Лист утверждения

|  |
| --- |
| Федеральное государственное бюджетное образовательное учреждение  Вышего образования  Российский экономический университет иимени Г.В.Плеханова  Московский приборостроительный техникум  Специальность 09.02.05 Прикладная информатика (по отраслям)  Профессиональный модуль: ПМ 02 Разработка, внедрение и адаптация программного обеспечения отраслевой направленности  Учебная практика: УП 02.01 Разработка и эксплуатация информационных систем |

|  |  |  |
| --- | --- | --- |
| Заказчик:  И.М. Щаников \_\_\_\_\_\_\_\_  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2018 г. | Консультант:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.А. Минаев  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |  |

Тема: «Разработка информационной системы: "Больница", реализация подсистем: Складские

|  |  |
| --- | --- |
| Подп и дата |  |
| Инв № дубл |  |
| Взам. инв.№ | Разработчики  студенты группы П-319  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Д.Э.Васильев  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.П. Макаров  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/В.А. Позднев |
| Подп и дата |  |
| Инд № подп |  |

помещения для медикаментов, Терапевтическое отделение, Карты

лечения, Запись на приём»

Описание языка

МПТ И-3-16 УП 02.01 РЭИС 08 18

Листов 10

|  |  |  |
| --- | --- | --- |
| Разработчики:  \_\_\_\_\_\_\_\_\_\_Д.А. Голованов  \_\_\_\_\_\_\_\_\_А.С. Гаврилочев  \_\_\_\_\_\_\_\_\_\_\_А.А. Игнатьев  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2018 г. | | |
|  | 2018 | ОЯ |

АННОТАЦИЯ

В данном документе приводится подробная информация о выбранном языке, на котором создаётся приложение для манипулирования таблицами базы данных.

СОДЕРЖАНИЕ

[1. ОБЩИЕ СВЕДЕНИЯ 4](#_Toc10452216)

[1.1. Общие характеристики языка разработки ПО 4](#_Toc10452217)

[1.2. Общие характеристики языка разработки БД 4](#_Toc10452218)

[2. ЭЛЕМЕНТЫ ЯЗЫКА 5](#_Toc10452219)

[2.1. Базовые элементы языка 5](#_Toc10452220)

[2.2. Дополнительные элементы языка 5](#_Toc10452221)

[3. СПОСОБЫ СТРУКТУРИРОВАНИЯ ПРОГРАММЫ 6](#_Toc10452222)

[3.1. Способы вызова процедур 6](#_Toc10452223)

[3.2. Методы структурирования программного кода 6](#_Toc10452224)

[4. СРЕДСТВА ОБМЕНА ДАННЫМИ 7](#_Toc10452225)

[4.1. Средства вывода данных 7](#_Toc10452226)

[4.2. Средства ввода данных 7](#_Toc10452227)

[5. ВСТРОЕННЫЕ ЭЛЕМЕНТЫ 8](#_Toc10452228)

[5.1. Описание классов 8](#_Toc10452229)

[5.2. Описание методов 8](#_Toc10452230)

[5.3. Прочие встроенные элементы 10](#_Toc10452231)

# 1. ОБЩИЕ СВЕДЕНИЯ

## 1.1. Общие характеристики языка разработки ПО

1.1.1. Возможности языка

Для разработки ПО был выбран язык C#.

C# является объектно-ориентированным языком, с которым можно не только просто разработать ПО, но и так же просто поддерживать его т.к. язык поддерживает контроль версий.

В C# представлена концепция пространств имен, аналогичная пакетам в Java. Это позволяет иерархически структурировать Вашу систему типов, делая код намного более понятным и позволяя избежать проблем с именованием.

C# имеет C стиль синтаксиса (для управляющих конструкций, блоков кода, описания сигнатуры методов и др.), имеет много общего с Java (отсутствие множественного наследования и шаблонов, наличие сборщика мусора) и Дельфи (ориентированность на создание компонент).

1.1.2. Область применения языка

На C# можно реализовать почти что угодно: классический WindowsForms Application, WPF, приложения для магазина Windows Store, Мобильные приложения, приложения для Linux. Но в основном используется для создания настольных приложений (WindowsForms Application и WPF)

## 1.2. Общие характеристики языка разработки БД

1.2.1. Возможности языка

Для реализации БД была выбрана СУБД Microsoft SQL Server.

