СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768575)

[1 ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ 6](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768576)

[1.1 Основные требования по технике безопасности во время прохождения учебной практики 6](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768577)

[1.2 Нормативно-правовые документы 7](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768578)

[2 ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ 9](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768579)

[2.1 Описание предметной области 9](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768580)

[2.2 Группы пользователей информационной системы 10](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768581)

[2.3 Основные требования, предъявляемые к информационной системе 11](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768582)

[3 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ (ДАЛЕЕ – ИС) «НАЗВАНИЕ» (Название взять из варианта) 12](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768583)

[3.1 Моделирование некоторых динамических аспектов системы 12](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768584)

[3.1.1 Создание диаграммы вариантов использования 12](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768585)

[3.1.2 Создание диаграммы деятельности 12](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768586)

[3.2 Разработка функциональной структуры информационной системы. Карта навигации по информационной системе 12](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768587)

[3.3 Разработка макета информационной системы «НАЗВАНИЕ» (Название взять из варианта) 12](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768588)

[4 РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «НАЗВАНИЕ» (Название взять из варианта) 12](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768589)

[4.1 Моделирование и разработка базы данных 12](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768590)

[4.2. Разработка программных модулей информационной системы «НАЗВАНИЕ» (Название взять из варианта) 12](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768591)

[5 РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ 13](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768592)

[5.1 Назначение эксперимента. Выбор и обоснование методики проведения тестирования 13](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768593)

[5.2 Технология тестирования 13](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768594)

[5.3 Результаты проведения тестирования 13](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768595)

[ЗАКЛЮЧЕНИЕ 14](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768596)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 15](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768597)

[ПРИЛОЖЕНИЕ A 16](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768598)

[16](file:///C:\Users\ADMIN\Downloads\Otchet.docx#_Toc152768599)

# ВВЕДЕНИЕ

Морская перевозка грузов является ключевой отраслью мировой экономики, обеспечивая мировую торговлю и перемещение товаров между различными странами и континентами. С учетом роста глобализации, спрос на морские перевозки постоянно растет, что подчеркивает важность разработки эффективной программной платформы для судоходных компаний, таких как "Балтика".

Прикладное программное обеспечение для судоходных компаний должно включать в себя ряд функций, чтобы обеспечить управление грузами, контейнерами и судами, а также предоставить инструменты для планирования маршрутов и отслеживания грузов. Программа может включать функции управления персоналом, бухгалтерию, а также веб-портал для клиентов.

Технологический прогресс в судостроении и логистике подталкивает разработчиков ПО к внедрению новых методов отслеживания грузов, управления флотом и оптимизации маршрутов для уменьшения затрат и снижения времени доставки. Также важно учитывать безопасность судов и грузов, устойчивость и соответствие международным правилам и стандартам.

Развитие отрасли морских перевозок связано с цифровой трансформацией и внедрением передовых программных решений.

# 1 ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ

Целью учебной практики является овладение видом профессиональной деятельности Разработка модулей программного обеспечения для компьютерных систем, общими и профессиональными компетенциями по специальности.

Основные задачи учебной практики:

- систематизация, обобщение, закрепление и углубление знаний и умений;

- формирование у студентов знаний, умений и навыков, профессиональных компетенций, профессионально значимых личностных качеств;

- развитие профессионального интереса, формирование мотивационно-целостного отношения к профессиональной деятельности, готовности к выполнению профессиональных задач в соответствии с нормами морали, профессиональной этики и служебного этикета.

Учебная практика профессионального модуля ПМ 01 Разработка модулей программного обеспечения для компьютерных систем проводится в кабинетах Информационно - вычислительного центра техникума преподавателями общепрофессиональных дисциплин профессионального цикла.

В период выполнения заданий учебной практики профессиональных модулей на студентов распространяются требования охраны труда и правила внутреннего распорядка, действующие в ГБПОУ ИО «АПЭТ»

## 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 обучающихся 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 обучающихся из класса.

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

Обучающийся, допустивший 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, обучающийся 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.

# 2 ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ

В рамках учебной практики было дано следующее задание: Разработать прикладное программное обеспечение деятельности судоходной компании «Балтика». Эта крупная компания занимается перевозками грузов между континентами. В ее собственности несколько десятков судов различного класса и грузоподъемности. К услугам этой компании обращаются тысячи клиентов из различных стран мира. На судне может находиться несколько партий грузов для различных грузополучателей из различных стран и городов. Одна партия груза может состоять из нескольких разновидностей грузов. У одной партии груза может быть только один отправитель и только один получатель. Судно следует по маршруту. Маршрут разрабатывается главным менеджером компании и проходит через несколько портов. В очередном порту назначения производится лишь частичная погрузка и выгрузка грузов, и судно следует дальше.

