СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc166875470)

[1 ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ 4](#_Toc166875471)

[1.1 Описание предметной области 4](#_Toc166875472)

[1.2 Группы пользователей информационной системы 4](#_Toc166875473)

[1.3 Основные требования, предъявляемые к информационной системе 5](#_Toc166875474)

[2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 7](#_Toc166875475)

[2.1 Моделирование некоторых динамических аспектов системы 7](#_Toc166875476)

[2.1.1 Создание диаграммы вариантов использования 8](#_Toc166875477)

[2.1.2 Создание диаграммы деятельности 9](#_Toc166875478)

[2.2 Разработка функциональной структуры информационной системы. Карта навигации по информационной системе 12](#_Toc166875479)

[2.3 Разработка макета информационной системы 13](#_Toc166875480)

[3 РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ 17](#_Toc166875481)

[3.1 Моделирование и разработка базы данных 17](#_Toc166875482)

[3.2 Разработка программных модулей информационной системы 20](#_Toc166875483)

[4 РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ 48](#_Toc166875484)

[5.1 Назначение эксперимента. Выбор и обоснование методики проведения тестирования 48](#_Toc166875485)

[5.2 Технология тестирования 48](#_Toc166875486)

[5.3 Результаты проведения тестирования 50](#_Toc166875487)

ВВЕДЕНИЕ

Развитие сельского хозяйства и садоводства неразрывно связано с качеством посевного материала. Институт, сфокусированный на сборе, разведении и распространении разнообразных семян, играет ключевую роль в этом процессе. Его обширный ассортимент охватывает не только базовые сорта, но и редкие и экзотические растения, создавая возможность для разносторонних аграрных и садоводческих проектов.

Компания активно работает как с оптовыми, так и розничными покупателями, стремясь обеспечить максимальную доступность своего ассортимента. Оптовые партнеры, занесенные в специализированную базу данных, получают информацию о новых поступлениях или изменениях в наличии сортов, что обеспечивает оперативность их реагирования на рыночные изменения и потребности клиентов.

Этот контекст создает необходимость в оптимизации процессов управления ассортиментом, углубленном изучении характеристик семян и повышении уровня обслуживания покупателей. В данном задании будет рассмотрена детальная стратегия по улучшению информационной базы, оптимизации системы поиска и выбора семян, а также укреплению взаимодействия с оптовыми партнерами с целью укрепления позиций института на рынке качественных семян.

# 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

Необходимо разработать и спроектировать прикладное программное обеспечение для эффективного управления продажами сортов семян.

Данный институт занимается сбором, выведением и продажей различных сортов семян. В его ассортименте можно найти семена практически всех возможных видов растений: от помидоров до редких цветов. Только что выведенные сорта заносятся в отдельный список для дальнейшего тестирования. Каждый сорт семян имеет свои характеристики, такие как морозоустойчивость, адаптация к местным условиям, сроки созревания (раннеспелый, среднеспелый, поздний) и т.п. Покупатель может выбрать сорт, отвечающий тем или иным характеристикам. Компания занимается как оптовыми, так и розничными продажами. Оптовые покупатели заносятся в базу главным образом для того, чтобы информировать их о поступлении новых или отсутствовавших в определенный момент в продаже сортов.

## Группы пользователей информационной системы

Пользователи, в зависимости от своих функциональных обязанностей, могут выполнять действия, направленные на мониторинг, управление, анализ данных и обеспечение безопасности в рамках проектируемой информационной системы "Учёт продаж сортов растений".

Роли и их обязанности:

* + Гость: просмотр данных.
  + Администратор: управление системами аутентификации, авторизации и шифрования; оптимизация работы системы для обеспечения высокой производительности; полный доступ ко всей системе.
  + Покупатель: получение информации обо всех доступных сортах растений и упаковок, в которых хранится определённый сорт.
  + Продавец: получение информации обо всех сортах растений, упаковок, в которых хранится определённый сорт, новых сортах, продажах, продавцах, покупателях и оптовых покупателях.

## Основные требования, предъявляемые к информационной системе

Гибкость:

* Система должна обладать высокой степенью конфигурируемости, чтобы пользователи могли адаптировать её под различные потребности селекционного института;
* Возможность легкого масштабирования системы при увеличении объема данных или числа пользователей.

Надёжность:

* 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
* Обеспечение высокой производительности при обработке данных, особенно в условиях одновременного использования множества пользователей

Безопасность:

* Реализация надежной системы аутентификации для проверки личности пользователей и установление строгих прав доступа.
* Использование шифрования для защиты передаваемой и хранимой чувствительной информации.
* Ведение журнала аудита для отслеживания действий пользователей и обнаружения возможных угроз безопасности.

Вывод по главе: было описано, что такое предметная область и её описание; группы пользователей, которые будут пользоваться информационной системой «Учёт продаж сортов растений» и основные требования к ней.

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

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 в области:

* обработки объектов будущей базы данных;
* написания программ (в том числе – отчётных и экранных форм), обеспечивающих выполнение запросов к данным;
* выполнения учёта функционирования конкретной среды (технологии).

Если выделять стадию проектирования информационных систем в качестве отдельного этапа, то его можно разместить между этапами анализа и разработки. Однако на практике чёткое разделение на этапы, как правило, затруднено или невозможно, поскольку проектирование, формально начинаясь с определения цели проекта, часто продолжается на стадиях тестирования и реализации.

## 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 (в том числе активные), интерфейсы, компоненты и узлы.

Использовать диаграммы деятельности для моделирования некоторых динамических аспектов системы вы можете в контексте почти любого моделируемого элемента. Однако чаще вы будете прибегать к таким диаграммам в контексте всей системы, подсистемы, операции или класса. Диаграмму деятельности можно присоединить к варианту использования, чтобы моделировать сценарий, и к кооперации, чтобы моделировать динамические аспекты поведения совокупности объектов.

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.

В приложении А на рисунке 1 представлена диаграмма вариантов использования по информационной системе «Бюро технической инвентаризации »

# 2.1.1 Создание диаграммы вариантов использования



Рисунок 1 – Диаграмма вариантов использования

На диаграмме представлены пользователи такие как администратор, продавец, покупатель и гость.

# 2.1.2 Создание диаграммы деятельности

На рисунках 2-5 представлена диаграмма деятельности (по ролям) информационной системы «Учёт продаж сортов растений». 

Рисунок 2 – Диаграмма деятельности (Администратор)

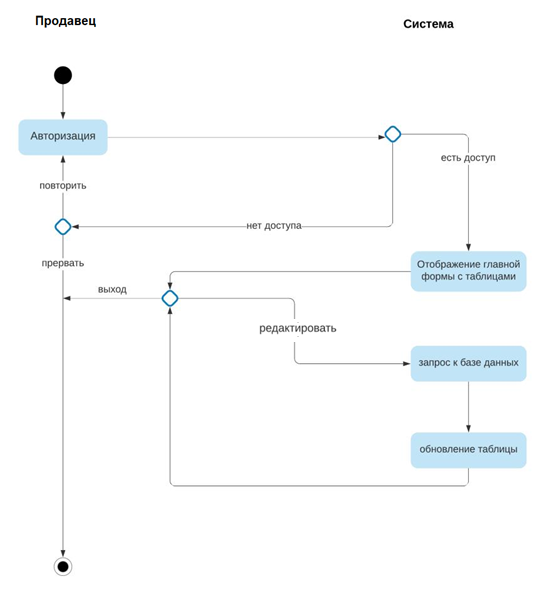


Рисунок 3 – Диаграмма деятельности (Продавец)

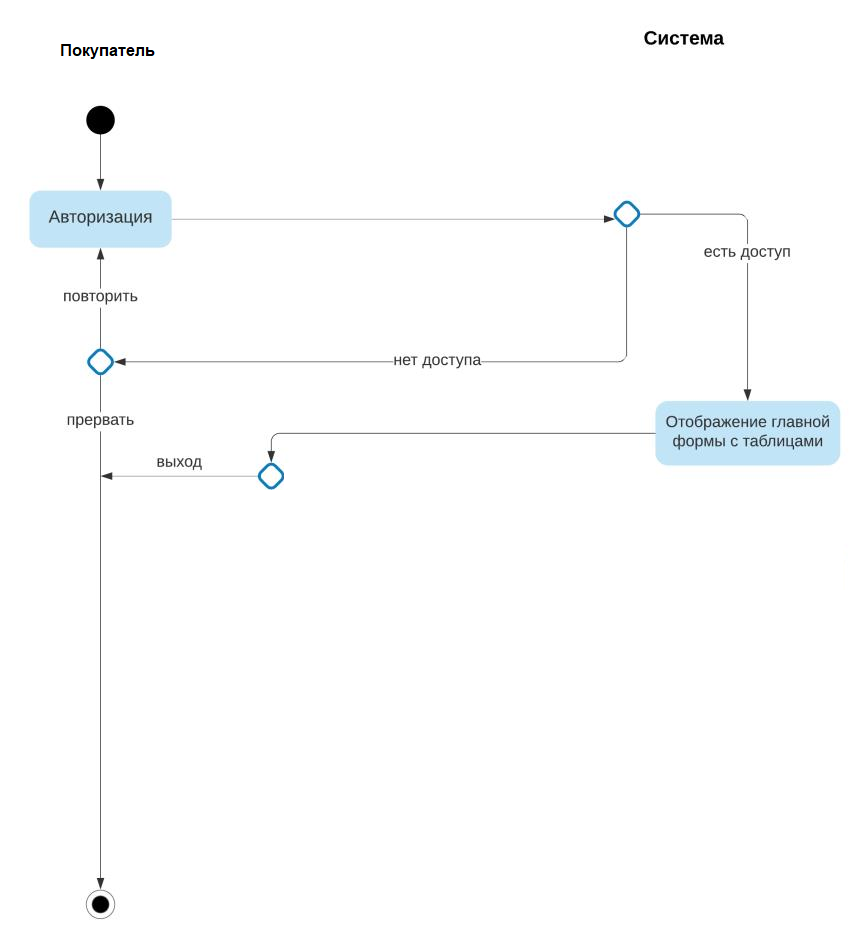


Рисунок 4 – Диаграмма деятельности (Покупатель)

