МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет» Факультет автоматики и вычислительной техники Кафедра электронных вычислительных машин

# Отчет

по производственной практике на ПАО «Кировский завод «Маяк», г. Киров

Выполнил студент группы ИВТ-31 /Жеребцов К. А./

Руководитель практики от ВятГУ /Караваева О.В./ Руководитель практики от предприятия /Думбрава А. В./

Киров 2022

График прохождения практики

|  |  |
| --- | --- |
| Дата | Вид деятельности |
| 06.09.2022 | Получение задания на практику |
| 07.09.2022 -19.11.2022 | Выполнение задания |
| 20.11.2022 | Подготовить и оформить отчет |

Введение

Производственная практика является неотъемлемой и важной частью учебного процесса. В ходе неё не только закрепляются теоретические знания, полученные во время обучения, но и отрабатываются необходимые практические умения для эффективной работы в профессиональной сфере, осваивается инструментарий для работы и взаимодействия в команде, оттачиваются коммуникативные навыки для общения с коллегами.

Данный документ представляет собой отчет по производственной практике, проходившей в период с 6 сентября по 27 ноября ПАО «Кировский завод «Маяк», г. Киров.

1. Общие сведения о предприятии

ПАО «Кировский завод «Маяк» был создан 30 июля 1941 года.

Физический адрес: Российская Федерация, Кировская область, город Киров, улица Молодой Гвардии, 67

Публичное акционерное общество (ПАО) — это общество, чьи акции и ценные бумаги, которые конвертируются в акции, публично размещаются или публично обращаются. То есть их можно свободно приобрести на рынке ценных бумаг.

Сегодня это крупное машиностроительное предприятие, специализирующееся на выпуске товаров народного потребления.

2. Задание

В соответствии с заданием от организации во время прохождения практики необходимо разработать программу, собирающую информацию по всем комплектующим, установленным на установки.

2.1 Функциональные требования к программе

Приложение должно обладать следующим функционалом:

1. возможность добавления и удаления установок
2. возможность добавления, удаления и редактирования комплектующих.

Комплектующие различаются по следующим полям:

* Тип комплектующего
* Обозначение
* Наименование
* Серийный номер
* Поставщик
* Изготовитель
* Дата изготовления
* Срок хранения
* Страна изготовления

1. Возможность фильтровать комплектующие, установленные на выбранной установке, по их типу

3. Описание выполненной работы.

3.1 База данных

Для программы была создана таблица базы данных (БД) в MS SQL Server Management Studio.

Для создания таблицы БД, необходимо создать локальный сервер, на котором она будет храниться. После этого можно создать таблицу со структурой, указанной на рисунке 1. Столбец «Установка» определяет к какой установке принадлежит компонент. Все остальные столбцы характеризуют сам компонент.



Рисунок 1 – Структура таблицы

3.2 Интерфейс программы

Интерфейс разработанного приложения представлен на рисунке 2.

3.2.1 Описание работы

1. Для выбора установки необходимо указать ее в выпадающем списке в левом верхнем углу («Открыть комплектующие») и нажать кнопку «ОК». После этого в таблице появятся все компоненты, установленные на установку.
2. Для удаления установки необходима выбрать ее в поле «Удалить установку» и нажать кнопку «Удалить». Далее можно будет подтвердить действие или отменить его.
3. В поле «Выборка комплектующих» можно задать тип комплектующего, и в таблице будут отображаться компоненты того типа, который был выбран.
4. Для удаления комплектующих необходимо выбрать эту строку и нажать кнопку «Удалить». После этого комплектующий будет удален.
5. Для редактирования компонентов необходимо выбрать один в таблице и нажать кнопку «Редактировать». После этого в полях под таблицей будут занесены данные об этом компоненте. Далее можно менять значения. После нажатия на кнопку «Изменить» новые данные запишутся в таблицу.
6. Для добавления нового компонента необходимо в поле «Установка» в нижней части окна выбрать установку и ввести данные о комплектующем, который нужно добавить. После нажатия кнопки «Добавить» он появиться в таблице.
7. Для добавления новой установки необходима в том же поле «Установка», что и в прошлом пункте, ввести название установки. Можно не указывать данные о установленном компоненте, тогда установка добавиться с пустой таблицей. Если же указать, то установка добавиться сразу с одним комплектующим. Добавление происходит после нажатия кнопки «Добавить».