## 2.1 Описание предметной области

Проанализировав предметную область и данные предоставленные вариантом учебной практики (Приложение А – рис. 2), можно сделать вывод о необходимости 7 таблиц (сущностей) со следующими названиями: Груз, Капитан, Отправитель, Получатель, Пользователи, Порт получения, Судна.

Таблица (сущность) Груз включает в себя следующие поля (атрибуты): Таможенный номер партии груза, Регистрационный номер судна, ИНН отравителя груза, Единица измерения груза, Заявленная величина груза, Застрахованная величина груза, Наименование порта отправителя, Дата отправления.

Таблица (сущность) Капитан включает в себя следующие поля (атрибуты): Код капитана, Фамилия, Имя, Отчество, ИНН капитана.

Таблица (сущность) Отправитель включает в себя следующие поля (атрибуты): ИНН отправителя груза, Отправитель груза, Наименование банка, Адрес.

Таблица (сущность) Получатель включает в себя следующие поля (атрибуты): ИНН получателя, Наименование банка, Адрес.

Таблица (сущность) Пользователи включает в себя следующие поля (атрибуты): Код пользователя, Логин, Пароль, Права доступа, Имя, Фамилия, Отчество.

Таблица (сущность) Порт получения включает в себя следующие поля (атрибуты): Код получения, ИНН получателя, Таможенный номер партии груза, Наименование, Дата получения,

Таблица (сущность) Судна включает в себя следующие поля (атрибуты): Регистрационный номер судна, Код капитана, Название судна, Тип судна, Грузоподъемность судна.

По вышеперечисленной информации выполнена логическая модель данных (Приложение А – Рисунок 3 - Логическая модель данных).

## 2.2 Группы пользователей информационной системы

* случайный пользователь, взаимодействие которого с ИС не обусловлено служебными обязанностями;
* конечный пользователь (потребитель информации) - лицо или коллектив.
* коллектив специалистов (персонал ИС), включающий Администратора, Менеджера, Директора.

## 2.3 Основные требования, предъявляемые к информационной системе

* + 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 «БАЛТИКА»

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.

## 3.1 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.

3.1.1 Создание диаграммы вариантов использования

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

3.1.2 Создание диаграммы деятельности

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

## 3.3 Разработка макета информационной системы «БАЛТИКА»

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

В случае если пользователь еще не зарегистрирован в системе, он может сделать это, нажав на кнопку «Зарегистрироваться». В этом случае откроется окно регистрации.

На этапе регистрации пользователю необходимо придумать свой логин и пароль.

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.

# 4 РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «БАЛТИКА»

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.

## 4.1 Моделирование и разработка базы данных

Проанализировав предметную область и данные предоставленные вариантом учебной практики (Приложение А – рис. 2), можно сделать вывод о необходимости 7 таблиц (сущностей) со следующими названиями: Груз, Капитан, Отправитель, Получатель, Пользователи, Порт получения, Судна.

Таблица 1 – Поля таблицы «Грузы»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Таможенный номер партии груза | Числовой |
| Регистрационный номер груза | Числовой |
| ИНН отправителя груза | Числовой |
| Единица измерения груза | Текстовый |
| Заявленная величина груза | Числовой |
| Наименование порта отправителя | Текстовый |
| Дата отправления | Дата |

Таблица 2 – Поля таблицы «Капитан»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код капитана | Числовой |
| Имя | Текстовый |
| Фамилия | Текстовый |
| Отчество | Текстовый |
| ИНН капитана | Числовой |

Таблица 3 – Поля таблицы «Отправитель»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| ИНН отправителя груза | Числовой |
| Отправитель груза | Текстовый |
| Наименование банка | Текстовый |
| Адрес | Текстовый |

Таблица 4– Поля таблицы «Получатель»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| ИНН получателя | Числовой |
| Наименование банка | Текстовый |
| Адрес | Текстовый |

## Таблица 5 – Поля таблицы «Пользователи»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код пользователя | Числовой |
| Логин | Текстовый |
| Пароль | Текстовый |
| Права доступа | Текстовый |
| Имя | Текстовый |
| Фамилия | Текстовый |
| Отчество | Текстовый |

## Таблица 6 – Поля таблицы «Порт получения»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код получения | Числовой |
| ИНН получателя | Числовой |
| Таможенный номер партии груза | Числовой |
| Наименование | Текстовый |
| Дата получения | Текстовый |