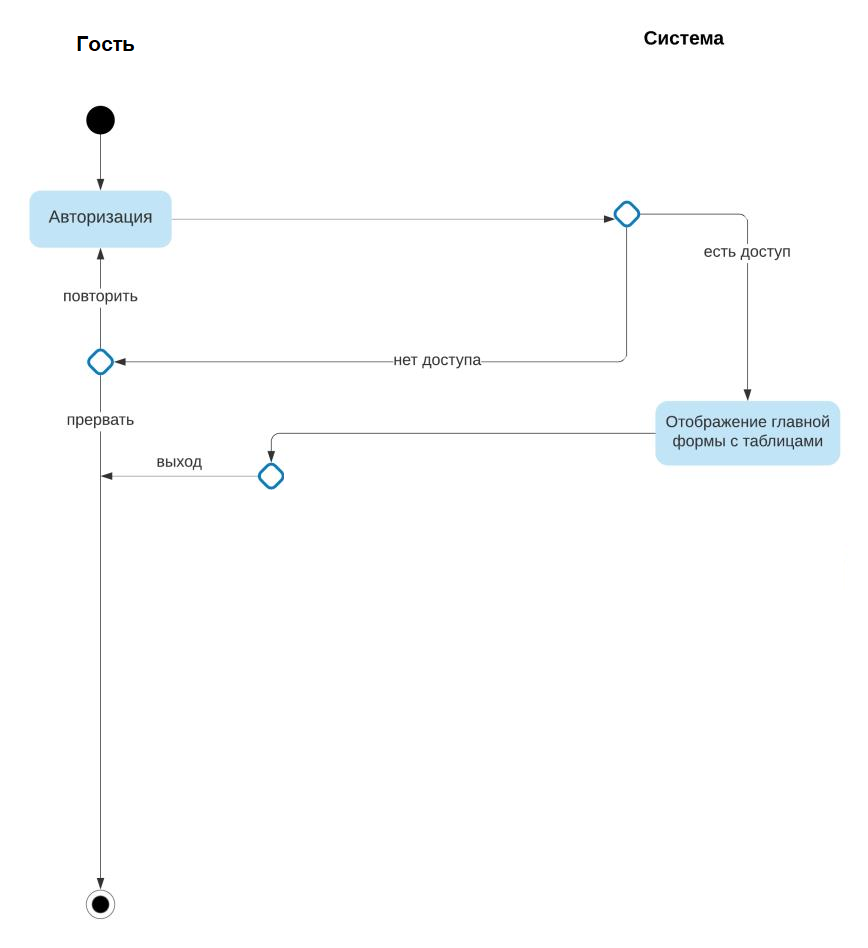


Рисунок 5 – Диаграмма деятельности (Гость)

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

На рисунке 6 представлена карта переходов по информационной системе «Учёт продаж сортов растений»

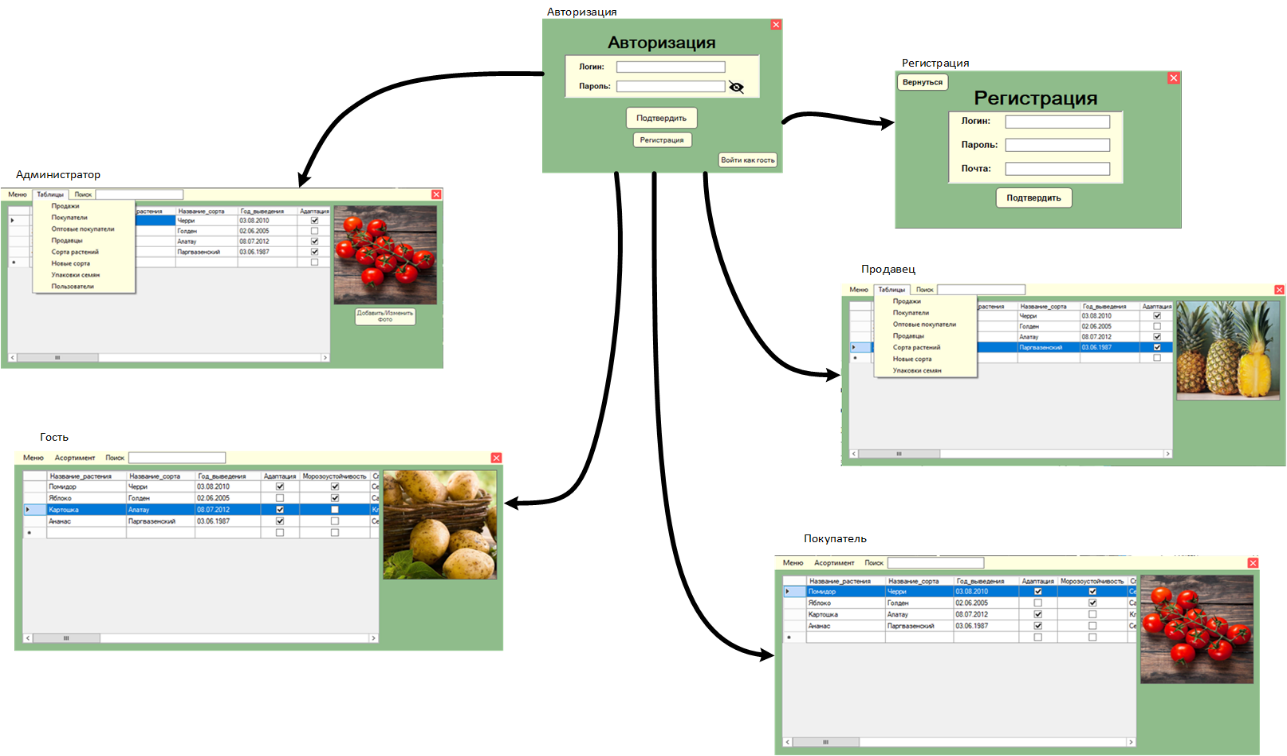


Рисунок 6 – Карта переходов

## Разработка макета информационной системы

На рисунке 7 представлено окно авторизации.

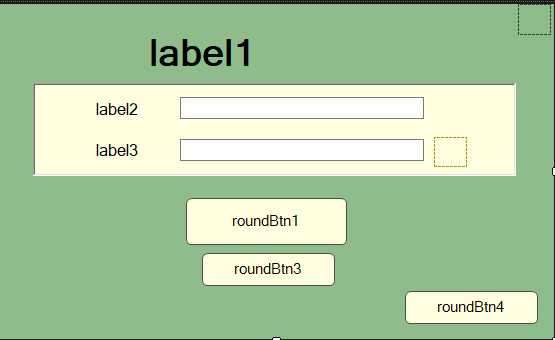


Рисунок 7 – Окно авторизации

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 «Регистрация». В этом случае откроется окно регистрации, представленное в приложении “А” на рисунке 8.

На этапе регистрации пользователю необходимо придумать свой логин и пароль, а также ввести персональные данные, такие как почта.

Если сотрудник с введенными данными не зарегистрирован в базе

данных, в качестве сотрудника, то пользователь не сможет зарегистрироваться, и, соответственно, продолжить работу, так как данная информационная система предназначена только для сотрудников образовательной организации.

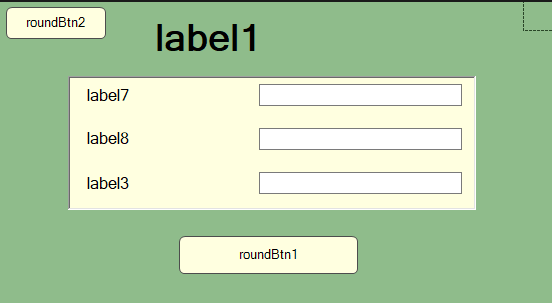


Рисунок 8 – Окно регистрации

После того, как пользователь успешно прошел авторизацию, он может продолжить работу с таблицами на главной форме. На рисунке 9-12 представлена главная форма, которая будет отображена, в зависимости от прав пользователя в системе. Например, продавец сможет изменять только определённые таблицы, а для гостя будет недоступны функции редактирования, добавления и удаления записи, так как данной группе пользователей доступен только просмотр данных.