Язык SQL позволяет определить достаточно сложные ограничения целостности, удовлетворение которым будет проверяться при всех модификациях базы данных.

Операторы языка SQL позволяют извлекать данные более чем из одной таблицы. Одна из возможностей сделать это заключается в связывании таблиц по одному общему полю.

Microsoft SQL работает быстро и стабильно, Oracle требует много ресурсов, и Microsoft SQL проще чем MySQL.

Microsoft SQL хорошо взаимодействует с другими продуктами Microsoft

1.2.2. Область применения языка

Язык Microsoft SQL используется для создания/дополнения базы данных, создания запросов для получения данных из базы данных. А также для обеспечения автономности БД и целостности данных.

# 2. ЭЛЕМЕНТЫ ЯЗЫКА

## 2.1. Базовые элементы языка

Библиотеки пространства имён, используемые в программе

using System;   
using [System.Data](https://vk.com/away.php?to=http%3A%2F%2FSystem.Data&cc_key=" \t "_blank);   
using System.Drawing;   
using System.Windows.Forms;   
using System.Data.SqlClient;   
using System.Threading;

Классы, используемые в программе

LinkLabel, Button, SqlParameter, DataTable, DataGridView, TextBox, Label, ComboBox, CheckBox, Panel, Thread , PictureBox , DataRow, SqlCommand,SqlConnection, DataBaseConfiguration, Resoursces, а такжесобственные классы: ConnectionForm,AutorizForm, RegForm, DB, CreateWriteForm, DiagForm, DischargeForm, DeliMediForm, EditDelForm, Shifrovanie, Reestr.

Визуальные объекты

Динамические формы,а также элементы, такие как: Button, DataGrid, ComboBox, TextBox, label, CheckBox, PictureBox

Не визуальные компоненты в программе не применялись

## 2.2. Дополнительные элементы языка

Также в программе применялась самостоятельно написанные библиотеки: RegEdit.dll и Shifrovanie.dll, отвечающие за шифрование первого входа в реестре.

# 3. СПОСОБЫ СТРУКТУРИРОВАНИЯ ПРОГРАММЫ

## 3.1. Способы вызова процедур

Для вызова процедур, осуществляющих ввод данных из БД в программу используется модуль DB.cs, с помощью которого происходит заполнение таблиц и прочих элементвом значениями из Базы данных

## 3.2. Методы структурирования программного кода

Для сокращения программного кода и увеличения производительности было применено сокращение программного кода в следующих классах: DB.cs, MainForm. Код выглядит следующим образом:

1. Вместо постоянного создания новых процедур по вызову из базы данных, используется следующий программный код, куда передаётся лишь название самой процедуры и её вызов:

SqlConnection sql = new SqlConnection(DataBaseConfiguration.connectString());   
[sql.Open](https://vk.com/away.php?to=http%3A%2F%2Fsql.Open&cc_key=)();   
SqlCommand UpdData = new SqlCommand(ProcName, sql);   
UpdData.CommandType = CommandType.StoredProcedure;   
for (int i = 0; i < SP.Length; i++)   
{   
UpdData.Parameters.Add(SP[i]);   
}   
UpdData.ExecuteNonQuery();   
sql.Close();  
LastOp = true;

1. Для сокращения создания динамических элементов на формах используется следующий код:

(ControlPanel.Controls["CreateRoleButton"] as Button).Dispose();

После происходит вызов и созданиек кнопки:

CreateButton(ControlPanel, "OpenForm", "Добавить запись", 70, 30, 220, 50, 12);

# 4. СРЕДСТВА ОБМЕНА ДАННЫМИ

## 4.1. Средства вывода данных

Для вывода документов используется отдельный модуль под названием DocumentOutPut.cs, в котором находится код с использованием библиотек Excel, Word и PDF. Здесь формируется шаблон, куда через переменные вставляются значения из базы данных.

## 4.2. Средства ввода данных

Для ввода данных в программу используется модуль DB.cs, с помощью которого происходит заполнение таблиц и прочих элементвом значениями из Базы данных

# 5. ВСТРОЕННЫЕ ЭЛЕМЕНТЫ

## 5.1. Описание классов

AutorizForm.cs – динамическая форма, в которой происходит авторизация в програму, т.е. ввод логина и пароля, после чего пользователь получает доступ к функциям, обозначенными его ролью.