## Таблица 7 – Поля таблицы «Судна»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Регистрационный номер судна | Числовой |
| Код капитана | Числовой |
| Название судна | Текстовый |
| Тип судна | Текстовый |
| Грузоподъемность судна | Числовой |

## 4.2. Разработка программных модулей информационной системы «БАЛТИКА»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Авторизация | Text | «Регистрация пользователя» |
| StartPosition | CenterScreen |
| Size | 407;311 |

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Авторизация | public partial class Avtorization : Form  {  public Avtorization()  {  InitializeComponent();  } |
| Событие textBox2\_KeyPress( | private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)  {  if (((e.KeyChar <= '9') && (e.KeyChar >= '0')) || (e.KeyChar == (char)Keys.Enter) || (e.KeyChar ==  (char)Keys.Back))  {  }  else e.KeyChar -= e.KeyChar;  }  //Проверка созданных пользователей  //Разграничение прав доступа |
| Событие textBox1\_KeyPress( | private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)  {  if ((e.KeyChar >= 'a') && (e.KeyChar <= 'z')||(e.KeyChar >= 'а') && (e.KeyChar <= 'я') || (e.KeyChar >= 'А') &&  (e.KeyChar <= 'Я')|| (e.KeyChar >= 'A') && (e.KeyChar <= 'Z')|| (e.KeyChar == (char)Keys.Enter) || (e.KeyChar ==  (char)Keys.Back))  { |

# 

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Пользователи | public partial class Пользователи : Form  {  public Пользователи()  {  InitializeComponent();  } |
| Событие ExitAvtorization\_Click | private void ExitAvtorization\_Click(object sender, EventArgs e)  {  Меню form = new Меню();  form.nameForm(nameform);  Hide();  form.Show();  } |
| События Exit\_Click | private void Exit\_Click(object sender, EventArgs e)  {  Application.Exit();  }  private void Пользователи\_Load(object sender, EventArgs e)  { |
| Событие Пользователи\_Load | private void Пользователи\_Load(object sender, EventArgs e)  } // TODO: данная строка кода позволяет загрузить данные в таблицу "шелков\_ИДDataSet.Пользователи".  При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.шелков\_ИДDataSet.Пользователи);  }  private void Update\_Click(object sender, EventArgs e)  {  try  {  this.пользователиTableAdapter.Update(this.шелков\_ИДDataSet.Пользователи);  }  catch  {  MessageBox.Show("Некоторые поля заполнены в неправильном формате", "Ошибка");  }  } |
| Событие button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = Шелков\_ИД; Integrated  Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"Delete from Пользователи where Код\_пользователя =  '{dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value.ToString()}' ", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  this.пользователиTableAdapter.Update(this.шелков\_ИДDataSet.Пользователи);  this.пользователиTableAdapter.Fill(this.шелков\_ИДDataSet.Пользователи);  }  }  } |

# Груз

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс Груз | public partial class Груз : Form  {  public Груз()  {  InitializeComponent();  } |
| Событие Exit\_Click | private void Exit\_Click(object sender, EventArgs e)  {  Application.Exit();  }  int nameform;  public void nameForm(int a)  {  nameform = a;  if (a == 3)  {  Update.Visible = false;  ExitAvtorization.Size = new Size(496, 23);  ExitAvtorization.Location = new Point(12, 168);  }  else  {  Update.Visible = true;  ExitAvtorization.Size = new Size(242, 23);  ExitAvtorization.Location = new Point(266, 168);  }  } |
| Метод ExitAvtorization\_Click | private void ExitAvtorization\_Click(object sender, EventArgs e)  {  Меню form = new Меню();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие Груз\_Load | private void Груз\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "шелков\_ИДDataSet.Груз". При  необходимости она может быть перемещена или удалена.  this.грузTableAdapter.Fill(this.шелков\_ИДDataSet.Груз);  } |
| Событие Update\_Click | private void Update\_Click(object sender, EventArgs e)  {  try  {  грузTableAdapter.Update(шелков\_ИДDataSet.Груз);  }  catch  {  MessageBox.Show("Некоторые поля заполнены в неправильном формате","Ошибка");  }  } |
| Событие dataGridView1\_CellClick | private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)  {  if (nameform == 3)  {  MessageBox.Show("Редактирование таблицы невозможно!");  }  } |
| Событие button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  try  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = Шелков\_ИД; Integrated  Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"Delete from Груз where [Таможенный номер партии груза] =  '{dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value.ToString()}' ", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  this.грузTableAdapter.Update(this.шелков\_ИДDataSet.Груз);  this.грузTableAdapter.Fill(this.шелков\_ИДDataSet.Груз);  }  catch  {  MessageBox.Show("Это поле соединено с другой таблицей", "Ошибка");  }  }  }  } |