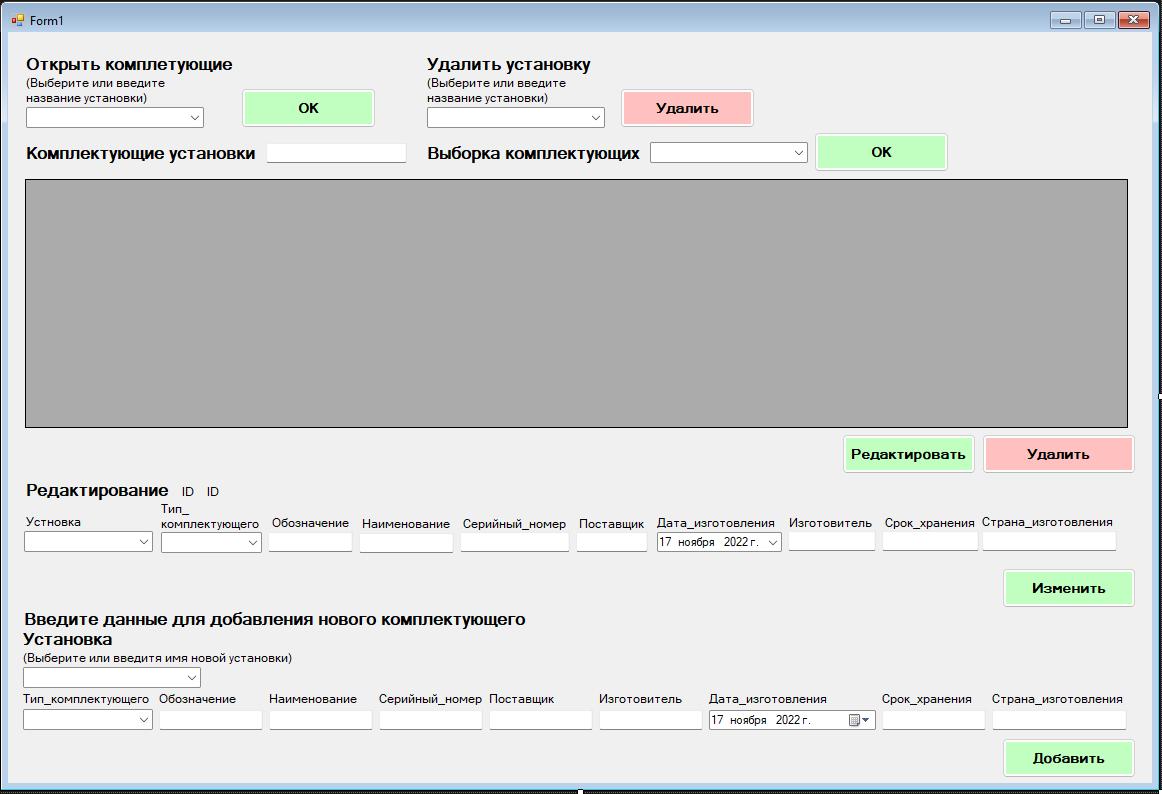


Рисунок 2 – Интерфейс приложения.

3.3 Реализация

Программа разработана на языке C# в Windows Forms в среде разработки Visual Studio.

3.3.1 Подключение к БД

Для того чтобы программа могла взаимодействовать с БД необходимо подключить программу к серверу, на котором храниться БД. В строке подключения указать имя сервера и имя базы данных, как показано на рисунке 3.



