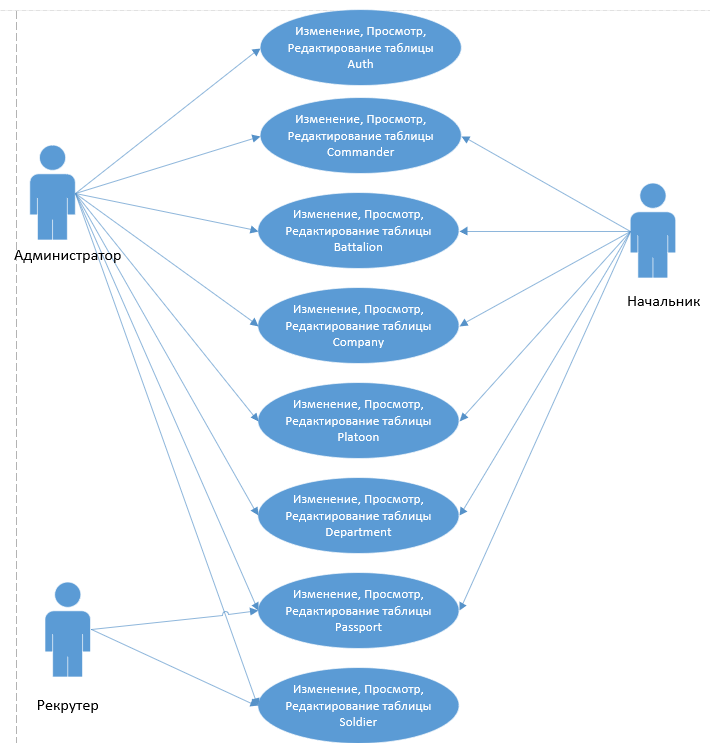


Рисунок 1– Диаграмма вариантов использования

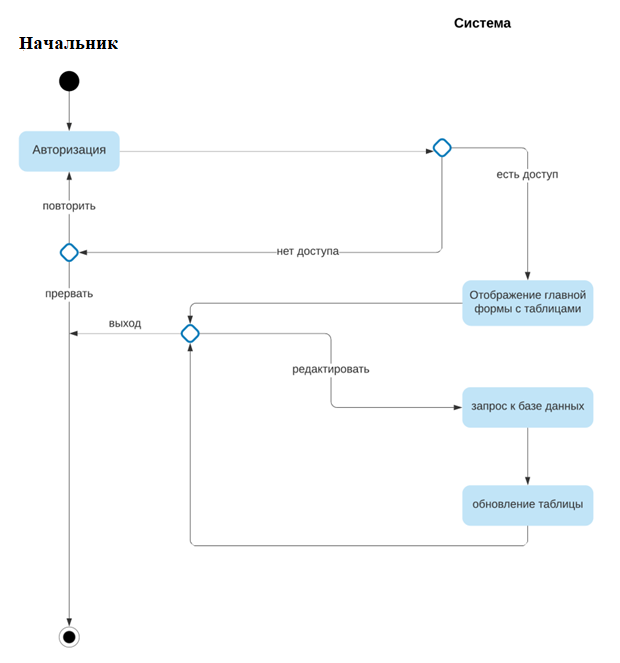


Рисунок 2 – Диаграмма деятельности (Начальник)

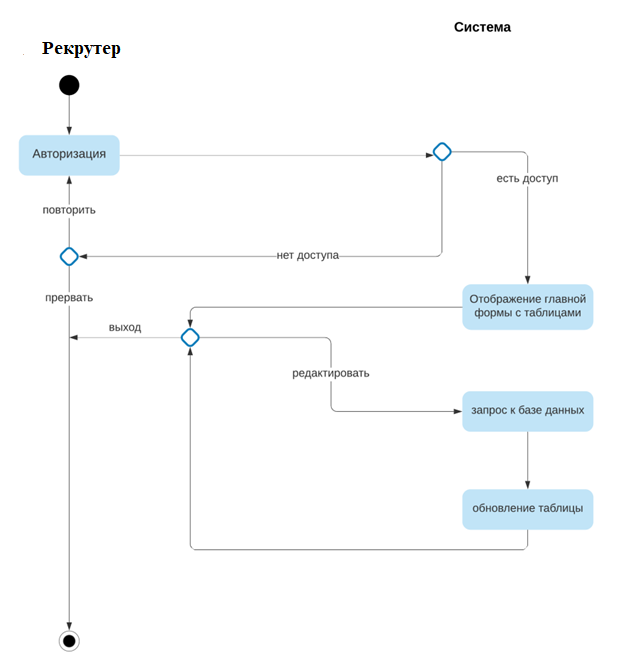


Рисунок 3 – Диаграмма деятельности (Рекрутер)