Капитан

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс class Капитан | public partial class Капитан : Form  {  public Капитан()  {  InitializeComponent();  } |
| Событие Exit\_Click | private void Exit\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Метод Update\_Click | private void Update\_Click(object sender, EventArgs e)  {  try  {  капитанTableAdapter.Update(шелков\_ИДDataSet.Капитан);  }  catch  {  MessageBox.Show("Некоторые поля заполнены в неправильном формате","Ошибка");  }  }  int nameform;  public void nameForm(int a)  {  nameform = a;  } |
| Событие ExitAvtorization\_Click | private void ExitAvtorization\_Click(object sender, EventArgs e)  {  Меню form = new Меню();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие Капитан\_Load | private void Капитан\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "шелков\_ИДDataSet.Капитан". При  необходимости она может быть перемещена или удалена.  this.капитанTableAdapter.Fill(this.шелков\_ИДDataSet.Капитан);  } |
| Событие button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  try  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = Шелков\_ИД; Integrated  Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"Delete from Капитан where [Код капитана] =  '{dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value.ToString()}' ", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  this.капитанTableAdapter.Update(this.шелков\_ИДDataSet.Капитан);  this.капитанTableAdapter.Fill(this.шелков\_ИДDataSet.Капитан);  }  catch  {  MessageBox.Show("Это поле соединено с другой таблицей", "Ошибка");  }  }  }  } |

Меню

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс class Меню | public partial class Меню : Form  {  public Меню()  {  InitializeComponent();  } |
| Событие Exit\_Click | private void Exit\_Click(object sender, EventArgs e)  {  Application.Exit();  }  int nameform; |
| Метод nameForm | public void nameForm(int a)  {  nameform = a;  switch (nameform)  {  case 1:  Capitan.Location = new Point(12, 12);  button2.Location = new Point(12, 41);  button3.Location = new Point(12, 70);  button4.Location = new Point(12, 99);  button5.Location = new Point(12, 128);  button6.Location = new Point(12, 157);  button8.Location = new Point(12, 186);  button1.Location = new Point(12, 215);  Exit.Location = new Point(12, 244);  Size = new Size(254, 314);  break;  case 2:  button8.Visible = false;  button1.Location = new Point(12, 186);  Exit.Location = new Point(12, 215);  Size = new Size(254, 285);  break;  case 3:  button2.Visible = false;  button3.Visible = false;  button5.Visible = false;  button8.Visible = false;  Capitan.Visible = false;  button4.Location = new Point(12, 12);  button6.Location = new Point(12, 41);  button1.Location = new Point(12, 70);  Exit.Location = new Point(12, 99);  Size = new Size(254, 167);  break;  case 4:  button2.Visible = false;  button3.Visible = false;  button5.Visible = false;  button8.Visible = false;  Capitan.Visible = false;  button6.Visible = false;  button4.Location = new Point(12, 12);  button1.Location = new Point(12, 41);  Exit.Location = new Point(12, 70);  Size = new Size(254, 137);  break;  }  } |
| Событие МенюАдмин\_Load | private void МенюАдмин\_Load(object sender, EventArgs e)  {    } |
| Событие Capitan\_Click | private void Capitan\_Click(object sender, EventArgs e)  {  Капитан form = new Капитан();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие button4\_Click | private void button4\_Click(object sender, EventArgs e)  {  Груз form = new Груз();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие button3\_Click | private void button3\_Click(object sender, EventArgs e)  {  Отправитель form = new Отправитель();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  Avtorization form = new Avtorization();  Hide();  form.Show();  } |
| Событие button5\_Click | private void button5\_Click(object sender, EventArgs e)  {  ПортПолучения form = new ПортПолучения();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие button6\_Click | private void button6\_Click(object sender, EventArgs e)  {  Получатель form = new Получатель();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие button2\_Click | private void button2\_Click(object sender, EventArgs e)  {  Судна form = new Судна();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие button8\_Click | private void button8\_Click(object sender, EventArgs e)  {  Пользователи form = new Пользователи();  form.nameForm(nameform);  Hide();  form.Show();  }  }  } |