ConnectionForm.cs – динамическая форма для подключения к базе данных, возможность смены БД и сервера.

CreateWriteForm.cs – динамическая форма создания формы записи с заполнением данных.

DataBaseConfiguration.cs – класс для поиска серверов и баз на сервере.

DB.cs – класс для управления БД в приложении.

DeliMediForm.cs – динамическая форма, созданная для управления поставками медикаментов.

DiagForm.cs – динамическая форма для создания и назначения диагнозов.

DischargeForm.cs – динамическая форма для создания и управления выписками.

DocumentOutPut.cs – динамическая форма, отвечающая за печать выходнывх документов.

EditDelForm.cs – класс, вызывающий окно с вопросом при удалении.

MedicamentsForm.cs – динамическая форма для управления таблицой медикаментов и назначения им различных параметров.

RegForm.cs – форма для регистрации, на которую можно перейти и создать новый аккуант для авторизации.

ProfileForm.cs – динамическая форма, содержащая в себе информацию о вошедшем в аккуант сотруднике, его должность, а также смену пароля.

## 5.2. Описание методов

Метод на выгрузку данных:

public static DataTable GetData(string ProcName, SqlParameter[] SP)

{

try

{

SqlConnection sql = new SqlConnection(DataBaseConfiguration.connectString());

sql.Open();

SqlCommand GetData = new SqlCommand(ProcName, sql);

GetData.CommandType = CommandType.StoredProcedure;

for (int i = 0; i < SP.Length; i++)

{

GetData.Parameters.Add(SP[i]);

}

DataTable DT = new DataTable();

SqlDataReader DR = GetData.ExecuteReader();

DT.Load(DR);

sql.Close();

LastOp = true;

return DT;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

LastOp = false;

return new DataTable();

}

}

Метод на изменение данных:

public static int WriteData(string ProcName, SqlParameter[] SP)

{

try

{

SqlConnection sql = new SqlConnection(DataBaseConfiguration.connectString());

sql.Open();

SqlCommand AddCitizen = new SqlCommand(ProcName, sql);

AddCitizen.CommandType = CommandType.StoredProcedure;

for (int i = 0; i < SP.Length; i++)

{

AddCitizen.Parameters.Add(SP[i]);

}

var id = AddCitizen.ExecuteNonQuery();

sql.Close();

LastOp = true;

return (int)id;

}

catch(Exception ex)

{

MessageBox.Show(ex.Message.ToString());

LastOp = false;

return -10;

}

}

Метод на логическое удаление:

public static void LogDelData(string ProcName, SqlParameter[] SP)

{

try

{

SqlConnection sql = new SqlConnection(DataBaseConfiguration.connectString());

sql.Open();

SqlCommand UpdData = new SqlCommand(ProcName, sql);

UpdData.CommandType = CommandType.StoredProcedure;

for (int i = 0; i < SP.Length; i++)

{

UpdData.Parameters.Add(SP[i]);

}

UpdData.ExecuteNonQuery();

sql.Close();

LastOp = true;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

LastOp = false;

}

}

Метод на обновление (редактирвание данных):

public static void UpdData(string ProcName, SqlParameter[] SP)

{

try

{

SqlConnection sql = new SqlConnection(DataBaseConfiguration.connectString());

sql.Open();

SqlCommand UpdData = new SqlCommand(ProcName, sql);

UpdData.CommandType = CommandType.StoredProcedure;

for (int i = 0; i < SP.Length; i++)

{

UpdData.Parameters.Add(SP[i]);

}

UpdData.ExecuteNonQuery();

sql.Close();

LastOp = true;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

LastOp = false;

}

}

}

}

Остальные методы, выполнены на подобие вышеперечисленных

## 5.3. Прочие встроенные элементы

Динамический объект и метод к нему

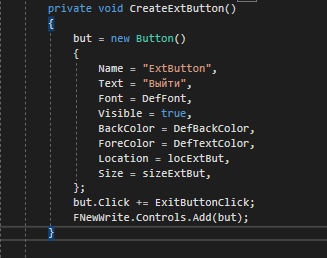


Рисунок 1 - Пример динамического объекта