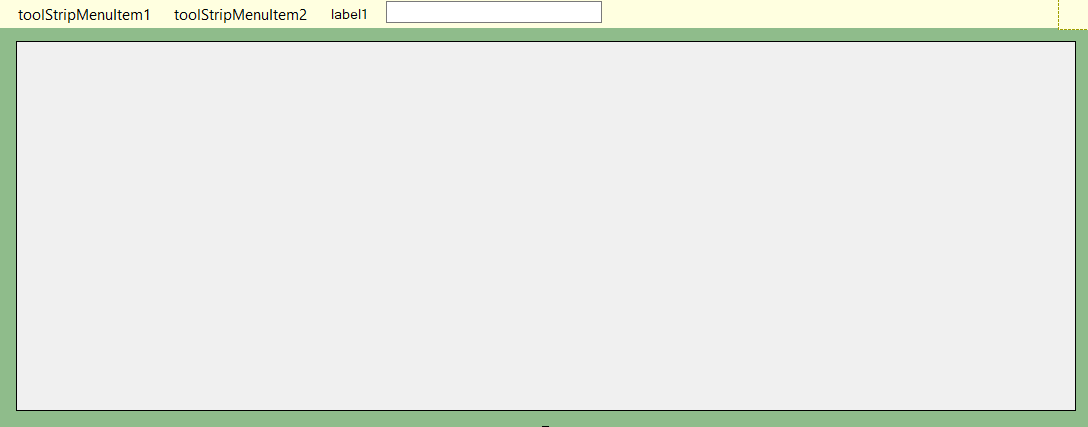


Рисунок 9 – Форма «Администратора»

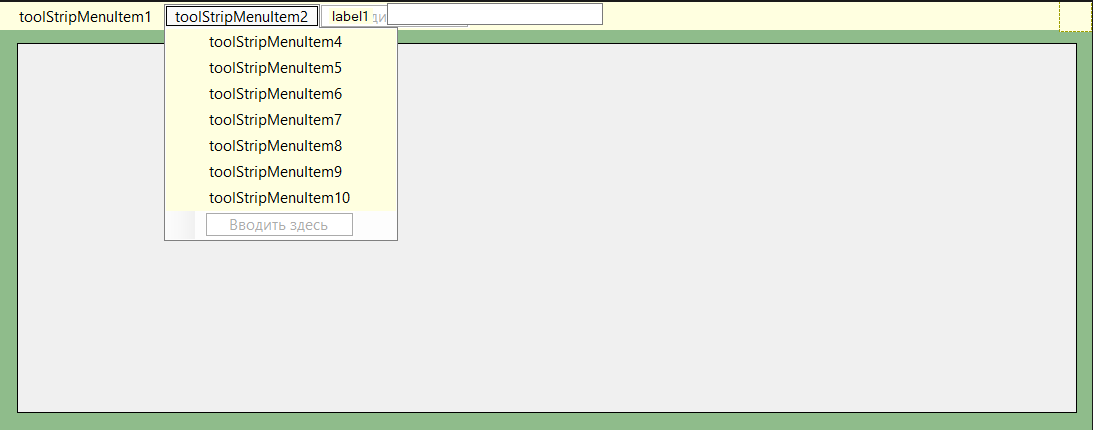


Рисунок 10 – Форма «Продавца»

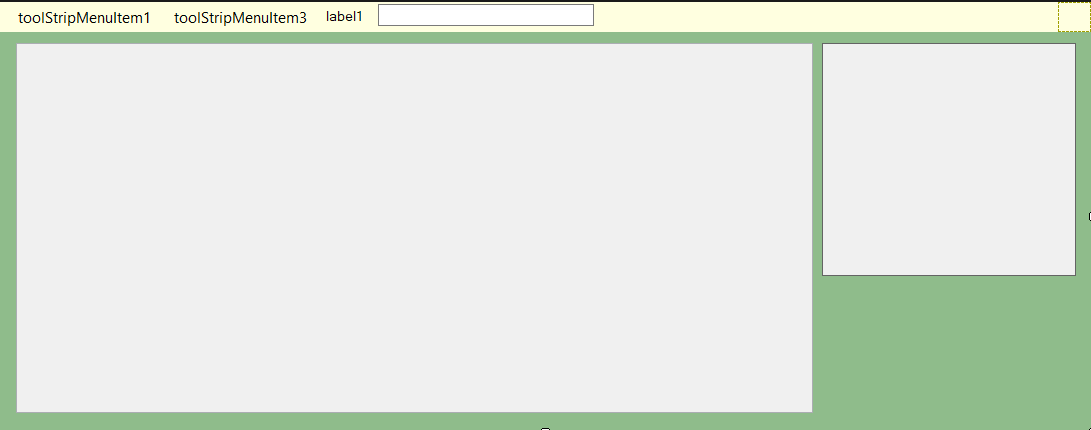


Рисунок 11 – Форма «Покупателя»

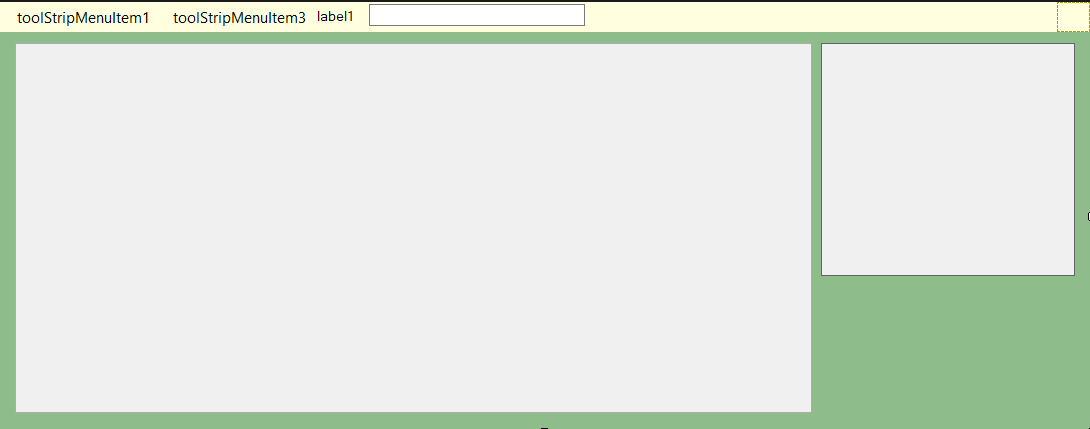


Рисунок 12 – Форма «Гостя»

Аналогичным образом в зависимости от роли пользователя выглядит главная форма, на которой расположены все необходимые таблицы. Например, администратор сможет изменять все таблицы, а для сотрудника будет доступно только изменять некоторые таблицы.

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

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

# РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ

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: «Сорта растений», «Новые сорта растений», «Упаковки семян», «Покупатели», «Оптовые покупатели», «Продавцы», «Продажи», «Пользователи».

На рисунке 13 представлена логическая модель данных информационной системы «Учёт продаж сортов растений».

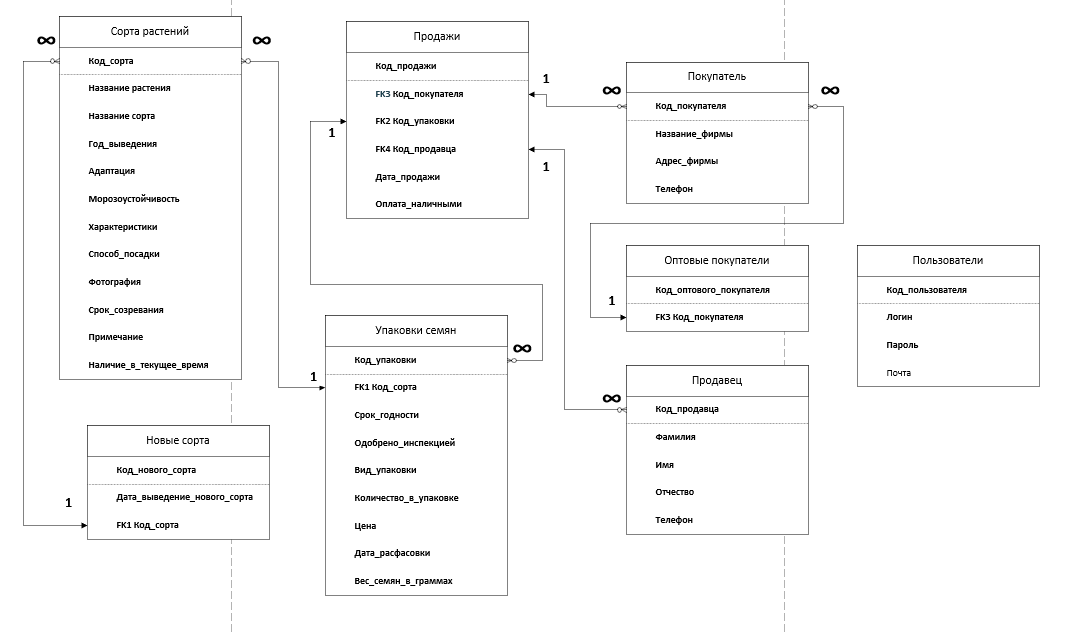


Рисунок 13 – Логическая модель данных

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 «Сорта растений»

| Название поля | Тип данных |
| --- | --- |
| Код\_сорта\_растения | Числовой |
| Название\_растения | Текстовый |
| Название\_сорта | Текстовый |
| Год\_выведения | Дата |
| Адаптация | Логический |
| Морозоустойчивость | Логический |
| Характеристики | Текстовый |
| Способ\_посадки | Текстовый |
| Фотография | OLE |
| Срок\_созревания | Текстовый |
| Примечание | Текстовый |
| Наличие\_в\_текущее\_время | Логический |

Таблица 2 – Поля таблицы «Новые сорта растений»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код\_нового\_сорта\_растения | Числовой |
| Дата\_выведения\_нового\_сорта | Дата |
| Код\_сорта\_растения | Числовой |

Таблица 3 – Поля таблицы «Упаковки семян»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код\_упаковки\_семян | Числовой |
| Срок\_годности | Дата |
| Одобрено\_инспекцией | Логический |
| Вид\_упаковки | Текстовый |
| Количество\_в\_упаковке | Числовой |
| Цена | Денежный |
| Дата\_расфасовки | Дата |
| Вес\_семян\_в\_граммах | Числовой |
| Код\_сорта\_растения | Числовой |

Таблица 4 – Поля таблицы «Покупатели»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код\_покупателя | Числовой |
| Название\_фирмы | Дата |
| Адрес\_фирмы | Текстовый |
| Телефон | Текстовый |

Таблица 5 – Поля таблицы «Оптовые покупатели»

| Название поля | Тип данных |
| --- | --- |
| Код\_оптового\_покупателя | Числовой |
| Код\_покупателя | Числовой |

Таблица 6 – Поля таблицы «Продавцы»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код\_продавца | Числовой |
| Фамилия | Текстовый |
| Имя | Текстовый |
| Отчество | Текстовый |
| Телефон | Текстовый |

Таблица 7 – Поля таблицы «Продажи»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код\_продажи | Числовой |
| Код\_покупателя | Числовой |
| Код\_упаковки\_семян | Числовой |
| Код\_продавца | Числовой |
| Дата\_продажи | Дата |
| Оплата\_наличными | Логический |

Таблица 8 – Поля таблицы «Пользователи»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Логин | Текстовый |
| Пароль | Числовой |
| Почта | Текстовый |

На рисунке 14 представлены все созданные таблицы в обозревателе объектов базы данных.