Рисунок 4 – Диаграмма деятельности (Администратор)

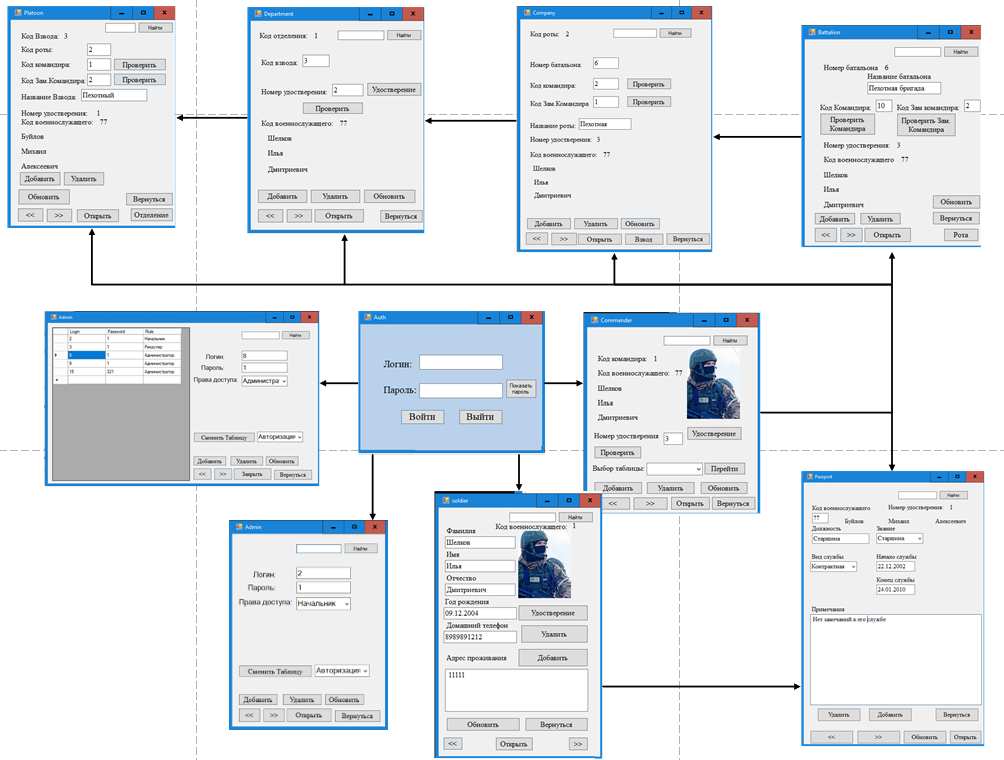


Рисунок 5 – Карта переходов

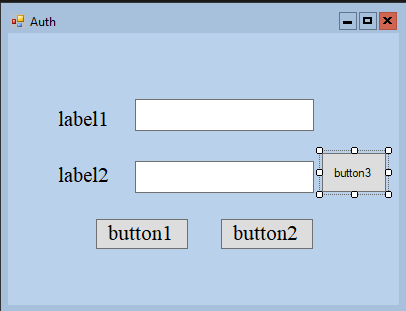


Рисунок 6 – Макет Авторизации

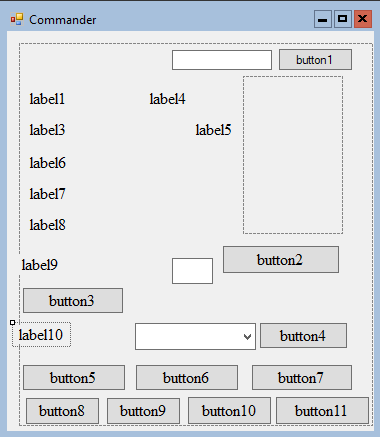


Рисунок 7 – Макет Командира

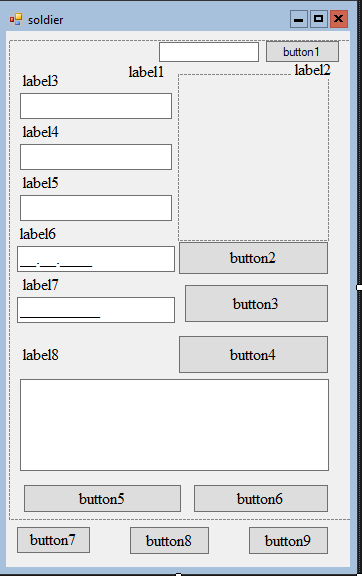


Рисунок 8 – Макет Солдата