Отправитель

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс class Отправитель | public partial class Отправитель : Form  {  public Отправитель()  {  InitializeComponent();  } |
| Событие nameForm | public void nameForm(int a)  {  nameform = a;  } |
| Метод ExitAvtorization\_Click | private void ExitAvtorization\_Click(object sender, EventArgs e)  {  Меню form = new Меню();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие Exit\_Click | private void Exit\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Событие Отправитель\_Load | private void Отправитель\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "шелков\_ИДDataSet.Отправитель". При  необходимости она может быть перемещена или удалена.  this.отправительTableAdapter.Fill(this.шелков\_ИДDataSet.Отправитель);  } |
| Событие Update\_Click | private void Update\_Click(object sender, EventArgs e)  {  try  {  отправительTableAdapter.Update(шелков\_ИДDataSet.Отправитель);  }  catch  {  MessageBox.Show("Некоторые поля заполнены в неправильном формате","Ошибка");  }  } |
| Событие button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  try  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = Шелков\_ИД; Integrated  Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"Delete from Отправитель where [ИНН отправителя груза] =  '{dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value.ToString()}' ", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  this.отправительTableAdapter.Update(this.шелков\_ИДDataSet.Отправитель);  this.отправительTableAdapter.Fill(this.шелков\_ИДDataSet.Отправитель);  }  catch  {  MessageBox.Show("Это поле соединено с другой таблицей", "Ошибка");  }  }  }  } |

# Получатель

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс class Получатель | public partial class Получатель : Form  {  public Получатель()  {  InitializeComponent();  } |
| Событие ExitAvtorization\_Click | private void ExitAvtorization\_Click(object sender, EventArgs e)  {  Меню form = new Меню();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие Exit\_Click | private void Exit\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Событие Получатель\_Load | private void Получатель\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "шелков\_ИДDataSet.Получатель". При  необходимости она может быть перемещена или удалена.  this.получательTableAdapter.Fill(this.шелков\_ИДDataSet.Получатель);  } |
| Событие Update\_Click | private void Update\_Click(object sender, EventArgs e)  {  try  {  this.получательTableAdapter.Update(this.шелков\_ИДDataSet.Получатель);  }  catch  {  MessageBox.Show("Некоторые поля заполнены в неправильном формате","Ошибка");  }  } |
| Событие button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  try  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = Шелков\_ИД; Integrated  Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"Delete from Получатель where [ИНН получателя] =  '{dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value.ToString()}' ", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  this.получательTableAdapter.Update(this.шелков\_ИДDataSet.Получатель);  this.получательTableAdapter.Fill(this.шелков\_ИДDataSet.Получатель);  }  catch  {  MessageBox.Show("Это поле соединено с другой таблицей", "Ошибка");  }  }  }  } |

# Пользователи

| Название объекта | Событийно-управляемая процедура |
| --- | --- |
| Класс class Пользователи | public partial class Пользователи : Form  {  public Пользователи()  {  InitializeComponent();  } |
| Событие nameForm | public void nameForm(int a)  {  nameform = a;  } |
| Метод ExitAvtorization\_Click | private void ExitAvtorization\_Click(object sender, EventArgs e)  {  Меню form = new Меню();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие Exit\_Click | private void Exit\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Событие Пользователи\_Load | private void Пользователи\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "шелков\_ИДDataSet.Пользователи".  При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.шелков\_ИДDataSet.Пользователи);  } |
| Событие Update\_Click | private void Update\_Click(object sender, EventArgs e)  {  try  {  this.пользователиTableAdapter.Update(this.шелков\_ИДDataSet.Пользователи);  }  catch  {  MessageBox.Show("Некоторые поля заполнены в неправильном формате", "Ошибка");  }  } |
| Событие button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = Шелков\_ИД; Integrated  Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"Delete from Пользователи where Код\_пользователя =  '{dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value.ToString()}' ", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  this.пользователиTableAdapter.Update(this.шелков\_ИДDataSet.Пользователи);  this.пользователиTableAdapter.Fill(this.шелков\_ИДDataSet.Пользователи);  }  }  } |