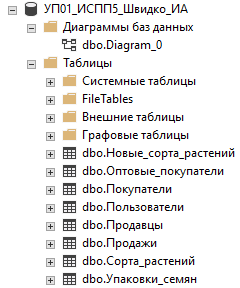
****

Рисунок 14 – Созданные таблицы

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

На рисунке 15 представлена физическая модель данных, отражающая все таблицы и их поля, а также связи между таблицами.

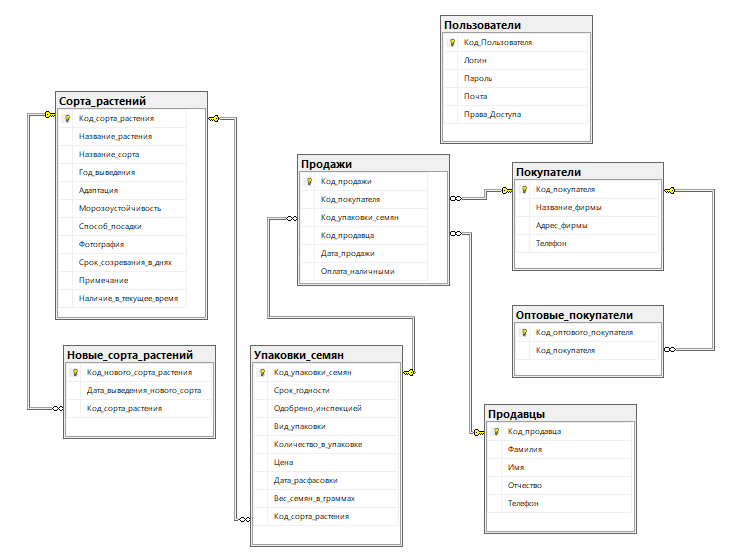


Рисунок 15 – Физическая модель базы данных

## Разработка программных модулей информационной системы

Разработанные в MS Visual Studio формы продемонстрированы на рисунках 16-21.

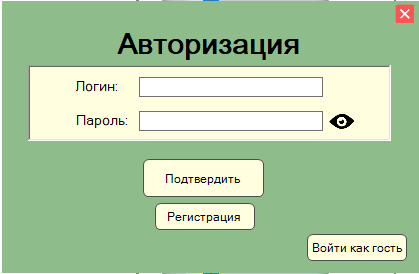


Рисунок 16 – Форма “Авторизация”

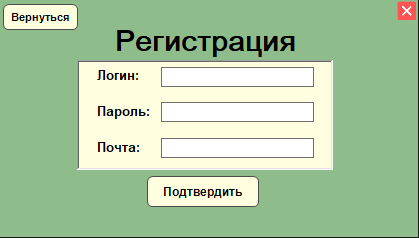


Рисунок 17 – Форма “Регистрация”

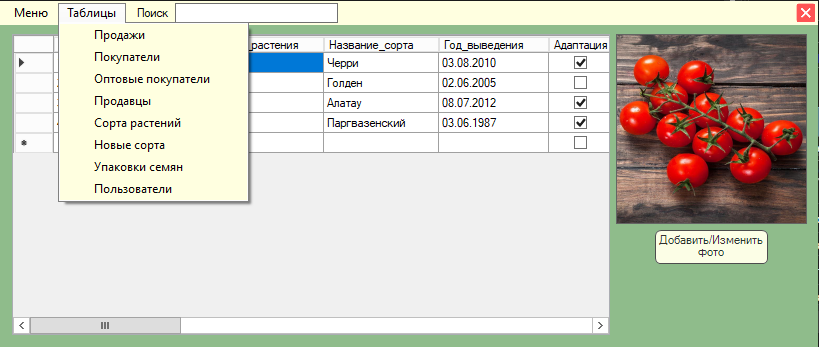


Рисунок 18 – Форма “ Администратора”

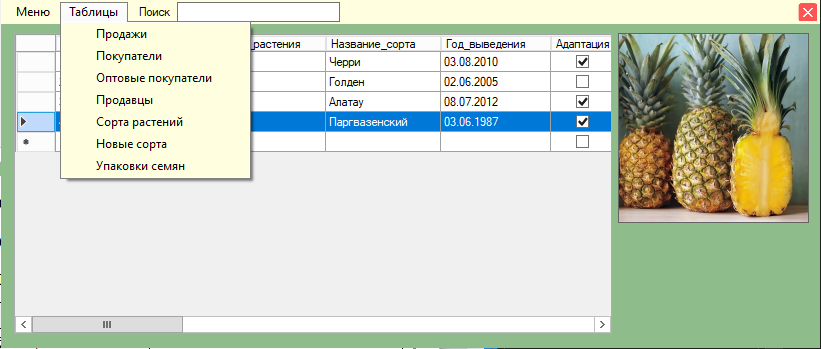


Рисунок 19 – Форма “Продавца”

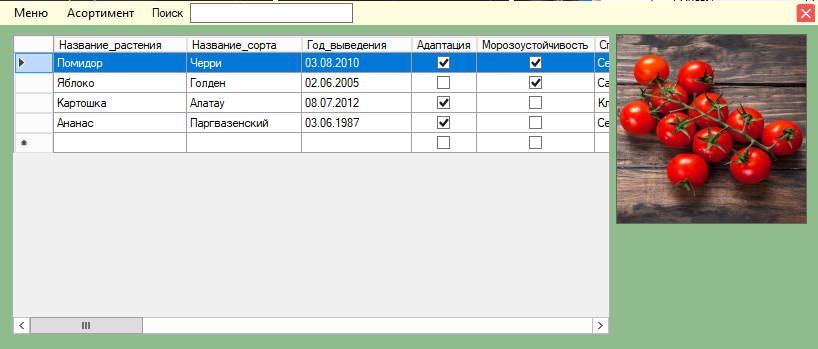


Рисунок 20 – Форма «Покупателя»

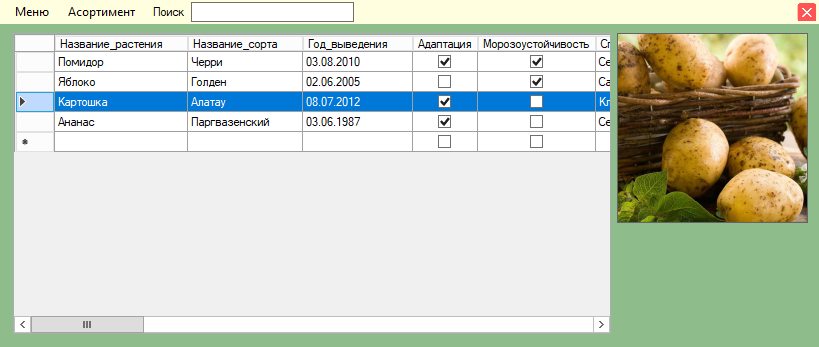


Рисунок 21 – Форма «Гостя»

Свойства элементов формы Авторизация представлены в таблице 9.

Таблица 9 – Свойства элементов формы Авторизация

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Авторизация | FormBorderStyle | None |
| Text | Авторизация |
| StartPosition | CenterScreen |
| BackColor | DarkSeaGreen |
| Size | 409; 272 |
| Авторизация\_label3 | Text | Авторизация |
| Font | Arial Unicode MS; 21,75pt; style=Bold |
| Видимость\_пароля\_PictureBox2 | Size | 25; 24 |
| SizeMode | StretchImage |
| выйти\_PictureBox1 | Size | 25; 25 |
| SizeMode | StretchImage |
| Кнопка\_Войти\_как\_гость\_roundBtn4 | Size | 100; 27 |
| Text | Войти как гость |
| Font | Microsoft Sans Serif; 9,25pt |
| Кнопка\_подтверждения\_roundBtn1 | Size | 121; 38 |
| Text | Подтвердить |
| Font | Microsoft Sans Serif; 10,25pt |
| BackColor | LightYellow |
| Кнопка\_Рагистрация\_roundBtn3 | Size | 100; 27 |
| Text | Регистрация |
| Font | Microsoft Sans Serif; 9,25pt |
| BackColor | LightYellow |
| Логин\_label1 | Size | 49; 16 |
| Text | Логин: |
| Font | Arial; 9,75pt; style=Bold |
| Пароль\_label2 | Size | 60; 16 |
| Text | Пароль: |
| Font | Arial; 9,75pt; style=Bold |
| Логин\_textBox1 | Font | Microsoft Sans Serif; 8,25pt |
| Size | 184; 20 |
| Пароль\_textBox2 | Font | Microsoft Sans Serif; 8,25pt |
| Size | 184; 20 |
| Скрыть\_пароль\_PictureBox3 | Size | 25; 24 |
| SizeMode | StretchImage |
| Фон\_panel1 | BackColor | LightYellow |
| Size | 330; 76 |
| BorderStyle | Fixed3D |