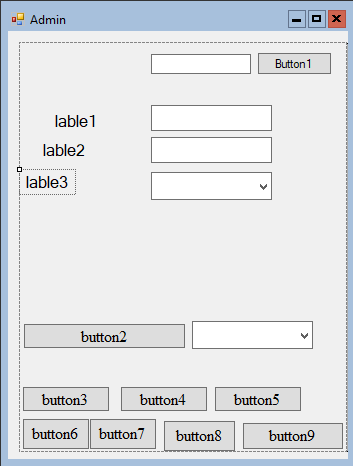


Рисунок 9 – Макет Администратора

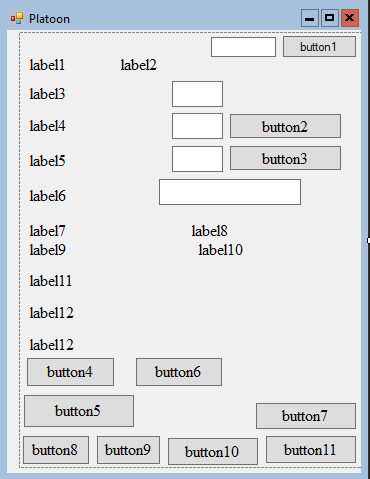


Рисунок 10 – Макет Взвода

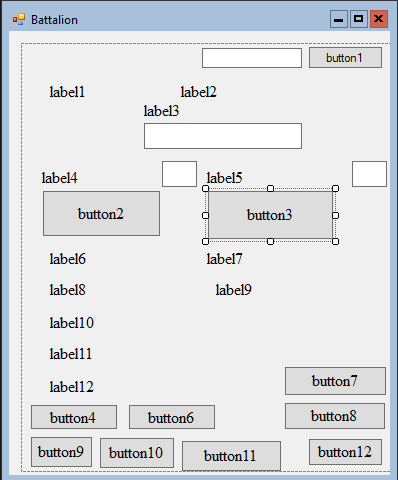


Рисунок 11 – Макет Батальона

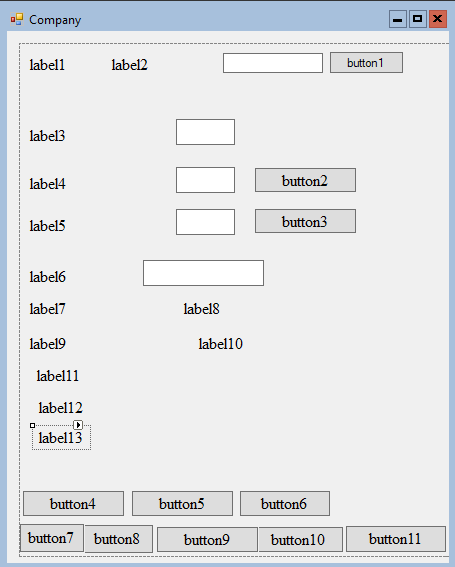


Рисунок 12 – Макет Роты

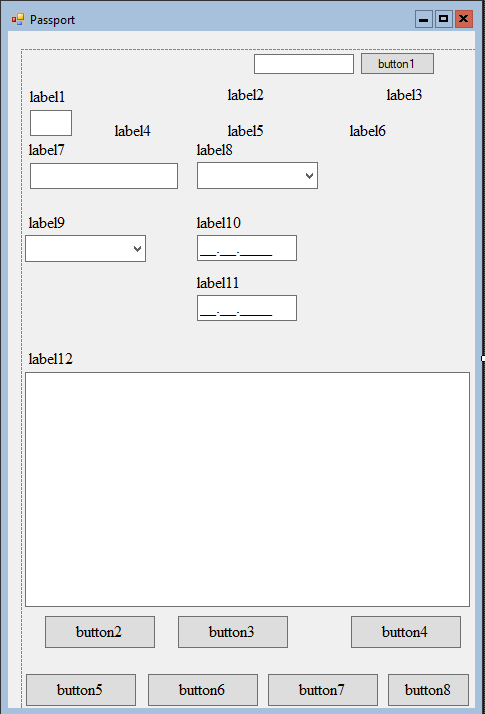