# Порт получения

|  |  |
| --- | --- |
| Событие class ПортПолучения | public partial class ПортПолучения : Form  {  public ПортПолучения()  {  InitializeComponent();  } |
| Событие nameForm | public void nameForm(int a)  {  nameform = a;  } |
| Событие ExitAvtorization\_Click | private void ExitAvtorization\_Click(object sender, EventArgs e)  {  Меню form = new Меню();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие Exit\_Click | private void Exit\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Событие ПортПолучения\_Load | private void ПортПолучения\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "шелков\_ИДDataSet.Порт\_получения".  При необходимости она может быть перемещена или удалена.  this.порт\_полученияTableAdapter.Fill(this.шелков\_ИДDataSet.Порт\_получения);  } |
| Событие button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  try  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = Шелков\_ИД; Integrated  Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"Delete from Пользователи where [Код получения] =  '{dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value.ToString()}' ", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  this.порт\_полученияTableAdapter.Update(this.шелков\_ИДDataSet.Порт\_получения);  this.порт\_полученияTableAdapter.Fill(this.шелков\_ИДDataSet.Порт\_получения);  }  catch  {  MessageBox.Show("Это поле соединено с другой таблицей", "Ошибка");  }  }  }  } |
| Событие Update\_Click | private void Update\_Click(object sender, EventArgs e)  {  try  {  this.порт\_полученияTableAdapter.Update(this.шелков\_ИДDataSet.Порт\_получения);  }  catch  {  MessageBox.Show("Некоторые поля заполнены в неправильном формате","Ошибка");  }  } |

# Регистрация

|  |  |
| --- | --- |
| Событие class Регистрация | public partial class Регистрация : Form  {  public Регистрация()  {  InitializeComponent();  } |
| Событие ExitAvtorization\_Click | private void ExitAvtorization\_Click(object sender, EventArgs e)  {  Avtorization form = new Avtorization();  Hide();  form.Show();  } |
| Событие Registration\_Click | private void Registration\_Click(object sender, EventArgs e)  {  if ((textBox1.Text == "") || (textBox2.Text == "") || (textBox3.Text == "") || (textBox4.Text == "") ||  (textBox5.Text == ""))  {  MessageBox.Show("Одно из полей не заполнено!!!");  }  else  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = Шелков\_ИД; Integrated  Security = True");  sqlConnect.Open();  string code = dataGridView1.Rows[dataGridView1.RowCount-2].Cells[0].Value.ToString();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Пользователи (Код\_пользователя,  Логин,Пароль,Фамилия,Имя,Отчество,Права\_доступа) VALUES  ('{int.Parse(code)+1}','{textBox1.Text}','{textBox2.Text}','{textBox3.Text}','{textBox4.Text}','{textBox5.Text}','Пользо  ватель');", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  MessageBox.Show($"Регистрация под логином {textBox1.Text} выполнена успешно!", "Успешно!");  Hide();  Avtorization form = new Avtorization();  form.Show();  }  } |
| Событие Exit\_Click | private void Exit\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Событие textBox1\_KeyPress | private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)  {  if ((e.KeyChar >= 'a') && (e.KeyChar <= 'z') || (e.KeyChar >= 'а') && (e.KeyChar <= 'я') || (e.KeyChar >= 'А') &&  (e.KeyChar <= 'Я') || (e.KeyChar >= 'A') && (e.KeyChar <= 'Z') || (e.KeyChar == (char)Keys.Enter) || (e.KeyChar ==  (char)Keys.Back))  {  }  else e.KeyChar -= e.KeyChar;  } |
| Событие textBox5\_KeyPress | private void textBox5\_KeyPress(object sender, KeyPressEventArgs e)  {  if ((e.KeyChar >= 'a') && (e.KeyChar <= 'z') || (e.KeyChar >= 'а') && (e.KeyChar <= 'я') || (e.KeyChar >= 'А') &&  (e.KeyChar <= 'Я') || (e.KeyChar >= 'A') && (e.KeyChar <= 'Z') || (e.KeyChar == (char)Keys.Enter) || (e.KeyChar ==  (char)Keys.Back))  {  }  else e.KeyChar -= e.KeyChar;  } |
| Событие textBox2\_KeyPress | private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)  {  if (((e.KeyChar <= '9') && (e.KeyChar >= '0')) || (e.KeyChar == (char)Keys.Enter) || (e.KeyChar ==  (char)Keys.Back))  {  }  else e.KeyChar -= e.KeyChar;  } |
| Событие Регистрация\_Load | private void Регистрация\_Load(object sender, EventArgs e)  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = Шелков\_ИД; Integrated  Security = True");  sqlConnect.Open();  SqlDataAdapter daa = new SqlDataAdapter($"select Код\_пользователя from Пользователи order by  Код\_пользователя", sqlConnect);  DataSet ds = new DataSet();  daa.Fill(ds);  dataGridView1.DataSource = ds.Tables[0];  }  }  } |
| Событие textBox4\_KeyPress | private void textBox4\_KeyPress(object sender, KeyPressEventArgs e)  {  if ((e.KeyChar >= 'a') && (e.KeyChar <= 'z') || (e.KeyChar >= 'а') && (e.KeyChar <= 'я') || (e.KeyChar >= 'А') &&  (e.KeyChar <= 'Я') || (e.KeyChar >= 'A') && (e.KeyChar <= 'Z') || (e.KeyChar == (char)Keys.Enter) || (e.KeyChar ==  (char)Keys.Back))  {  }  else e.KeyChar -= e.KeyChar;  } |
| Событие textBox3\_KeyPress | private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)  {  if ((e.KeyChar >= 'a') && (e.KeyChar <= 'z') || (e.KeyChar >= 'а') && (e.KeyChar <= 'я') || (e.KeyChar >= 'А') &&  (e.KeyChar <= 'Я') || (e.KeyChar >= 'A') && (e.KeyChar <= 'Z') || (e.KeyChar == (char)Keys.Enter) || (e.KeyChar ==  (char)Keys.Back))  {  }  else e.KeyChar -= e.KeyChar;  } |