Событийно-управляемые процедуры в форме авторизации представлены в таблице 10

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

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| class Авторизация | public partial class Авторизация : Form  {  public int ogr = 5;  public Авторизация()  {  InitializeComponent();  Видимость\_пароля\_PictureBox2.Visible = false;  Скрыть\_пароль\_PictureBox3.Visible = true;  }  } |
| Кнопка\_подтверждения\_Click | private void Кнопка\_подтверждения\_Click(object sender, EventArgs e) //Переходит на форму 'Администратор', 'Продавец' и 'Покупатель'  {  if (Логин\_textBox1.Text == "" || Пароль\_textBox2.Text == "")  {  MessageBox.Show("Поля не должны быть пустыми", "Ошибка");  }  else  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Швидко\_ИА; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter("select \* from Пользователи", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  int flag = -1;  for (int i = 0; i < dt.Rows.Count; i++)  {  if ((dt.Rows[i]["Права\_Доступа"].ToString() == "Администратор") && (dt.Rows[i]["Логин"].ToString() == Логин\_textBox1.Text) && (dt.Rows[i]["Пароль"].ToString() == Пароль\_textBox2.Text))  flag = 0;  if ((dt.Rows[i]["Права\_Доступа"].ToString() == "Продавец") && (dt.Rows[i]["Логин"].ToString() == Логин\_textBox1.Text) && (dt.Rows[i]["Пароль"].ToString() == Пароль\_textBox2.Text))  {  flag = 1;  }  if ((dt.Rows[i]["Права\_Доступа"].ToString() == "Покупатель") && (dt.Rows[i]["Логин"].ToString() == Логин\_textBox1.Text) && (dt.Rows[i]["Пароль"].ToString() == Пароль\_textBox2.Text))  {  flag = 2;  }  }  if ((flag == 2) || (flag == 1) || (flag == 0))  {  if (flag == 0)  {  this.Hide();  Администратор UsrFrm = new Администратор();  UsrFrm.Show();  }  if (flag == 1)  {  this.Hide();  Продавец UsrFrm5 = new Продавец();  UsrFrm5.Show();  }  if (flag == 2)  {  this.Hide();  Покупатель UsrFrm6 = new Покупатель();  UsrFrm6.Show();  }  }  else  MessageBox.Show("Пересоздайте пользователя! Такого пользователя не существует.", "Ошибка");  }  } |
| Кнопка\_Рагистрация\_Click | private void Кнопка\_Рагистрация\_Click(object sender, EventArgs e) //Переходит на форму 'Регистрация'  {  this.Hide();  Регистрация UsrFrm4 = new Регистрация();  UsrFrm4.Show();  } |
| Кнопка\_Войти\_как\_гость\_Click | private void Кнопка\_Войти\_как\_гость\_Click(object sender, EventArgs e) //Переходит на форму 'Гость'  {  this.Hide();  Гость UsrFrm3 = new Гость();  UsrFrm3.Show();  } |
| выйти\_Click | private void выйти\_Click(object sender, EventArgs e) //Прекращает работу программы  {  Application.Exit();  } |
| Пароль\_KeyPress | private void Пароль\_KeyPress(object sender, KeyPressEventArgs e) //Проверяет на отсутствие букв  {  if (!char.IsDigit(e.KeyChar) && (e.KeyChar != (char)Keys.Back))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать буквы", "Ошибка");  }  } |
| Авторизация\_MouseMove | private void Авторизация\_MouseMove(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Авторизация\_MouseDown | private void Скрыть\_пароль\_PictureBox3\_Click(object sender, EventArgs e) //Скрывает пароль  {  Пароль\_textBox2.UseSystemPasswordChar = false;  Видимость\_пароля\_PictureBox2.Visible = true;  Скрыть\_пароль\_PictureBox3.Visible = false;  } |
| Показать\_пароль\_Click | private void Видимость\_пароля\_PictureBox2\_Click(object sender, EventArgs e) //Показывает пароль  {  Пароль\_textBox2.UseSystemPasswordChar = true;  Видимость\_пароля\_PictureBox2.Visible = false;  Скрыть\_пароль\_PictureBox3.Visible = true;  } |

Свойства элементов формы Регистрация представлены в таблице 11.

Таблица 11 – Свойства элементов формы Регистрация

| Объект | Свойство | Значение |
| --- | --- | --- |
| Регистрация | FormBorderStyle | None |
| Text | Регистрация |
| StartPosition | CenterScreen |
| BackColor | DarkSeaGreen |
| Size | 409; 272 |
| ShowIcon | False |
| Регистрация\_label1 | Text | Регистрация |
| Font | Arial Unicode MS; 21,75pt; style=Bold |
| Выйти\_pictureBox1 | Size | 25; 25 |
| SizeMode | StretchImage |
| Вернуться\_roundBtn2 | Size | 75; 26 |
| Text | Вернуться |
| Font | Arial; 8,25pt; style=Bold |
| BackColor | LightYellow |
| Логин\_textBox5 | Font | Microsoft Sans Serif; 8,25pt |
| Size | 153; 20 |
| Пароль\_label8 | Text | Пароль: |
| Font | Arial; 9,75pt; style=Bold |
| Пароль\_textBox4 | Font | Microsoft Sans Serif; 8,25pt |
| Size | 153; 20 |
| Подтвердить\_roundBtn1 | Size | 112; 31 |
| Text | Подтвердить |
| Font | Arial; 8,25pt; style=Bold |
| BackColor | LightYellow |
| Почта\_label3 | Text | Почта: |
| Font | Arial; 9,75pt; style=Bold |
| Почта\_textBox2 | Font | Microsoft Sans Serif; 8,25pt |
| Size | 153; 20 |
| Фон\_panel1 | BackColor | LightYellow |
| Size | 256; 110 |
| BorderStyle | Fixed3D |

Событийно-управляемые процедуры в форме Регистрация представлены в таблице 12

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

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| class Регистрация | public partial class Регистрация : Form  {  public Регистрация()  {  InitializeComponent();  }  } |
| Кнопка\_подтверждения\_Click | private void Кнопка\_подтверждения\_Click(object sender, EventArgs e) //Переходит на форму 'Авторизация'  {  if ((Логин\_textBox5.Text == "") || (Пароль\_textBox4.Text == "") || (Почта\_textBox2.Text == ""))  {  MessageBox.Show("Не все данные заполнены!", "Ошибка");  }  else  {  using (SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Швидко\_ИА; Integrated Security = True"))  {  SqlDataAdapter info = new SqlDataAdapter($"INSERT INTO Пользователи (Логин,Пароль,Почта) VALUES('{Логин\_textBox5.Text}','{Пароль\_textBox4.Text}','{Почта\_textBox2.Text}');", sqlConnect);  DataTable tt = new DataTable();  info.Fill(tt);  }  MessageBox.Show("Вы зарегестрированы как 'Покупатель'", "Уведомление");  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  }  } |
| Пароль\_KeyPress | private void Пароль\_KeyPress(object sender, KeyPressEventArgs e) //Проверяет на отсутствие букв  {  if (!char.IsDigit(e.KeyChar) && (e.KeyChar != (char)Keys.Back))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать буквы", "Ошибка");  }  } |
| Вернуться\_Click | private void Вернуться\_Click(object sender, EventArgs e) //Переходит на форму 'Авторизация'  {  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |
| выйти\_Click | private void выйти\_Click(object sender, EventArgs e) //Прекращает работу программы  {  Application.Exit();  } |
| Регистрация\_MouseMove | private void Регистрация\_MouseMove(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Регистрация\_MouseDown | private void Регистрация\_MouseDown(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  Mouse = new Point(e.X, e.Y);  } |

Свойства элементов формы Администратор представлены в таблице 13.

Таблица 13 – Свойства элементов формы Администратор

| Объект | Свойство | Значение |
| --- | --- | --- |
| Администратор | FormBorderStyle | None |
| Text | Администратор |
| StartPosition | CenterScreen |
| BackColor | DarkSeaGreen |
| Size | 819; 346 |
| вернуться\_ToolStripMenuItem2 | BackColor | LightYellow |
| Text | Вернуться |
| ForeColor | Red |
| выйти\_PictureBox1 | Size | 25; 25 |
| SizeMode | StretchImage |
| Меню\_ToolStripMenuItem1 | BackColor | LightYellow |
| Text | Меню |
| ForeColor | ControlText |
| Найти\_фото\_roundBtn2 | Size | 113; 34 |
| Text | Добавить/Изменить фото |
| Font | Microsoft Sans Serif; 8,25pt |
| BackColor | LightYellow |
| обновить\_ToolStripMenuItem3 | BackColor | LightYellow |
| Text | Обновить |
| ForeColor | Blue |
| Поиск\_Label1 | BackColor | LightYellow |
| Size | 39; 13 |
| BorderStyle | None |
| Поле\_для\_таблиц\_DataGridView1 | Size | 598; 301 |
| BackColor | Control |
| Строка\_поиска\_TextBox1 | Font | Microsoft Sans Serif; 8,25pt |
| Size | 163; 20 |
| Таблица\_Новые\_сорта\_растений\_ToolStripMenuItem10 | BackColor | LightYellow |
| Text | Новые сорта |
| ForeColor | ControlText |
| Таблица\_Оптовые\_покупатели\_ToolStripMenuItem7 | BackColor | LightYellow |
| Text | Оптовые покупатели |
| ForeColor | ControlText |
| Таблица\_Покупатели\_ToolStripMenuItem6 | BackColor | LightYellow |
| Text | Покупатели |
| ForeColor | ControlText |
| Таблица\_Пользователи\_ToolStripMenuItem12 | BackColor | LightYellow |
| Text | Пользователи |
| ForeColor | ControlText |
| Таблица\_Продавцы\_ToolStripMenuItem8 | BackColor | LightYellow |
| Text | Продавцы |
| ForeColor | ControlText |
| Таблица\_Продажи\_ToolStripMenuItem5 | BackColor | LightYellow |
| Text | Продажи |
| ForeColor | ControlText |
| Таблица\_Сорта\_растений\_ToolStripMenuItem9 | BackColor | LightYellow |
| Text | Сорта растений |
| ForeColor | ControlText |
| Таблица\_Упаковки\_семян\_ToolStripMenuItem11 | BackColor | LightYellow |
| Text | Упаковки семян |
| ForeColor | ControlText |
| Таблицы\_ToolStripMenuItem4 | BackColor | LightYellow |
| Text | Таблицы |
| ForeColor | ControlText |
| Фотография\_представителя\_PictureBox2 | Size | 191; 190 |
| SizeMode | StretchImage |

Событийно-управляемые процедуры в форме Администратор представлены в таблице 14.

