# РЕАЛИЗАЦИЯ

## Разработка классов информационной системы

Windows Forms — это технология пользовательского интерфейса для .NET, представляющая собой набор управляемых библиотек, которые упрощают выполнение стандартных задач, таких как чтение из файловой системы и запись в нее. С помощью среды разработки, такой как Visual Studio, можно создавать интеллектуальные клиентские приложения Windows Forms, которые отображают информацию, запрашивают ввод пользователя и взаимодействуют с удаленными компьютерами по сети [15].

В Windows Forms форма — это визуальная поверхность, на которой выводится информация для пользователя. Обычно приложение Windows Forms строится путем добавления элементов управления в формы и создания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш. Элемент управления — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.

При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления создается событие. Приложение реагирует на эти события, как задано в коде, и обрабатывает события при их возникновении.

В Windows Forms предусмотрено множество элементов управления, которые можно добавлять в формы. Например, элементы управления могут отображать текстовые поля, кнопки, раскрывающиеся списки, переключатели и даже веб-страницы.

Используя функцию перетаскивания конструктора Windows Forms в Visual Studio, можно легко создавать приложения Windows Forms. Просто выделите элемент управления с помощью курсора и поместите его на нужное место в форме. Для преодоления трудностей, связанных с выравниванием элементов управления, конструктор предоставляет такие средства, как линии сетки и линии привязки. С помощью элементов управления FlowLayoutPanel, TableLayoutPanel и SplitContainer можно гораздо быстрее создавать сложные макеты форм.

Наконец, если нужно создать свои собственные элементы пользовательского интерфейса, пространство имен System.Drawing содержит широкий набор классов, необходимых для отрисовки линий, кругов и других фигур непосредственно на форме.

При создании формы, создается класс, соответствующий этой форме. Наше приложение сделано однооконным, т.е. будет только один класс отображения.

Тем не менее, для отображения и работы с данными в базе данных каждой таблице создаются свои классы адаптеры.