Судна

|  |  |
| --- | --- |
| Событие class Судна | public partial class Судна : Form  {  public Судна()  {  InitializeComponent();  } |
| Событие ExitAvtorization\_Click | private void ExitAvtorization\_Click(object sender, EventArgs e)  {  Меню form = new Меню();  form.nameForm(nameform);  Hide();  form.Show();  } |
| Событие Exit\_Click | private void Exit\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Событие Update\_Click | private void Update\_Click(object sender, EventArgs e)  {  try  {  this.суднаTableAdapter.Update(this.шелков\_ИДDataSet.Судна);  }  catch  {  MessageBox.Show("Некоторые поля заполнены в неправильном формате","Ошибка");  }  } |
| Событие dataGridView1\_CellContentClick | private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  } |
| Событие button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  try  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = Шелков\_ИД; Integrated  Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter($"Delete from Судна where [Регистрационный номер судна] =  '{dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value.ToString()}' ", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  this.суднаTableAdapter.Update(this.шелков\_ИДDataSet.Судна);  this.суднаTableAdapter.Fill(this.шелков\_ИДDataSet.Судна);  }  catch  {  MessageBox.Show("Это поле соединено с другой таблицей", "Ошибка");  }  }  }  } |
| Событие Судна\_Load | private void Судна\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "шелков\_ИДDataSet.Судна". При  необходимости она может быть перемещена или удалена.  this.суднаTableAdapter.Fill(this.шелков\_ИДDataSet.Судна);  } |