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

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| class Администратор | public partial class Администратор : Form  {  public Администратор()  {  InitializeComponent();  Фотография\_представителя\_PictureBox2.Visible = false;  Найти\_фото\_roundBtn2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  }  public void tab(int i) //Добавляет таблицы  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Швидко\_ИА; Integrated Security = True");  sqlConnect.Open();  switch (i)  {  case 0: Поле\_для\_таблиц\_DataGridView1.DataSource = продажиBindingSource; break;  case 1: Поле\_для\_таблиц\_DataGridView1.DataSource = покупателиBindingSource; break;  case 2: Поле\_для\_таблиц\_DataGridView1.DataSource = оптовыепокупателиBindingSource; break;  case 3: Поле\_для\_таблиц\_DataGridView1.DataSource = продавцыBindingSource; break;  case 4: Поле\_для\_таблиц\_DataGridView1.DataSource = сортарастенийBindingSource; break;  case 5: Поле\_для\_таблиц\_DataGridView1.DataSource = новыесортарастенийBindingSource; break;  case 6: Поле\_для\_таблиц\_DataGridView1.DataSource = упаковкисемянBindingSource; break;  case 7: Поле\_для\_таблиц\_DataGridView1.DataSource = пользователиBindingSource; break;  }  } |
| вернуться\_Click | private void вернуться\_Click(object sender, EventArgs e) //Переходит на форму 'Авторизация'  {  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |
| обновить\_Click | private void обновить\_Click(object sender, EventArgs e) //Обновляет и сохраняет изменениям в таблицах  {  switch (c)  {  case 0: продажиTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Продажи);  продажиTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Продажи); break;  case 1: покупателиTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Покупатели);  покупателиTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Покупатели); break;  case 2: оптовые\_покупателиTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Оптовые\_покупатели);  оптовые\_покупателиTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Оптовые\_покупатели); break;  case 3: продавцыTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Продавцы);  продавцыTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Продавцы); break;  case 4: сорта\_растенийTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Сорта\_растений);  сорта\_растенийTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Сорта\_растений); break;  case 5: новые\_сорта\_растенийTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Новые\_сорта\_растений);  новые\_сорта\_растенийTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Новые\_сорта\_растений); break;  case 6: упаковки\_семянTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Упаковки\_семян);  упаковки\_семянTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Упаковки\_семян); break;  case 7: пользователиTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Пользователи);  пользователиTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Пользователи); break;  }  } |
| выйти\_Click | private void выйти\_Click(object sender, EventArgs e) //Прекращает работу программы  {  Application.Exit();  } |
| Таблица\_Продажи\_Click | private void Таблица\_Продажи\_Click(object sender, EventArgs e) //Открывает таблицу 'Продажи'  {  tab(0);  c = 0;  Фотография\_представителя\_PictureBox2.Visible = false;  Найти\_фото\_roundBtn2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Покупатели\_Click | private void Таблица\_Покупатели\_Click(object sender, EventArgs e) //Открывает таблицу 'Покупатели'  {  tab(1);  c = 1;  Фотография\_представителя\_PictureBox2.Visible = false;  Найти\_фото\_roundBtn2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Оптовые\_покупатели\_Click | private void Таблица\_Оптовые\_покупатели\_Click(object sender, EventArgs e) //Открывает таблицу 'Оптовые покупатели'  {  tab(2);  c = 2;  Фотография\_представителя\_PictureBox2.Visible = false;  Найти\_фото\_roundBtn2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Продавцы\_Click | private void Таблица\_Продавцы\_Click(object sender, EventArgs e) //Открывает таблицу 'Продавцы'  {  tab(3);  c = 3;  Фотография\_представителя\_PictureBox2.Visible = false;  Найти\_фото\_roundBtn2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Сорта\_растений\_Click | private void Таблица\_Сорта\_растений\_Click(object sender, EventArgs e) //Открывает таблицу 'Сорта растени'  {  tab(4);  c = 4;  Фотография\_представителя\_PictureBox2.Visible = true;  Найти\_фото\_roundBtn2.Visible = true;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(598, 301);  } |
| Таблица\_Новые\_сорта\_растений\_Click | private void Таблица\_Новые\_сорта\_растений\_Click(object sender, EventArgs e) //Открывает таблицу 'Новые сорта растений'  {  tab(5);  c = 5;  Фотография\_представителя\_PictureBox2.Visible = false;  Найти\_фото\_roundBtn2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Упаковки\_семян\_Click | private void Таблица\_Упаковки\_семян\_Click(object sender, EventArgs e) //Открывает таблицу 'Упаковки семян'  {  tab(6);  c = 6;  Фотография\_представителя\_PictureBox2.Visible = false;  Найти\_фото\_roundBtn2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Пользователи\_Click | private void Таблица\_Пользователи\_Click(object sender, EventArgs e) //Открывает таблицу 'Пользователи'  {  tab(7);  c = 7;  Фотография\_представителя\_PictureBox2.Visible = false;  Найти\_фото\_roundBtn2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Поисковая\_строка\_TextChanged | private void Поисковая\_строка\_TextChanged(object sender, EventArgs e) //Выделение строки, которая подходит под введённые данные в поисковую строку  {  for (int i = 0; i < Поле\_для\_таблиц\_DataGridView1.RowCount; i++)  {  Поле\_для\_таблиц\_DataGridView1.Rows[i].Selected = false;  for (int j = 0; j < Поле\_для\_таблиц\_DataGridView1.ColumnCount; j++)  if (Поле\_для\_таблиц\_DataGridView1.Rows[i].Cells[j].Value != null)  if (Поле\_для\_таблиц\_DataGridView1.Rows[i].Cells[j].Value.ToString().Contains(Строка\_поиска\_TextBox1.Text))  {  Поле\_для\_таблиц\_DataGridView1.Rows[i].Selected = true;  break;  }  }  } |
| Найти\_фото\_Click | private void Найти\_фото\_Click(object sender, EventArgs e) //Открытие окна, для поиска и добавление фотографий  {  SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Швидко\_ИА; Integrated Security = True");  sqlConnect.Open();  OpenFileDialog load = new OpenFileDialog();  if (load.ShowDialog() == DialogResult.OK)  {  Фотография\_представителя\_PictureBox2.ImageLocation = load.FileName;  SqlDataAdapter a = new SqlDataAdapter($"Update Сорта\_растений set Фотография = '{load.FileName}' where Код\_сорта\_растения = '{Поле\_для\_таблиц\_DataGridView1.Rows[Поле\_для\_таблиц\_DataGridView1.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);  DataSet dt = new DataSet();  a.Fill(dt);  }  } |
| Поле\_для\_таблиц\_DataGridView1\_CellClick | private void Поле\_для\_таблиц\_DataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) //Изменение фотографий при нажатие на разные строки  {  try  {  pic = Поле\_для\_таблиц\_DataGridView1.Rows[Поле\_для\_таблиц\_DataGridView1.SelectedCells[0].RowIndex].Cells[7].Value.ToString();  Фотография\_представителя\_PictureBox2.Image = Image.FromFile(pic);  }  catch { }  } |
| Администратор\_MouseMove | private void Администратор\_MouseMove(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Администратор\_MouseDown | private void Администратор\_MouseDown(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  Mouse = new Point(e.X, e.Y);  } |
| Администратор\_Load | private void Администратор\_Load(object sender, EventArgs e) //Загрузка данных в таблицы  {  this.упаковки\_семянTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Упаковки\_семян);  this.продажиTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Продажи);  this.продавцыTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Продавцы);  this.пользователиTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Пользователи);  this.покупателиTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Покупатели);  this.оптовые\_покупателиTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Оптовые\_покупатели);  this.новые\_сорта\_растенийTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Новые\_сорта\_растений);  this.сорта\_растенийTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Сорта\_растений);  } |

Свойства элементов формы Продавец представлены в таблице 15.

Таблица 15 – Свойства элементов формы Продавец

| Объект | Свойство | Значение |
| --- | --- | --- |
| Продавец | FormBorderStyle | None |
| Text | Продавец |
| StartPosition | CenterScreen |
| Size | 819; 348 |
| ShowIcon | False |
| вернуться\_ToolStripMenuItem4 | BackColor | LightYellow |
| Text | Вернуться |
| ForeColor | Red |
| выйти\_PictureBox1 | Size | 25; 25 |
| SizeMode | StretchImage |
| Меню\_ToolStripMenuItem1 | BackColor | LightYellow |
| Text | Меню |
| ForeColor | ControlText |
| обновить\_ToolStripMenuItem3 | BackColor | LightYellow |
| Text | Обновить |
| ForeColor | Blue |
| Поиск\_Label1 | BackColor | LightYellow |
| Size | 39; 13 |
| BorderStyle | None |
| Поле\_для\_таблиц\_DataGridView1 | Size | 598; 301 |
| BackColor | Control |
| Строка\_поиска\_TextBox1 | Font | Microsoft Sans Serif; 8,25pt |
| Size | 163; 20 |
| Таблица\_Новые\_сорта\_растений\_ToolStripMenuItem10 | BackColor | LightYellow |
| Text | Новые сорта |
| ForeColor | ControlText |
| Таблица\_Оптовые\_покупатели\_ToolStripMenuItem7 | BackColor | LightYellow |
| Text | Оптовые покупатели |
| ForeColor | ControlText |
| Таблица\_Покупатели\_ToolStripMenuItem6 | BackColor | LightYellow |
| Text | Покупатели |
| ForeColor | ControlText |
| Таблица\_Продавцы\_ToolStripMenuItem8 | BackColor | LightYellow |
| Text | Продавцы |
| ForeColor | ControlText |
| Таблица\_Продажи\_ToolStripMenuItem5 | BackColor | LightYellow |
| Text | Продажи |
| ForeColor | ControlText |
| Таблица\_Сорта\_растений\_ToolStripMenuItem9 | BackColor | LightYellow |
| Text | Сорта растений |
| ForeColor | ControlText |
| Таблица\_Упаковки\_семян\_ToolStripMenuItem11 | BackColor | LightYellow |
| Text | Упаковки семян |
| ForeColor | ControlText |
| Таблицы\_ToolStripMenuItem2 | BackColor | LightYellow |
| Text | Таблицы |
| ForeColor | ControlText |
| Фотография\_представителя\_PictureBox2 | Size | 191; 190 |
| SizeMode | StretchImage |

Событийно-управляемые процедуры в форме Продавец представлены в таблице 16.