Рисунок 3 – Строка подключения.

Тут «DESKTOP-PVL4P4H» - имя сервера, а «Ustanovki» - имя базы данных.

Далее необходимо указать в качестве нового источника данных таблицу БД. При добавлении источника данных нужно указать созданную строку подключения и выбрать нужную таблицу.

После проделанных действий программа сможет читать данные, хранящиеся в БД и изменять их.

3.3.2 Обработка запросов, возвращающих данные

Для обработки запросов необходимо создать подключение, как показано на рисунке 4.

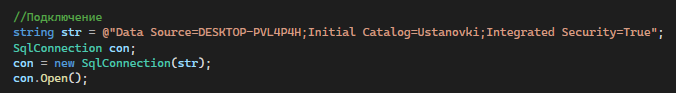


Рисунок 4 – Код подключения.

После этого можно создать запрос, например, отображения всех компонентов выбранной установки, как показано на рисунках 5. Процедура UpdateGrid() обрабатывает запрос и выводит данные в таблицу. Ее реализация представлена на рисунке 6.

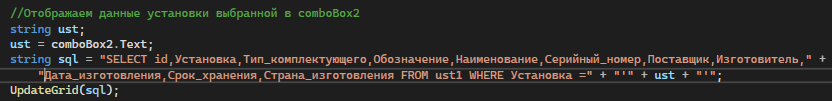


Рисунок 5 – Создание строки запроса

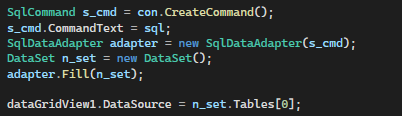


Рисунок 6 – Реализация UdateGrid()

3.3.3 Обработка запросов, не возвращающих данные

К примеру, таким запросом является «Удаление строки». В процессе его выполнения нужно также создается строка запроса и вызывается процедура ExecSQL() (рисунок 7).

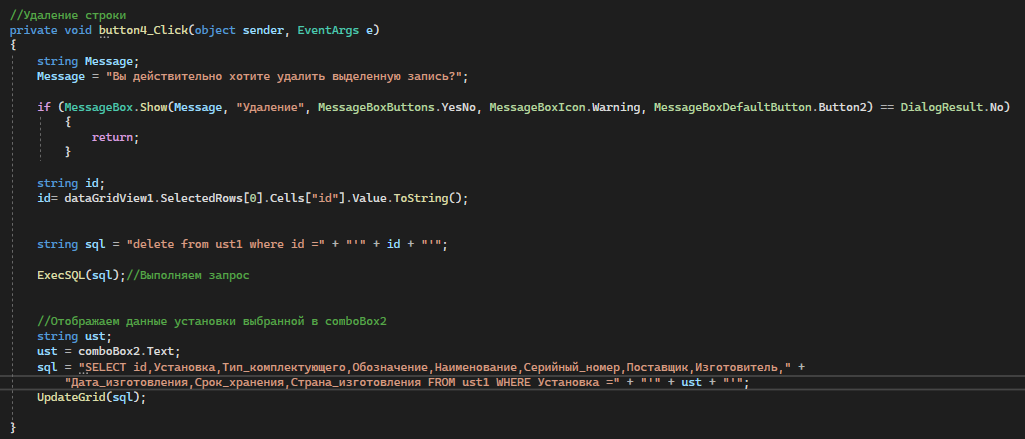


Рисунок 7 – Процедура удаления строки

Реализация процедуры ExecSQL(), выполняющей запрос без возврата данных, представлена на рисунке 8.

