Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский экономический университет им. 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

Междисциплинарный курс: МДК 02.01 Разработка, внедрение и адаптация программного обеспечения отраслевой направленности

МПТ.И-1,2,3-17.МДК.02.01 00 20

ОТЧЁТ

по практическим работам:

Практическая работа № 28 «Организация модулей работы с источниками данных. Таблицы и представления.»

Практическая работа № 29 «Организация модулей работы с источниками данных. Хранимые процедуры.»

Практическая работа № 30 «Разработка модуля подключения к источнику данных.»

Практическая работа № 31 «Разработка интерфейсов справочников.»

Практическая работа № 32 «Разработка стандартных функций баз данных.»

Практическая работа № 33 «Разработка автоматизированных функций программного обеспечения.»

Практическая работа № 34 «Разработка экспорта данных в Microsoft Word, Excel и формат PDF.»

Практическая работа № 35 «Разработка системы аутентификации пользователей.»

Практическая работа № 36 «Публикация программного продукта в Git Hub.»

|  |  |
| --- | --- |
|  | Выполнил |
| Проверил: | Студент группы И-2-17 |
| И.М. Щаников \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н. Лыткин |
| «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |
| Оценка: \_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_) |  |

2020

СОДЕРЖАНИЕ

[1. ЦЕЛИ ПРАКТИЧЕСКОЙ РАБОТЫ 3](#_Toc21173925)

[2. ЭТАПЫ ВЫПОЛНЕНИЯ 5](#_Toc21173926)

[3. ВЫВОД 6](#_Toc21173927)

[4. СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 7](#_Toc21173928)

1. ЦЕЛИ ПРАКТИЧЕСКОЙ РАБОТЫ

Практическая работа № 28:

* Разрабатывать программное обеспечение с помощью языков программирования информационного контента;
* Программировать на встроенных алгоритмических языках.

Практическая работа № 29:

* Разрабатывать программное обеспечение с помощью языков программирования информационного контента;
* Программировать на встроенных алгоритмических языках.

Практическая работа № 30:

* Разрабатывать программное обеспечение с помощью языков программирования информационного контента;
* Программировать на встроенных алгоритмических языках.

Практическая работа № 31:

* Разрабатывать программное обеспечение с помощью языков программирования информационного контента;
* Создавать анимации в специализированных программных средах;
* Работать с мультимедийными инструментальными средствами.

Практическая работа № 32:

* Разрабатывать программное обеспечение с помощью языков программирования информационного контента.

Практическая работа № 33:

* Разрабатывать программное обеспечение с помощью языков программирования информационного контента;
* Программировать на встроенных алгоритмических языках.

Практическая работа № 34:

* Разрабатывать программное обеспечение с помощью языков программирования информационного контента;
* Программировать на встроенных алгоритмических языках.

Практическая работа № 35:

* Разрабатывать программное обеспечение с помощью языков программирования информационного контента;
* Программировать на встроенных алгоритмических языках.

Практическая работа № 36:

* Размещать информационный контент в глобальных и локальных сетях;
* Использовать инструментальные среды поддержки разработки, системы управления контентом;
* Использовать системы управления контентом для решения поставленных задач.

1. ЭТАПЫ ВЫПОЛНЕНИЯ
   1. Практическая работа № 28
      1. Описание структуры базы данных

Таблица 1 – Описание структуры базы данных

|  |  |
| --- | --- |
| Название элемента | Краткое описание назначения объекта |
| [dbo].[Dolgnost] | С помощью данной сущности производится хранение должностей |
| [dbo].[Sotrudniki] | С помощью данной сущности производится хранение сотрудников |
| [dbo].[Pokupatel] | С помощью данной сущности производится хранение покупателей |
| [dbo].[Usluga] | С помощью данной сущности производится хранение услуг |
| [dbo].[Tovar] | С помощью данной сущности производится хранение товаров |
| [dbo].[Dogovora] | С помощью данной сущности производится хранение договоров |
| [dbo].[Tovarnaya\_Nakladnaya] | С помощью данной сущности производится хранение товарных накладных |
| [dbo].[Transportnaya\_Nakladnaya] | С помощью данной сущности производится хранение транспортных накладных |