Рисунок 13 – Макет Удостоверения личности

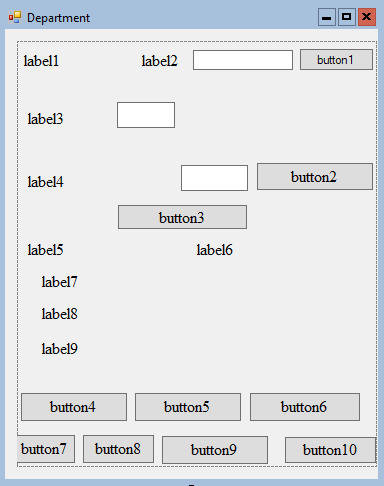


Рисунок 14 – Макет Отделения

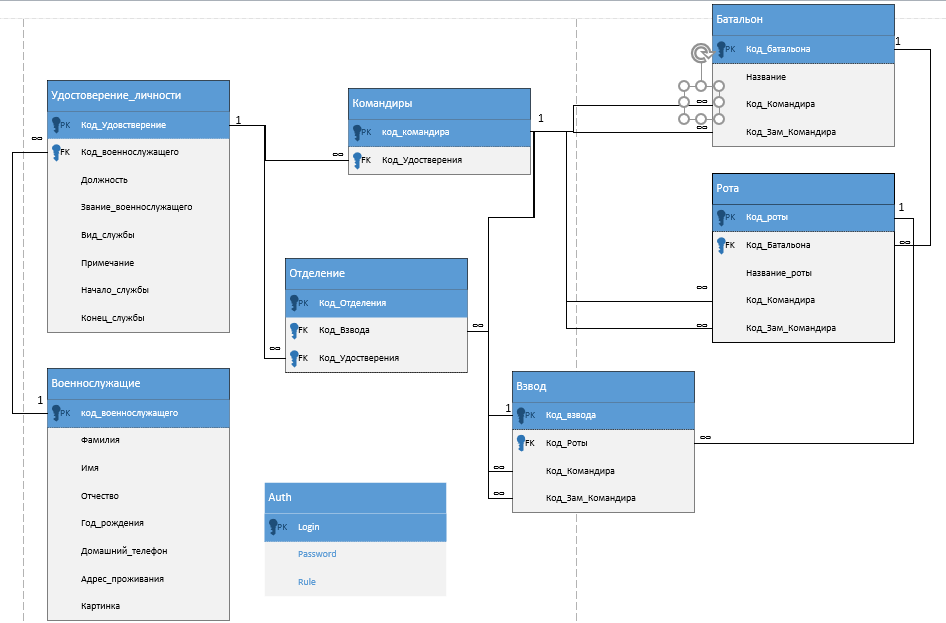


Рисунок 15– Логическая модель базы данных

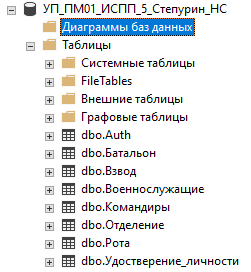


Рисунок 16 – Созданные таблицы

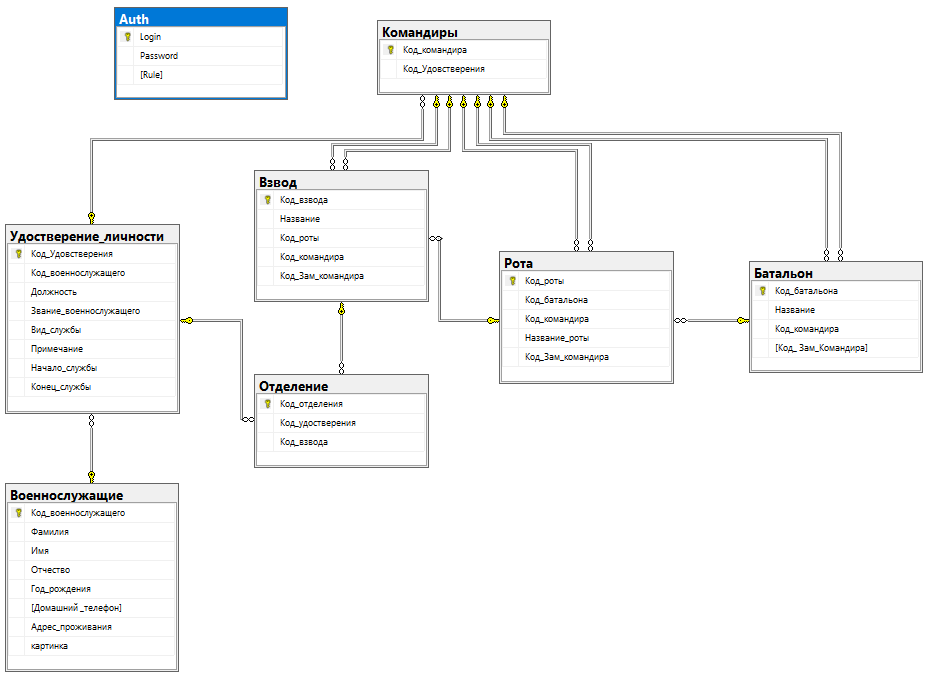