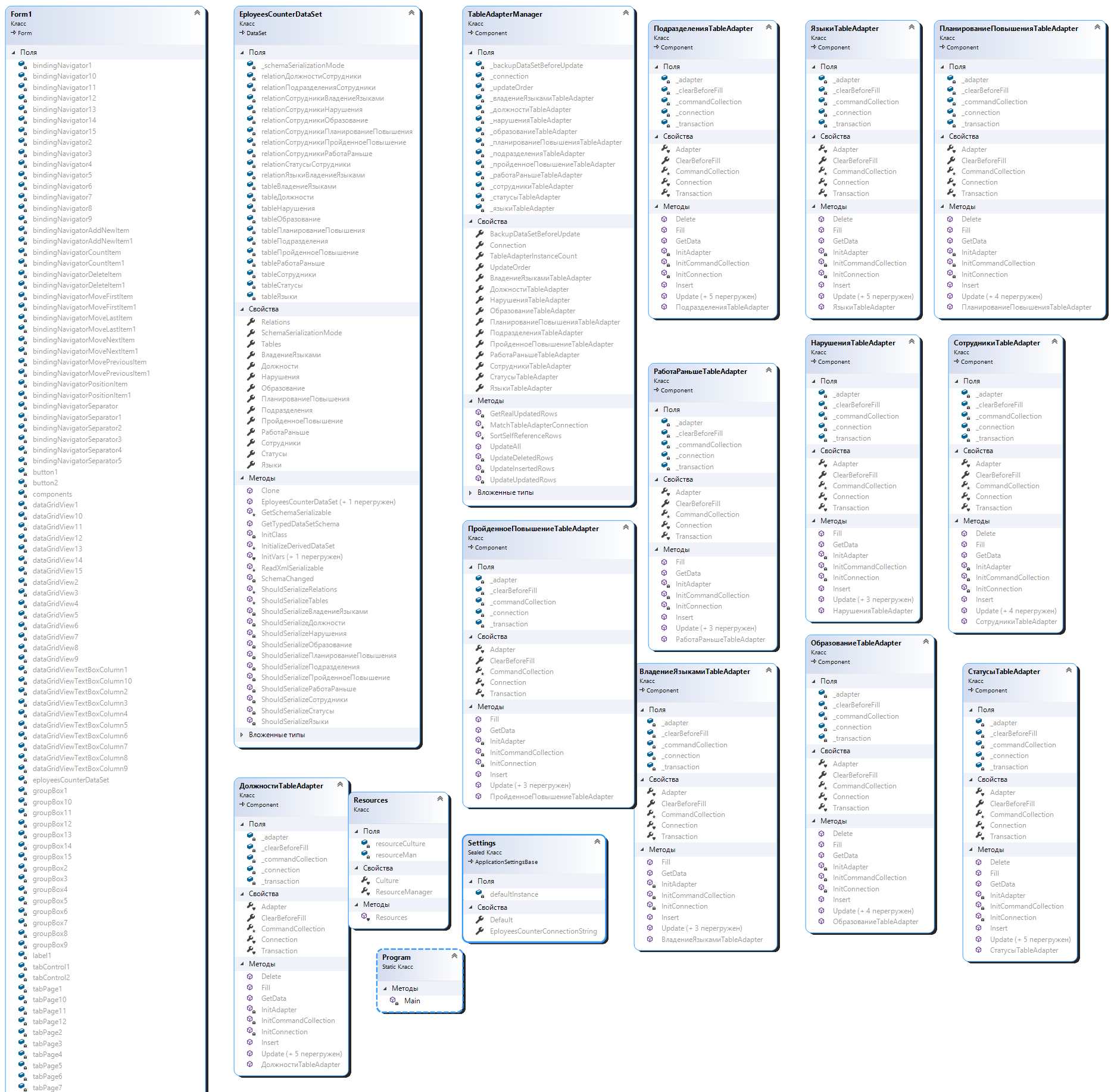


Рисунок 4.1 – Классы приложения

## [Разработка интерфейса программного продукта](#_Toc474749001)

При запуске приложений будет отображаться список всех сотрудников предприятия. На этой же вкладке можно поискать сотрудника по его фио.