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

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| class Продавец | public partial class Продавец : Form  {  public Продавец()  {  InitializeComponent();  Фотография\_представителя\_PictureBox2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  }  public void Tub(int i) //Добавляет таблицы  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Швидко\_ИА; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter[] so = new SqlDataAdapter[7];  so[0] = new SqlDataAdapter("select \* from Продажи", sqlConnect);  so[1] = new SqlDataAdapter("select \* from Покупатели", sqlConnect);  so[2] = new SqlDataAdapter("select \* from Оптовые\_покупатели", sqlConnect);  so[3] = new SqlDataAdapter("select \* from Продавцы", sqlConnect);  so[4] = new SqlDataAdapter("select \* from Сорта\_растений", sqlConnect);  so[5] = new SqlDataAdapter("select \* from Новые\_сорта\_растений", sqlConnect);  so[6] = new SqlDataAdapter("select \* from Упаковки\_семян", sqlConnect);  DataSet ps = new DataSet();  so[i].Fill(ps);  Поле\_для\_таблиц\_DataGridView1.DataSource = ps.Tables[0];  }  } |
| Продавец\_Load | private void Продавец\_Load(object sender, EventArgs e) //Загрузка данных в таблицы  {  this.упаковки\_семянTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Упаковки\_семян);  this.сорта\_растенийTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Сорта\_растений);  this.продажиTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Продажи);  this.продавцыTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Продавцы);  this.покупателиTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Покупатели);  this.оптовые\_покупателиTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Оптовые\_покупатели);  this.новые\_сорта\_растенийTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Новые\_сорта\_растений);  } |
| обновить\_Click | private void обновить\_Click(object sender, EventArgs e) //Обновляет и сохраняет изменениям в таблицах  {  switch (t)  {  case 0:  продажиTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Продажи);  продажиTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Продажи); break;  case 1:  покупателиTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Покупатели);  покупателиTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Покупатели); break;  case 2:  оптовые\_покупателиTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Оптовые\_покупатели);  оптовые\_покупателиTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Оптовые\_покупатели); break;  case 3:  продавцыTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Продавцы);  продавцыTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Продавцы); break;  case 4:  сорта\_растенийTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Сорта\_растений);  сорта\_растенийTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Сорта\_растений); break;  case 5:  новые\_сорта\_растенийTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Новые\_сорта\_растений);  новые\_сорта\_растенийTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Новые\_сорта\_растений); break;  case 6:  упаковки\_семянTableAdapter.Update(уП01\_ИСПП5\_Швидко\_ИАDataSet.Упаковки\_семян);  упаковки\_семянTableAdapter.Fill(уП01\_ИСПП5\_Швидко\_ИАDataSet.Упаковки\_семян); break;  }  } |
| вернуться\_Click | private void вернуться\_Click(object sender, EventArgs e) //Переходит на форму 'Авторизация'  {  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |
| Таблица\_Продажи\_Click | private void Таблица\_Продажи\_Click(object sender, EventArgs e) //Открывает таблицу 'Продажи'  {  Tub(0);  t = 0;  Поле\_для\_таблиц\_DataGridView1.ReadOnly = false;  Фотография\_представителя\_PictureBox2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Покупатели\_Click | private void Таблица\_Покупатели\_Click(object sender, EventArgs e) //Открывает таблицу 'Покупатели'  {  Tub(1);  t = 1;  Поле\_для\_таблиц\_DataGridView1.ReadOnly = false;  Фотография\_представителя\_PictureBox2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Оптовые\_покупатели\_Click | private void Таблица\_Оптовые\_покупатели\_Click(object sender, EventArgs e) //Открывает таблицу 'Оптовые покупатели'  {  Tub(2);  t = 2;  Поле\_для\_таблиц\_DataGridView1.ReadOnly = false;  Фотография\_представителя\_PictureBox2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Продавцы\_Click | private void Таблица\_Продавцы\_Click(object sender, EventArgs e) //Открывает таблицу 'Продавцы'  {  Tub(3);  t = 3;  Поле\_для\_таблиц\_DataGridView1.ReadOnly = true;  Фотография\_представителя\_PictureBox2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Сорта\_растений\_Click | private void Таблица\_Сорта\_растений\_Click(object sender, EventArgs e) //Открывает таблицу 'Сорта растени'  {  Tub(4);  t = 4;  Поле\_для\_таблиц\_DataGridView1.ReadOnly = true;  Фотография\_представителя\_PictureBox2.Visible = true;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(598, 301);  } |
| Таблица\_Новые\_сорта\_растений\_Click | private void Таблица\_Новые\_сорта\_растений\_Click(object sender, EventArgs e) //Открывает таблицу 'Новые сорта растений'  {  Tub(5);  t = 5;  Поле\_для\_таблиц\_DataGridView1.ReadOnly = true;  Фотография\_представителя\_PictureBox2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Таблица\_Упаковки\_семян\_Click | private void Таблица\_Упаковки\_семян\_Click(object sender, EventArgs e) //Открывает таблицу 'Упаковки семян'  {  Tub(6);  t = 6;  Поле\_для\_таблиц\_DataGridView1.ReadOnly = true;  Фотография\_представителя\_PictureBox2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  } |
| Поисковая\_строка\_TextChanged | private void Поисковая\_строка\_TextChanged(object sender, EventArgs e) //Выделение строки, которая подходит под введённые данные в поисковую строку  {  for (int i = 0; i < Поле\_для\_таблиц\_DataGridView1.RowCount; i++)  {  Поле\_для\_таблиц\_DataGridView1.Rows[i].Selected = false;  for (int j = 0; j < Поле\_для\_таблиц\_DataGridView1.ColumnCount; j++)  if (Поле\_для\_таблиц\_DataGridView1.Rows[i].Cells[j].Value != null)  if (Поле\_для\_таблиц\_DataGridView1.Rows[i].Cells[j].Value.ToString().Contains(Строка\_поиска\_TextBox1.Text))  {  Поле\_для\_таблиц\_DataGridView1.Rows[i].Selected = true;  break;  }  }  } |
| Продавец\_MouseMove | private void Продавец\_MouseMove(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Продавец\_MouseDown | private void Продавец\_MouseDown(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  Mouse = new Point(e.X, e.Y);  } |
| выйти\_Click | private void выйти\_Click(object sender, EventArgs e) //Прекращает работу программы  {  Application.Exit();  } |
| Поле\_для\_таблиц\_DataGridView1\_CellClick | private void Поле\_для\_таблиц\_DataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) //Изменение фотографий при нажатие на разные строки  {  try  {  pic = Поле\_для\_таблиц\_DataGridView1.Rows[Поле\_для\_таблиц\_DataGridView1.SelectedCells[0].RowIndex].Cells[7].Value.ToString();  Фотография\_представителя\_PictureBox2.Image = Image.FromFile(pic);  }  catch { }  } |

Свойства элементов формы Покупатель представлены в таблице 17.

Таблица 17 – Свойства элементов формы Покупатель

| Объект | Свойство | Значение |
| --- | --- | --- |
| Покупатель | FormBorderStyle | None |
| Text | Покупатель |
| StartPosition | CenterScreen |
| Size | 819; 348 |
| ShowIcon | False |
| вернуться\_ToolStripMenuItem2 | BackColor | LightYellow |
| Text | Вернуться |
| ForeColor | Red |
| выйти\_PictureBox1 | Size | 25; 25 |
| SizeMode | StretchImage |
| Меню\_ToolStripMenuItem1 | BackColor | LightYellow |
| Text | Меню |
| ForeColor | ControlText |
| Поиск\_Label1 | BackColor | LightYellow |
| Size | 39; 13 |
| BorderStyle | None |
| Поле\_для\_таблиц\_DataGridView1 | Size | 598; 301 |
| BackColor | Control |
| Строка\_поиска\_TextBox1 | Font | Microsoft Sans Serif; 8,25pt |
| Size | 163; 20 |
| Асортимент\_ToolStripMenuItem3 | BackColor | LightYellow |
| Text | Асортимент |
| ForeColor | ControlText |
| Фотография\_представителя\_PictureBox2 | Size | 191; 190 |
| SizeMode | StretchImage |

Событийно-управляемые процедуры в форме Покупатель представлены в таблице 18.