# 5 РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ

## 5.1 Назначение эксперимента. Выбор и обоснование методики проведения тестирования

## 5.2 Технология тестирования

## 5.3 Результаты проведения тестирования

# ЗАКЛЮЧЕНИЕ

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

# ПРИЛОЖЕНИЕ A

# 

Рисунок 1 - Диаграмма вариантов использования

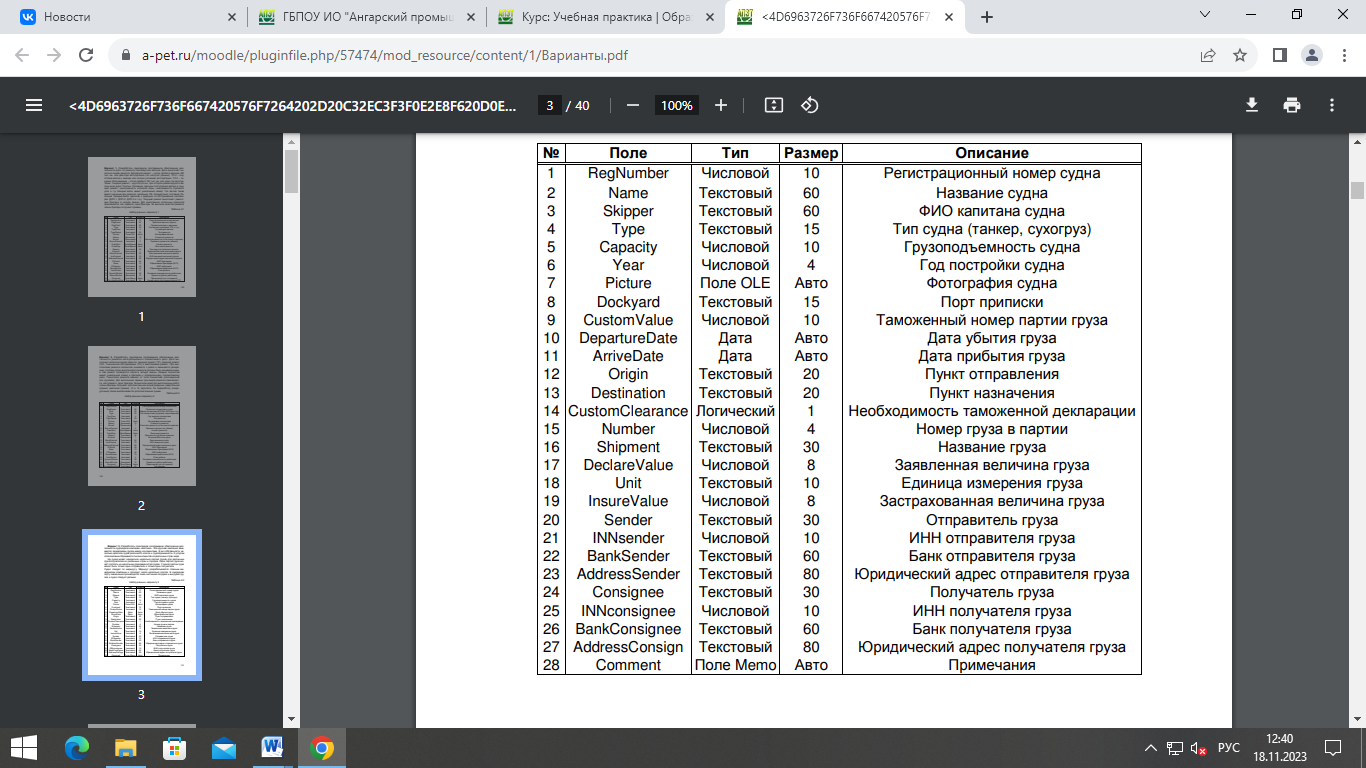


Рисунок 2 - Данные предоставленные вариантом учебной практики

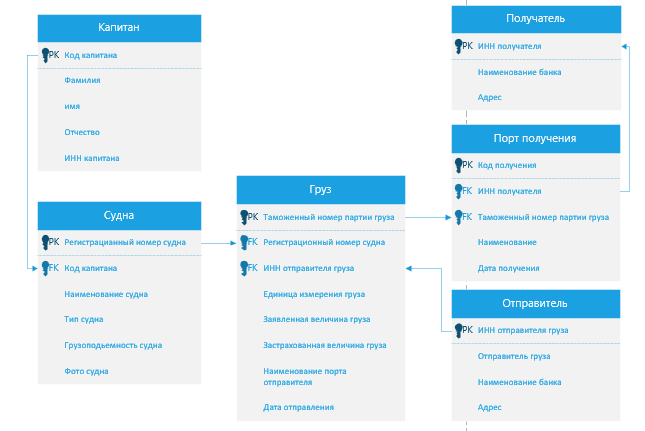


Рисунок 3 - Логическая модель данных

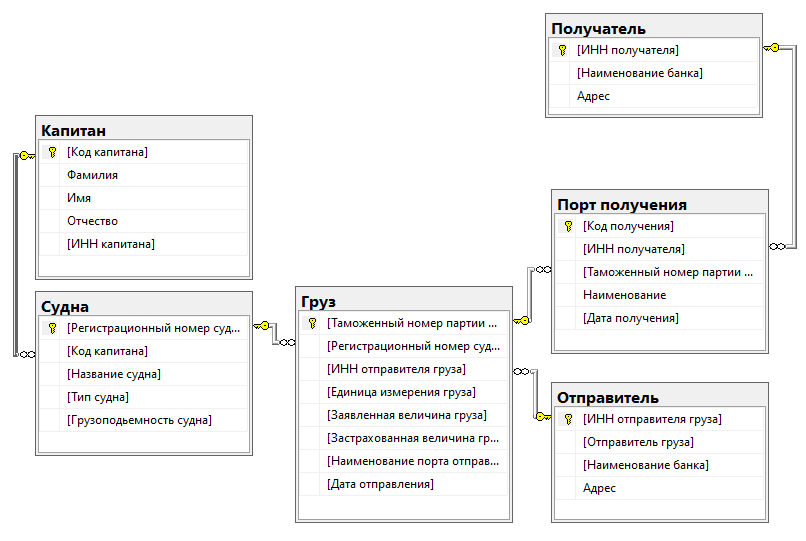


Рисунок 4 – Физическая модель базы данных