Рисунок 17 – Физическая модель базы данных

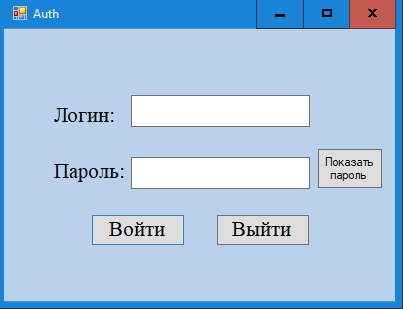


Рисунок 18 – Форма «Авторизация»

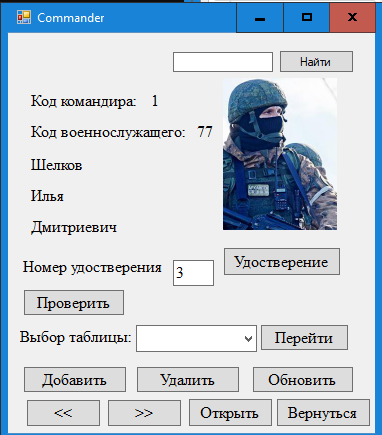


Рисунок 19– Форма «Командира»

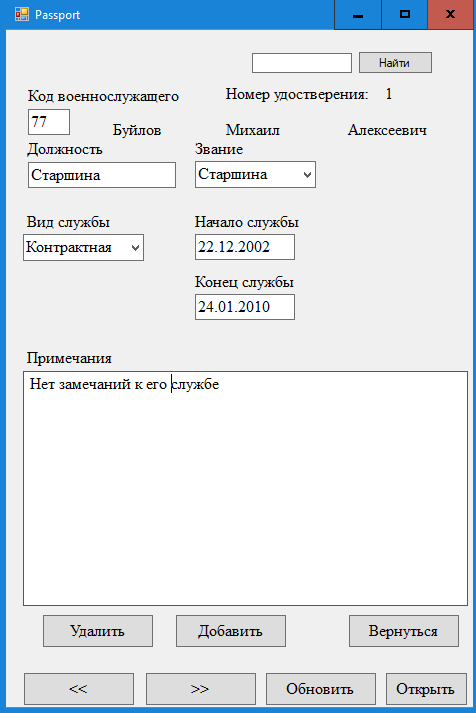


Рисунок 20 – Форма «Удостоверения»

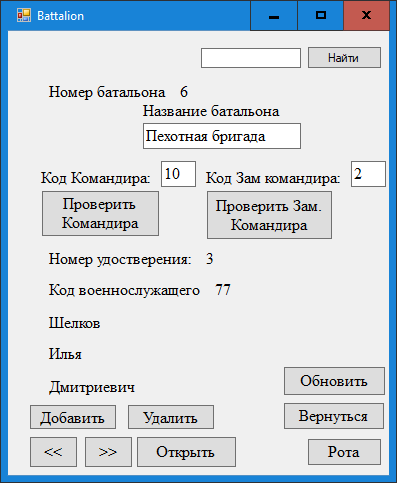


Рисунок 21 – Форма «Батальона»