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

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| class Покупатель | public partial class Покупатель : Form  {  public Покупатель()  {  InitializeComponent();  Фотография\_представителя\_PictureBox2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  }  public void Zap(int v) //Добавляет таблицы  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Швидко\_ИА; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter[] sos = new SqlDataAdapter[1];  sos[0] = new SqlDataAdapter("select Название\_растения, Название\_сорта, Год\_выведения, Адаптация, Морозоустойчивость, Способ\_посадки, Фотография, Срок\_созревания\_в\_днях, Примечание, Наличие\_в\_текущее\_время, Срок\_годности, Одобрено\_инспекцией, Вид\_упаковки, Количество\_в\_упаковке, Цена, Вес\_семян\_в\_граммах from Сорта\_растений join Упаковки\_семян on Сорта\_растений.Код\_сорта\_растения = Упаковки\_семян.Код\_сорта\_растения", sqlConnect);  DataSet pk = new DataSet();  sos[v].Fill(pk);  Поле\_для\_таблиц\_DataGridView1.DataSource = pk.Tables[0];  } |
| вернуться\_Click | private void вернуться\_Click(object sender, EventArgs e) //Переходит на форму 'Авторизация'  {  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |
| Асортимент\_ToolStripMenuItem3\_Click | private void Асортимент\_ToolStripMenuItem3\_Click(object sender, EventArgs e) //Открывает таблицы 'Сорта растени' и 'Упаковки семян'  {  Zap(0);  Фотография\_представителя\_PictureBox2.Visible = true;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(598, 301);  } |
| Покупатель\_Load | private void Покупатель\_Load(object sender, EventArgs e) //Загрузка данных в таблицы  {  this.упаковки\_семянTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Упаковки\_семян);  this.сорта\_растенийTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Сорта\_растений);  } |
| выйти\_Click | private void выйти\_Click(object sender, EventArgs e) //Прекращает работу программы  {  Application.Exit();  } |
| Поисковая\_строка\_TextChanged | private void Поисковая\_строка\_TextChanged(object sender, EventArgs e) //Выделение строки, которая подходит под введённые данные в поисковую строку  {  for (int i = 0; i < Поле\_для\_таблиц\_DataGridView1.RowCount; i++)  {  Поле\_для\_таблиц\_DataGridView1.Rows[i].Selected = false;  for (int j = 0; j < Поле\_для\_таблиц\_DataGridView1.ColumnCount; j++)  if (Поле\_для\_таблиц\_DataGridView1.Rows[i].Cells[j].Value != null)  if (Поле\_для\_таблиц\_DataGridView1.Rows[i].Cells[j].Value.ToString().Contains(Строка\_поиска\_TextBox1.Text))  {  Поле\_для\_таблиц\_DataGridView1.Rows[i].Selected = true;  break;  }  }  } |
| Покупатель\_MouseMove | private void Покупатель\_MouseMove(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Покупатель\_MouseDown | private void Покупатель\_MouseDown(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  Mouse = new Point(e.X, e.Y);  } |
| Поле\_для\_таблиц\_DataGridView1\_CellClick | private void Покупатель\_MouseDown(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  Mouse = new Point(e.X, e.Y);  }  string pic;  private void Поле\_для\_таблиц\_DataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) //Изменение фотографий при нажатие на разные строки  {  try  {  pic = Поле\_для\_таблиц\_DataGridView1.Rows[Поле\_для\_таблиц\_DataGridView1.SelectedCells[0].RowIndex].Cells[6].Value.ToString();  Фотография\_представителя\_PictureBox2.Image = Image.FromFile(pic);  }  catch { }  } |

Свойства элементов формы Гость представлены в таблице 19.

Таблица 19 – Свойства элементов формы Гость

| Объект | Свойство | Значение |
| --- | --- | --- |
| Гость | FormBorderStyle | None |
| Text | Гость |
| StartPosition | CenterScreen |
| Size | 819; 348 |
| ShowIcon | False |
| вернуться\_ToolStripMenuItem2 | BackColor | LightYellow |
| Text | Вернуться |
| ForeColor | Red |
| выйти\_PictureBox1 | Size | 25; 25 |
| SizeMode | StretchImage |
| Меню\_ToolStripMenuItem1 | BackColor | LightYellow |
| Text | Меню |
| ForeColor | ControlText |
| Поиск\_Label1 | BackColor | LightYellow |
| Size | 39; 13 |
| BorderStyle | None |
| Поле\_для\_таблиц\_DataGridView1 | Size | 598; 301 |
| BackColor | Control |
| Строка\_поиска\_TextBox1 | Font | Microsoft Sans Serif; 8,25pt |
| Size | 163; 20 |
| Асортимент\_ToolStripMenuItem3 | BackColor | LightYellow |
| Text | Асортимент |
| ForeColor | ControlText |
| Фотография\_представителя\_PictureBox2 | Size | 191; 190 |
| SizeMode | StretchImage |

Событийно-управляемые процедуры в форме Гость представлены в таблице 20.

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

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| class Гость | public partial class Гость : Form  {  public Гость()  {  InitializeComponent();  Фотография\_представителя\_PictureBox2.Visible = false;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(794, 301);  }  public void Zap(int v) //Добавляет таблицы  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Швидко\_ИА; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter[] sos = new SqlDataAdapter[1];  sos[0] = new SqlDataAdapter("select Название\_растения, Название\_сорта, Год\_выведения, Адаптация, Морозоустойчивость, Способ\_посадки, Фотография, Срок\_созревания\_в\_днях, Примечание, Наличие\_в\_текущее\_время, Срок\_годности, Одобрено\_инспекцией, Вид\_упаковки, Количество\_в\_упаковке, Цена, Вес\_семян\_в\_граммах from Сорта\_растений join Упаковки\_семян on Сорта\_растений.Код\_сорта\_растения = Упаковки\_семян.Код\_сорта\_растения", sqlConnect);  DataSet pk = new DataSet();  sos[v].Fill(pk);  Поле\_для\_таблиц\_DataGridView1.DataSource = pk.Tables[0];  } |
| Гость\_Load | private void Гость\_Load(object sender, EventArgs e) //Загрузка данных в таблицы  {  this.упаковки\_семянTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Упаковки\_семян);  this.сорта\_растенийTableAdapter.Fill(this.уП01\_ИСПП5\_Швидко\_ИАDataSet.Сорта\_растений);  } |
| вернуться\_Click | private void вернуться\_Click(object sender, EventArgs e) //Переходит на форму 'Авторизация'  {  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |
| выйти\_Click | private void выйти\_Click(object sender, EventArgs e) //Прекращает работу программы  {  Application.Exit();  } |
| Асортимент\_ToolStripMenuItem3\_Click | private void Асортимент\_ToolStripMenuItem3\_Click(object sender, EventArgs e) //Открывает таблицы 'Сорта растени' и 'Упаковки семян'  {  Zap(0);  Фотография\_представителя\_PictureBox2.Visible = true;  Поле\_для\_таблиц\_DataGridView1.Size = new Size(598, 301);  } |
| Поисковая\_строка\_TextChanged | private void Поисковая\_строка\_TextChanged(object sender, EventArgs e) //Выделение строки, которая подходит под введённые данные в поисковую строку  {  for (int i = 0; i < Поле\_для\_таблиц\_DataGridView1.RowCount; i++)  {  Поле\_для\_таблиц\_DataGridView1.Rows[i].Selected = false;  for (int j = 0; j < Поле\_для\_таблиц\_DataGridView1.ColumnCount; j++)  if (Поле\_для\_таблиц\_DataGridView1.Rows[i].Cells[j].Value != null)  if (Поле\_для\_таблиц\_DataGridView1.Rows[i].Cells[j].Value.ToString().Contains(Строка\_поиска\_TextBox1.Text))  {  Поле\_для\_таблиц\_DataGridView1.Rows[i].Selected = true;  break;  }  }  } |
| Гость\_MouseMove | private void Гость\_MouseMove(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Гость\_MouseDown | private void Гость\_MouseDown(object sender, MouseEventArgs e) //Перемещает форму при удержании мыши  {  Mouse = new Point(e.X, e.Y);  } |
| Поле\_для\_таблиц\_DataGridView1\_CellClick | private void Поле\_для\_таблиц\_DataGridView1\_CellClick\_1(object sender, DataGridViewCellEventArgs e) //Изменение фотографий при нажатие на разные строки  {  try  {  pic = Поле\_для\_таблиц\_DataGridView1.Rows[Поле\_для\_таблиц\_DataGridView1.SelectedCells[0].RowIndex].Cells[6].Value.ToString();  Фотография\_представителя\_PictureBox2.Image = Image.FromFile(pic);  }  catch { }  } |

Вывод по главе: была полностью спроектирована и разработана система базы данных, были полностью спроектированы и разработаны все программные модули информационной системы “ Учёт продаж сортов растений ”.

# 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 сорт растения, который в дальнейшем приобретёт покупатель.
* Действия пользователя: пользователь ищет в таблице упаковку семян с определённым сортом растений и сколько он будет стоить и т п.
* В результате в результате отображена цена упаковки семян 500 руб.
* 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 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.

В таблице 21 представлены сведения о пользователях.

Таблица 21 – Информация о пользователях

| ФИО | Пол | Возраст | Образование |
| --- | --- | --- | --- |
| Попов Андрей Евгеньевич | Мужской | 24 | Среднее специальное |
| Троцкий Юрий Александрович | Мужской | 24 | Среднее специальное |
| Козлов Владислав Владимирович | Мужской | 30 | Среднее специальное |
| Шелковникова Анастасия Анатольевна | Женский | 35 | Среднее специальное |
| Комаров Александр Иванович | Мужской | 34 | Среднее специальное |

## 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.

В приложении “А” на рисунке 21 представлены результаты проведения нагрузочного тестирования.

Исходя из результатов 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 23 – Результаты проведения юзабилити-тестирования

| Вопрос | Попов Андрей Евгеньевич | Троцкий Юрий Александрович | Козлов Владислав Владимирович | Шелковникова Анастасия Анатольевна | Комаров Александр Иванович |
| --- | --- | --- | --- | --- | --- |
| Присутствуют ли такие дизайн-элементы, которые неприятны для зрения? | нет | нет | нет | нет | нет |
| 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 | нет |
| Понятен ли смысл приложения? | да | да | да | да | да |
| Общая оценка | 7/10 | 9/10 | 8/10 | 8/10 | 7/10 |

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.