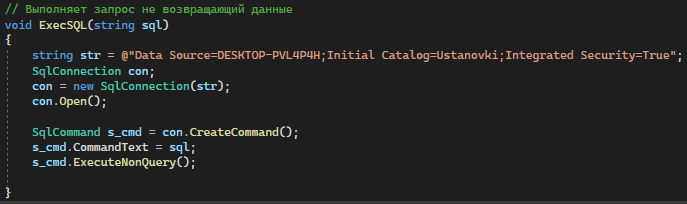


Рисунок 8 – Реализация процедуры ExecSQL()

Заключение

В ходе производственной практики был получен опыт по разработке приложений в Windows Forms в MS Visual Studio, а также в MS SQL Serever Manager. Кроме того, был изучен способ взаимодействия приложения Windows Forms и сервера с таблицей БД. Получено представление об общей структуре приложения. При работе с базой данных, получены навыки в построении простейших запросов БД MSsql.

Приложение А

Листинг кода

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Configuration;

**namespace** Dbproject

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

//Обновляет данные в dataGridView1

void UpdateGrid(string sql)

{

string str = @"Data Source=DESKTOP-PVL4P4H;Initial Catalog=Ustanovki;Integrated Security=True";

SqlConnection con;

con = new SqlConnection(str);

con.Open();

SqlCommand s\_cmd = con.CreateCommand();

s\_cmd.CommandText = sql;

SqlDataAdapter adapter = new SqlDataAdapter(s\_cmd);

DataSet n\_set = new DataSet();

adapter.Fill(n\_set);

dataGridView1.DataSource = n\_set.Tables[0];

}

// Выполняет запрос не возвращающий данные

void ExecSQL(string sql)

{

string str = @"Data Source=DESKTOP-PVL4P4H;Initial Catalog=Ustanovki;Integrated Security=True";

SqlConnection con;

con = new SqlConnection(str);

con.Open();

SqlCommand s\_cmd = con.CreateCommand();

s\_cmd.CommandText = sql;

s\_cmd.ExecuteNonQuery();

}

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

{

// **TODO**: данная строка кода позволяет загрузить данные в таблицу "ustanovkiDataSet9.ust1". При необходимости она может быть перемещена или удалена.

this.ust1TableAdapter8.Fill(this.ustanovkiDataSet9.ust1);

}

//Добавление

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

{

string ConnectionString = @"Data Source=DESKTOP-PVL4P4H;Initial Catalog=Ustanovki;Integrated Security=True";

using (SqlConnection conn = new SqlConnection(ConnectionString))

{

conn.Open();

SqlCommand command = new SqlCommand("INSERT INTO ust1 (Установка,Тип\_комплектующего,Обозначение,Наименование,Серийный\_номер,Поставщик,Изготовитель,Дата\_изготовления,Срок\_хранения,Страна\_изготовления) values (@Установка,@Тип\_комплектующего,@Обозначение,@Наименование,@Серийный\_номер,@Поставщик,@Изготовитель,@Дата\_изготовления,@Срок\_хранения,@Страна\_изготовления)");

command.Connection = conn;

command.Parameters.AddWithValue("Установка", comboBox4.Text);

command.Parameters.AddWithValue("Тип\_комплектующего", comboBox1.Text);

command.Parameters.AddWithValue("Обозначение", textBox2.Text);

command.Parameters.AddWithValue("Наименование", textBox3.Text);

command.Parameters.AddWithValue("Серийный\_номер", textBox4.Text);

command.Parameters.AddWithValue("Поставщик", textBox5.Text);

command.Parameters.AddWithValue("Изготовитель", textBox6.Text);

command.Parameters.AddWithValue("Дата\_изготовления", dateTimePicker1.Text);

command.Parameters.AddWithValue("Срок\_хранения", textBox8.Text);

command.Parameters.AddWithValue("Страна\_изготовления", textBox9.Text);

command.ExecuteNonQuery();

}

//Обновляем

string sql = "select \* from ust1";

UpdateGrid(sql);