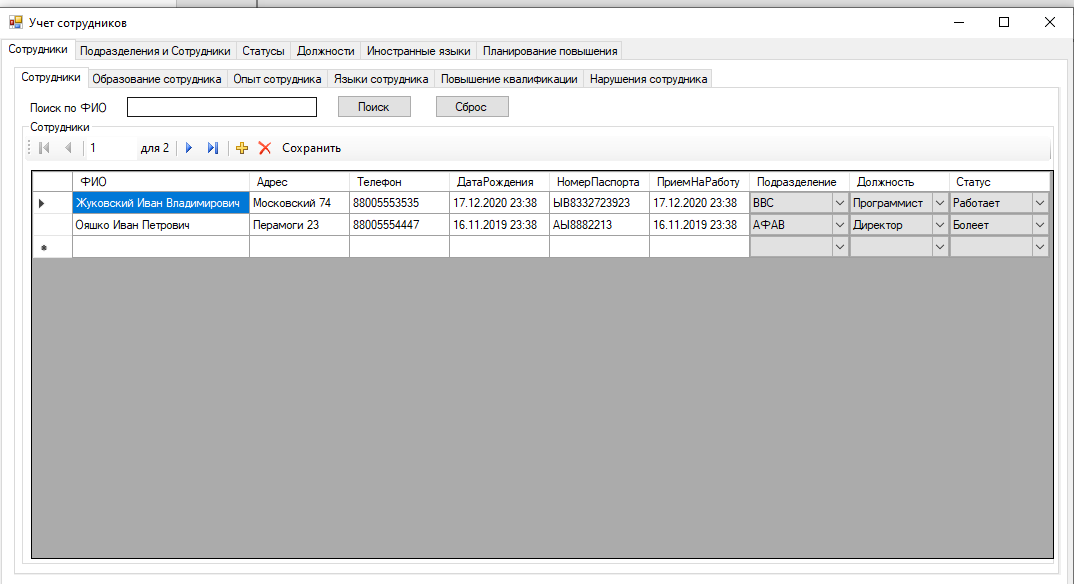


Рисунок 4.2 – Вид приложения при запуске

Все подвкладки вкладки сотрудника определяются выбором сотрудника из списка. Так, если выбрать первого сотрудника, то во всех последующих подвкладках будет отображаться информация именно о выбранном сотруднике. Так, на вкладке образование показывается список мест, где учился выбранный ранее сотрудник.

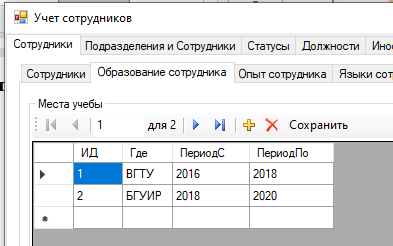


Рисунок 4.3 – Образование выбранного сотрудника

На странице опыта сотрудника отображается список мест, где работал сотрудник до текущего предприятия.

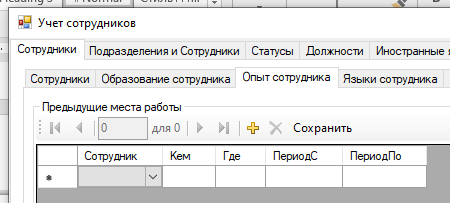


Рисунок 4.4 – Опыт работы выбранного сотрудника

На вкладке языков отображаются все языки, которыми владеет выбранный ранее сотрудник.

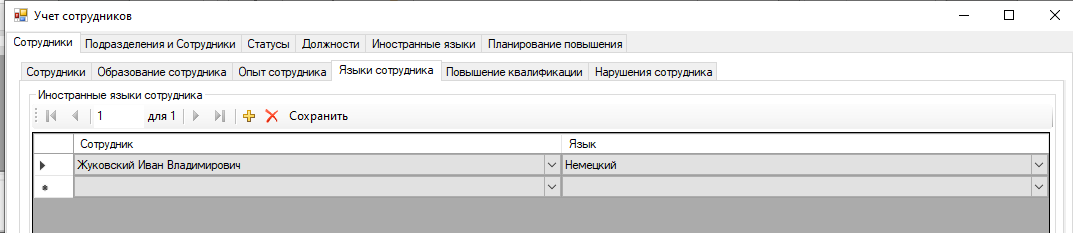


Рисунок 4.5 – Языки, которыми владеет выбранный сотрудник

На вкладке повышения квалификации отображены все повышения данного сотрудника, с указанием что и когда это было.

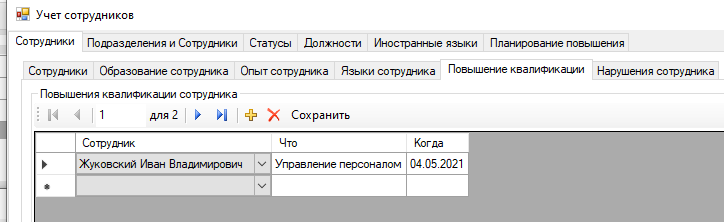


Рисунок 4.6 – Пройденные повышения квалификации выбранного сотрудника

На странице нарушений отображается список всех нарушений сотрудника, если таковые имеются.

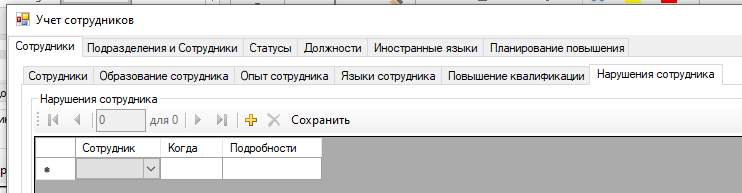


Рисунок 4.7 – Список нарушений выбранного сотрудника

При переходе по верхнему уровню вкладок, можно взаимодействовать с данными всех таблиц напрямую. Кроме того, при выборе элементов в таблице под ними будет отображаться список сотрудников, имеющих отношение к выбранному элементу.