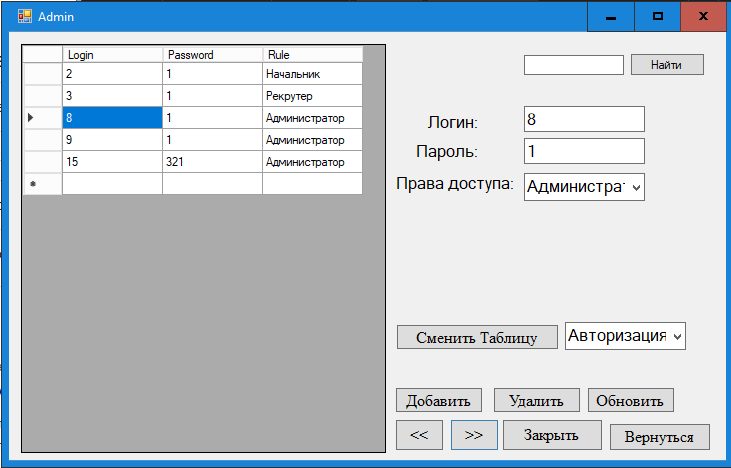


Рисунок 22 – Форма «Администратора»

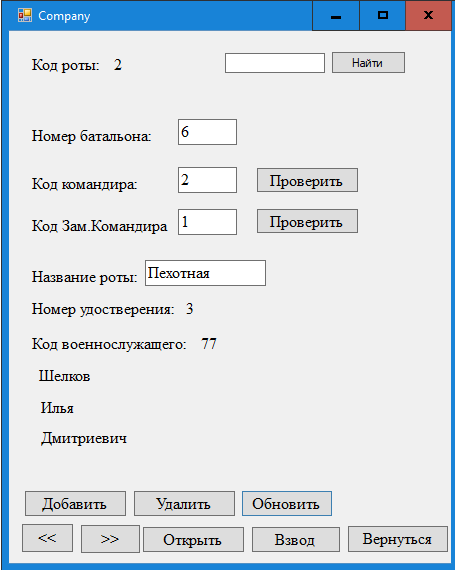


Рисунок 23 – Форма «Роты»

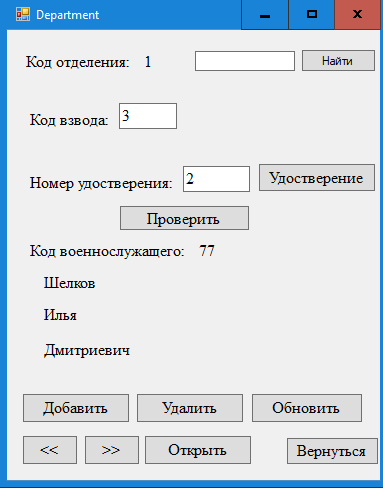


Рисунок 24– Форма «Отделения»

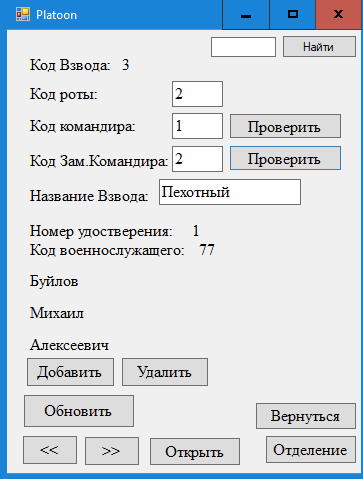


Рисунок 25 – Форма «Взвода»

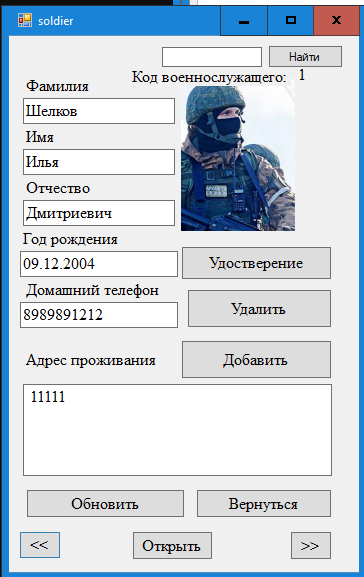


Рисунок 26 – Форма «Солдата»