//Отображаем данные установки выбранной в comboBox2

string ust;

ust = comboBox2.Text;

sql = "SELECT id,Установка,Тип\_комплектующего,Обозначение,Наименование,Серийный\_номер,Поставщик,Изготовитель,Дата\_изготовления,Срок\_хранения,Страна\_изготовления FROM ust1 **WHERE** Установка =" + "'" + ust + "'";

UpdateGrid(sql);

}

//Удаление строки

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

{

string Message;

Message = "Вы действительно хотите удалить выделенную запись?";

if (MessageBox.Show(Message, "Удаление", MessageBoxButtons.YesNo, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2) == DialogResult.No)

{

return;

}

string id;

id= dataGridView1.SelectedRows[0].Cells["id"].Value.ToString();

//string sql = "delete from ust1 where Установка ="+ id;

string sql = "delete from ust1 **where** id =" + "'" + id + "'";

ExecSQL(sql);//Выполняем запрос

//Обновляем

// sql = "select \* from ust1";

// UpdateGrid(sql);

//Отображаем данные установки выбранной в comboBox2

string ust;

ust = comboBox2.Text;

sql = "SELECT id,Установка,Тип\_комплектующего,Обозначение,Наименование,Серийный\_номер,Поставщик,Изготовитель,Дата\_изготовления,Срок\_хранения,Страна\_изготовления FROM ust1 **WHERE** Установка =" + "'" + ust + "'";

UpdateGrid(sql);

}

//OK открыть

**private** void button3\_Click(object sender, EventArgs e)

{

//Подключение

string str = @"Data Source=DESKTOP-PVL4P4H;Initial Catalog=Ustanovki;Integrated Security=True";

SqlConnection con;

con = new SqlConnection(str);

con.Open();

//Отображаем данные установки выбранной в comboBox2

string ust;

ust = comboBox2.Text;

string sql = "SELECT id,Установка,Тип\_комплектующего,Обозначение,Наименование,Серийный\_номер,Поставщик,Изготовитель," +

"Дата\_изготовления,Срок\_хранения,Страна\_изготовления FROM ust1 WHERE Установка =" + "'" + ust + "'";

UpdateGrid(sql);

//Вывод названия установки textBox7

textBox7.Text = ust.ToString();

}

//Удаление установки

**private** void button6\_Click(object sender, EventArgs e)

{

string Message;

Message = "Вы действительно хотите удалить выделенную установку?";

if (MessageBox.Show(Message, "Удаление", MessageBoxButtons.YesNo, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2) == DialogResult.No)

{

return;

}

string delust;

delust = comboBox3.Text;

string sql = "delete from ust1 **where** Установка =" + "'" + delust + "'" ;

ExecSQL(sql);//Выполняем запрос

//Обновляем

sql = "select \* from ust1";

UpdateGrid(sql);

//Отображаем данные установки выбранной в comboBox2

string ust;

ust = comboBox2.Text;

sql = "SELECT id,Установка,Тип\_комплектующего,Обозначение,Наименование,Серийный\_номер,Поставщик,Изготовитель,Дата\_изготовления,Срок\_хранения,Страна\_изготовления FROM ust1 **WHERE** Установка =" + "'" + ust + "'";

UpdateGrid(sql);

}

//редатировать

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

{

label30.Text = dataGridView1.SelectedRows[0].Cells["id"].Value.ToString();

comboBox5.Text = dataGridView1.SelectedRows[0].Cells["Установка"].Value.ToString();

comboBox6.Text = dataGridView1.SelectedRows[0].Cells["Тип\_комплектующего"].Value.ToString();

textBox10.Text = dataGridView1.SelectedRows[0].Cells["Обозначение"].Value.ToString();

textBox11.Text = dataGridView1.SelectedRows[0].Cells["Наименование"].Value.ToString();

textBox12.Text = dataGridView1.SelectedRows[0].Cells["Серийный\_номер"].Value.ToString();

textBox13.Text = dataGridView1.SelectedRows[0].Cells["Поставщик"].Value.ToString();

textBox14.Text = dataGridView1.SelectedRows[0].Cells["Изготовитель"].Value.ToString();

dateTimePicker2.Text = dataGridView1.SelectedRows[0].Cells["Дата\_изготовления"].Value.ToString();

textBox16.Text = dataGridView1.SelectedRows[0].Cells["Срок\_хранения"].Value.ToString();

textBox17.Text = dataGridView1.SelectedRows[0].Cells["Страна\_изготовления"].Value.ToString();