Так, на странице подразделений можно добавлять, изменять и удалять подразделения. Также, при выборе подразделения в таблице ниже будут показаны все сотрудники выбранного подразделения.

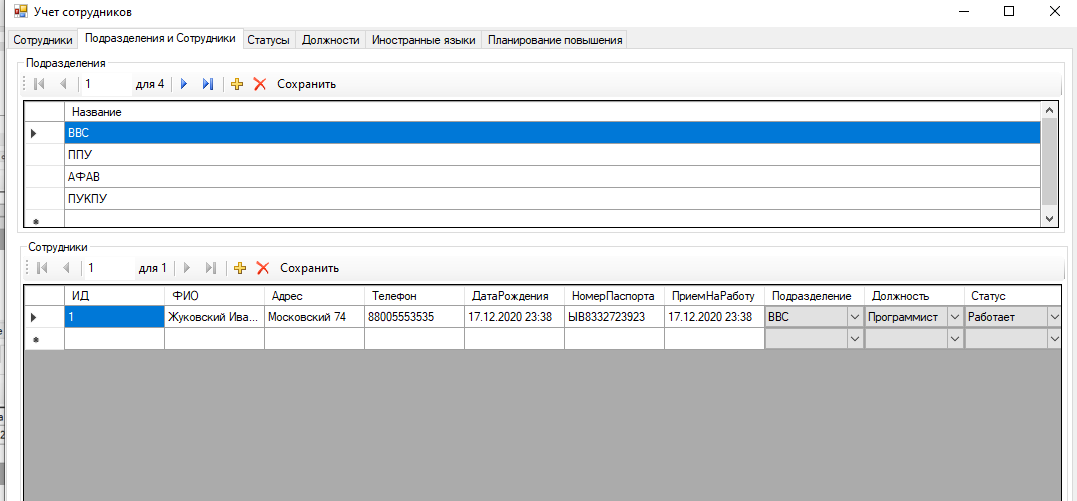


Рисунок 4.8 – Подразделения предприятия

На вкладке статусов происходит то же самое, при выборе статуса в таблице ниже будут отображены все сотрудники с выбранным статусом.