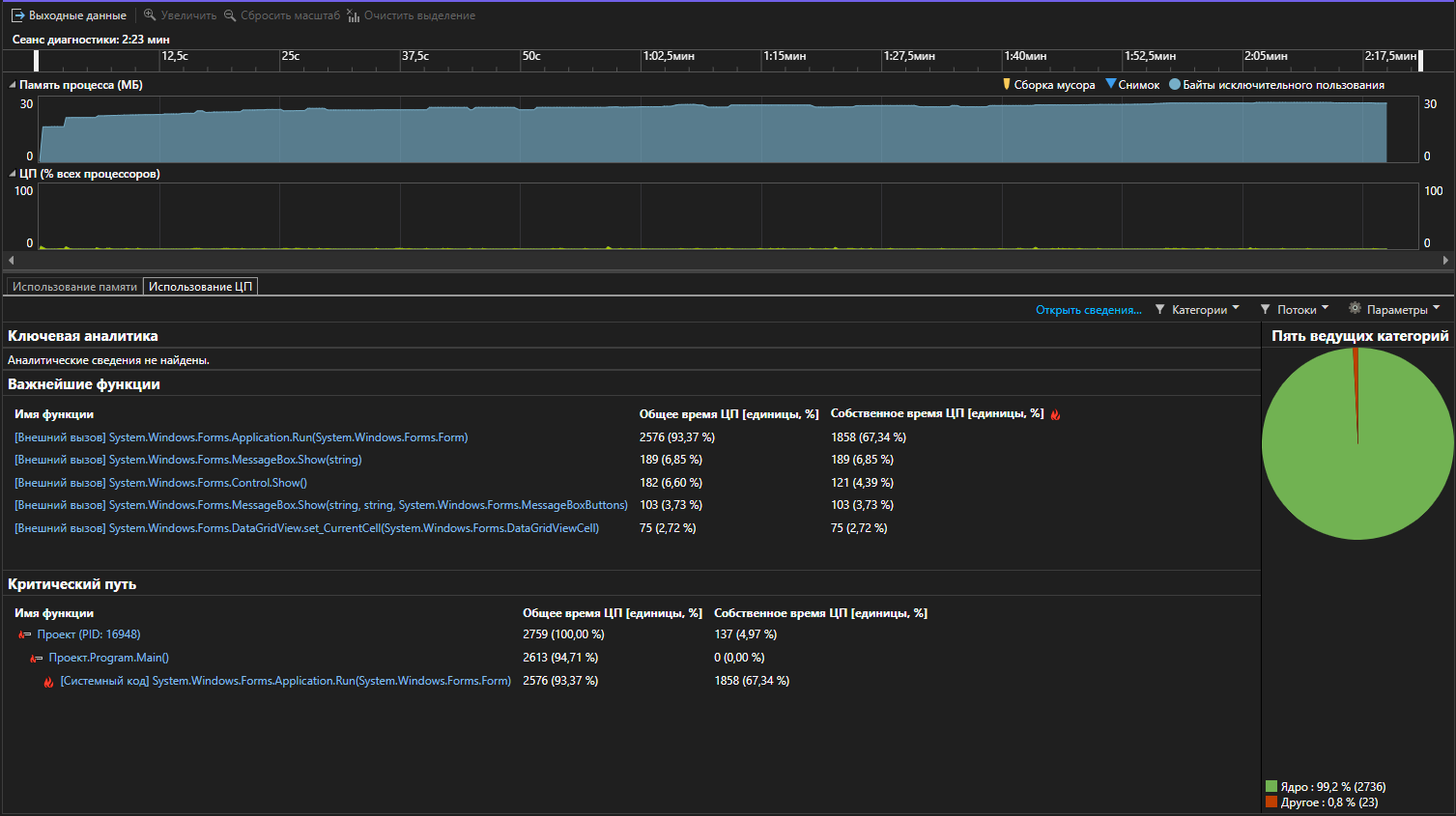


Рисунок 27 – Нагрузочное тестирование приложения