}

//сохранить редактирование

**private** void button5\_Click(object sender, EventArgs e)

{

//Подключение

string str = @"Data Source=DESKTOP-PVL4P4H;Initial Catalog=Ustanovki;Integrated Security=True";

SqlConnection con;

con = new SqlConnection(str);

con.Open();

SqlCommand command = new SqlCommand("Update ust1 set Установка=@Установка,Тип\_комплектующего=@Тип\_комплектующего,Обозначение=@Обозначение,Наименование=@Наименование,Серийный\_номер=@Серийный\_номер,Поставщик=@Поставщик,Изготовитель=@Изготовитель,Дата\_изготовления=@Дата\_изготовления,Срок\_хранения=@Срок\_хранения,Страна\_изготовления=@Страна\_изготовления where id=@id", con);

command.Parameters.AddWithValue("@id", int.Parse(label30.Text));

command.Parameters.AddWithValue("@Установка", comboBox5.Text);

command.Parameters.AddWithValue("@Тип\_комплектующего", comboBox6.Text);

command.Parameters.AddWithValue("@Обозначение", textBox10.Text);

command.Parameters.AddWithValue("@Наименование", textBox11.Text);

command.Parameters.AddWithValue("@Серийный\_номер", textBox12.Text);

command.Parameters.AddWithValue("@Поставщик", textBox13.Text);

command.Parameters.AddWithValue("@Изготовитель", textBox14.Text);

command.Parameters.AddWithValue("@Дата\_изготовления", dateTimePicker2.Text);

command.Parameters.AddWithValue("@Срок\_хранения", textBox16.Text);

command.Parameters.AddWithValue("@Страна\_изготовления", textBox17.Text);

command.ExecuteNonQuery();

//Отображаем данные установки выбранной в comboBox2

string ust;

ust = comboBox2.Text;

string sql = "SELECT id,Установка,Тип\_комплектующего,Обозначение,Наименование,Серийный\_номер,Поставщик,Изготовитель,Дата\_изготовления,Срок\_хранения,Страна\_изготовления FROM ust1 WHERE Установка =" + "'" + ust + "'";

UpdateGrid(sql);

}

//открыть комплектующие

**private** void button7\_Click(object sender, EventArgs e)

{

//Подключение

string str = @"Data Source=DESKTOP-PVL4P4H;Initial Catalog=Ustanovki;Integrated Security=True";

SqlConnection con;

con = new SqlConnection(str);

con.Open();

//Отображаем данные установки выбранной в comboBox2

string ust;

ust = comboBox2.Text;

string comp;

comp = comboBox7.Text;

string sql = "SELECT id,Установка,Тип\_комплектующего,Обозначение,Наименование,Серийный\_номер,Поставщик,Изготовитель,Дата\_изготовления,Срок\_хранения,Страна\_изготовления FROM ust1 WHERE Установка =" + "'" + ust + "'" + " and Тип\_комплектующего=" + "'" + comp + "'";

UpdateGrid(sql);

}

}

}

Список используемых источников

1. Документация по MSSQL [Электронный ресурс] // Microsoft. URL: <https://learn.microsoft.com/ru-ru/sql/relational-databases/databases/create-a-database?view=sql-server-ver16>
2. Visual Studio documentation [Электронный ресурс] // Средства разработчика, техническая документация и примеры кода | Microsoft Docs. URL: <https://learn.microsoft.com/ru-ru/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022>