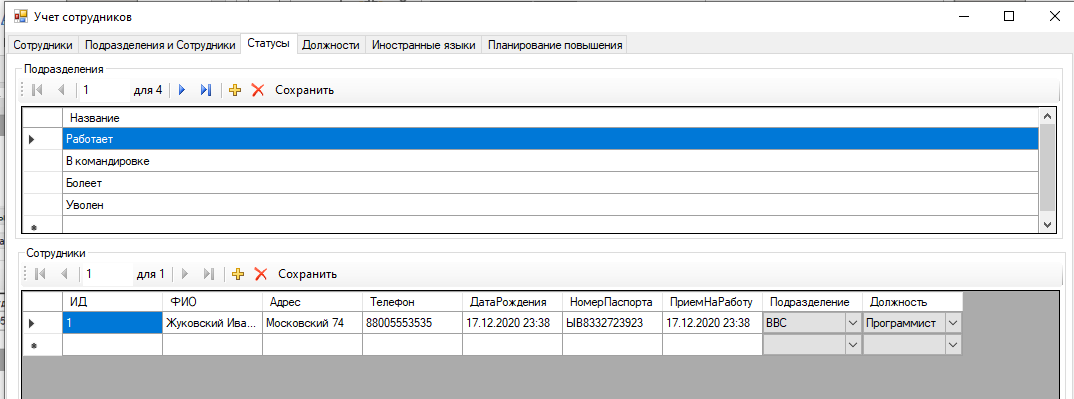


Рисунок 4.9 – Статусы сотрудников

Это поведение касается так же должностей и иностранных языков

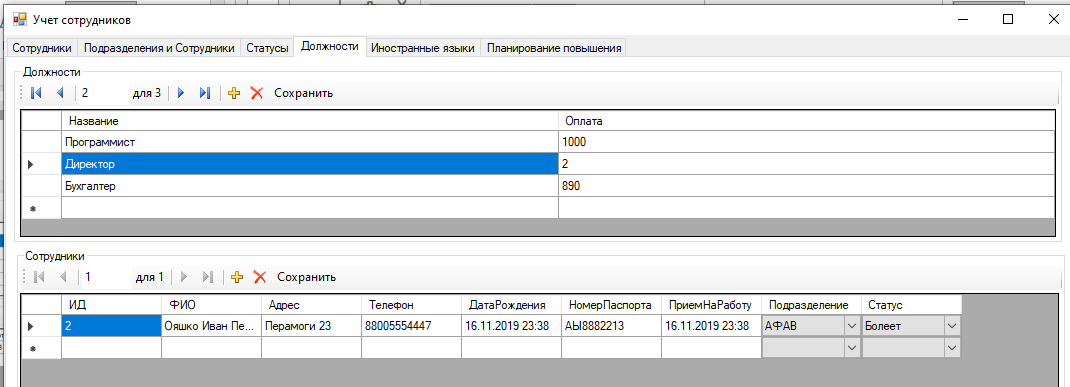


Рисунок 4.10 – Должности сотрудников

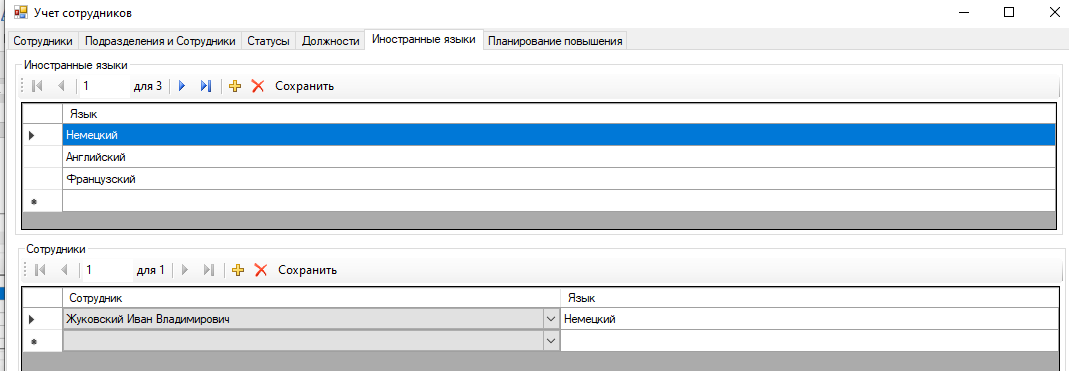


Рисунок 4.11 – Иностранные языки

Вкладка планирования повышения предлагает место для хранения планов на повышение квалификации сотрудников с указанием что, кому и когда планируется повышать.

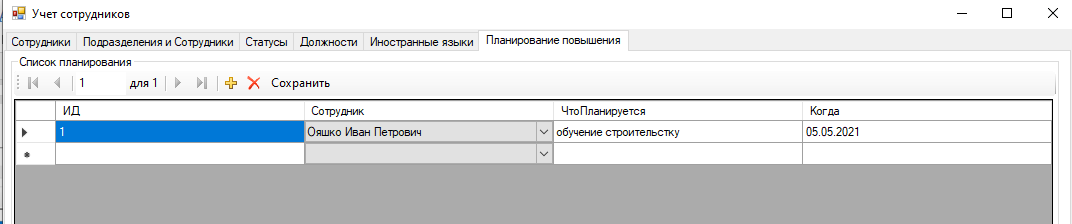


Рисунок 4.12 – Планирование повышений

## [Разработка алгоритмов реализации вариантов использования](#_Toc474749002)

При решении общей задачи, возникали более мелкие, для решения которых необходимо было использовать и разрабатывать алгоритмы. Несколько таких примеров будет показано ниже.

1. Сохранение информации в базу данных посредством BindingNavigator

Использование встроенного компонента помогает избегать множество проблем, когда работа идет только с одной таблицей. Такой компонент реализует все стандартные функции для отображения и изменения информации, однако не реализует метода сохранения информации. Так, для сохранения данных в таблицах, использующих данные только своей таблицы, используется следующий код.

private void toolStripButton1\_Click(object sender, EventArgs e)

{

try

{

статусыBindingSource.EndEdit();

this.статусыTableAdapter.Update(eployeesCounterDataSet);

MessageBox.Show("Сохранено");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

this.статусыTableAdapter.Fill(eployeesCounterDataSet.Статусы);

}

1. Поиск сотрудника по ФИО

Поскольку при запуске приложения мы сразу получаем список всех сотрудников предприятия, было бы неправильно при каждом поиске сотрудника отправлять запрос в базу данных, уже име на руках всех. Поэтому было решено использовать фильтрацию данных по внутреннему состоянию dataGridView

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

{

сотрудникиBindingSource.RemoveFilter();

}

private void button1\_Click(object sender, EventArgs e)

{

сотрудникиBindingSource.Filter = $"ФИО LIKE '%{textBox1.Text}%'";

}

## Модульное тестирование алгоритмов реализации вариантов использования

Модульное тестирование, или юнит-тестирование - это процесс в программировании, который направлен на проверку отдельных небольших частей приложения, также называемых атомарными, которые можно исследовать изолированно от других подобных частей. При выполнении данного тестирования могут проверяться как отдельные функции или методы классов, так и сами классы, взаимодействие классов, небольшие библиотеки и отдельные части приложения [16]. Довольно часто данный вид тестирования реализуется с использованием специальных технологий и инструментальных средств автоматизации тестирования, значительно упрощающих и ускоряющих разработку соответствующих тест-кейсов. Таким образом, оценивая каждый элемент отдельно и подтверждая правильность его работы, установить проблему значительно проще чем, если бы элемент был частью системы.

Примеры тестирования алгоритмов реализации вариантов использования:

1. Тестирование поиска сотрудника по ФИО.

Таблица 4.1 - Тестирование поиска сотрудника по ФИО

|  |  |  |  |
| --- | --- | --- | --- |
| Краткое описание | Предварительные условия | Шаги для воспроизведения | Ожидаемый результат |
| Поиск сотрудника | Запустить приложение | 1. Ввести часть ФИО сотрудника. 2. Нажать поиск | В таблице сотрудников отобразятся только те сотрудники, в ФИО которых есть введенная часть для поиска |

2. Тестирование добавления иностранного языка.

Таблица 4.2 - Тестирование добавления иностранного языка

|  |  |  |  |
| --- | --- | --- | --- |
| Краткое описание | Предварительные условия | Шаги для воспроизведения | Ожидаемый результат |
| Добавление иностранного языка | Запустить приложение.  Перейти на вкладку иностранных языков | Ввести в таблицу иностранных языков новый язык.  Нажать сохранить.  В таблице под новым языком выбрать первого сотрудника.  Нажать сохранить. | Появится сообщение об успешном сохранении Языка и владения языка. На странице языков выбранного сотрудника будет отображаться новый язык. |

3. Тестирование связи вкладок

Таблица 4.3 - Тестирование связи вкладок

|  |  |  |  |
| --- | --- | --- | --- |
| Краткое описание | Предварительные условия | Шаги для воспроизведения | Ожидаемый результат |
| Связь вкладок | 1. Запустить приложение. | На вкладке сотрудников изменить подразделение, должность и статус первого сотрудника.  Нажать сохранить. | После успешного сохранения на вкладках статуса, подразделения и должностей выбранный сотрудник будет отображаться уже с новым подразделением, должностью и статусом. |