* + 1. Описание модулей работы с таблицами и представлениями

Таблица 2 – Описание модулей программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название модуля | Краткое описание | Пространства имён | Классы | Количество строк |
| Table\_Class.cs | Модуль предназначен для работы с сущностями | System;  System.Data;  System.Data.Sql;  System.Data.SqlClient;  Microsoft.Win32;  ING | DataTable, SqlCommand, SqlDependency, Configuration\_Class | 52 |

* + 1. Код программы

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ING

{

class Table\_Class

{

//Глообальный класс виртуальной таблицы

public DataTable table = new DataTable();

//Локальная переменная SQLCommand

private SqlCommand command = new SqlCommand("", Configuration\_class.connection);

//Глобальная переменная организации взависимости и прослушивание серверов

public SqlDependency Dependency = new SqlDependency();

/// <summary>

/// Заполнение DataTable в зависимости от введеного SQL запроса

/// </summary>

/// <param name="SQL\_Select\_Query"></param>

public Table\_Class(String SQL\_Select\_Query)

{

command.Notification = null; //Отключение уведомлений

command.CommandText = SQL\_Select\_Query;//Присвоение SQL запроса SqlCommand

Dependency.AddCommandDependency(command);//Присвоение команд в связку

//прослушивание

try

{

//Запуск прослушиания

SqlDependency.Start(Configuration\_class.connection.ConnectionString);

//Открытия подключения

Configuration\_class.connection.Open();

//Запись данных в табличном виде в вирт. табл.

table.Load(command.ExecuteReader());

}

catch (Exception ex)

{

//Сообщение об ошибке

System.Windows.Forms.MessageBox.Show(ex.Message);

}

finally

{

//Закрытие подключения

Configuration\_class.connection.Close();

}

}

}

}

* 1. Практическая работа № 29
     1. Описание структуры базы данных

Таблица 1 – Описание структуры базы данных

|  |  |
| --- | --- |
| Название элемента | Краткое описание назначения объекта |
| [dbo].[Dolgnost\_insert] | Данная процедура предназначена для добавления данных в сущность должность |
| [dbo].[Dolgnost\_update] | Данная процедура предназначена для изменения данных, в сущности, должность |
| [dbo].[Dolgnost\_delete] | Данная процедура предназначена для удаления данных, в сущности, должность |
| [dbo].[Sotrudniki\_insert] | Данная процедура предназначена для добавления данных в сущность сотрудники |
| [dbo].[Sotrudniki\_update] | Данная процедура предназначена для изменения данных, в сущности, сотрудники |
| [dbo].[Sotrudniki\_delete] | Данная процедура предназначена для удаления данных, в сущности, сотрудники |
| [dbo].[Pokupatel\_insert] | Данная процедура предназначена для добавления данных в сущность покупатели |
| [dbo].[Pokupatel\_update] | Данная процедура предназначена для изменения данных, в сущности, покупатели |
| [dbo].[Pokupatel\_delete] | Данная процедура предназначена для удаления данных, в сущности, покупатели |
| [dbo].[Usluga\_insert] | Данная процедура предназначена для добавления данных в сущность услуга |
| [dbo].[Usluga\_update] | Данная процедура предназначена для изменения данных, в сущности, услуга |
| [dbo].[Usluga\_delete] | Данная процедура предназначена для удаления данных, в сущности, услуга |
| [dbo].[Dogovora\_insert] | Данная процедура предназначена для добавления данных в сущность договора |
| [dbo].[Dogovora\_update] | Данная процедура предназначена для изменения данных, в сущности, договора |
| [dbo].[Dogovora\_delete] | Данная процедура предназначена для удаления данных, в сущности, договора |
| [dbo].[Tovarnaya\_Nakladnaya\_insert] | Данная процедура предназначена для добавления данных в сущность товарная накладная |
| [dbo].[Tovarnaya\_Nakladnaya\_update] | Данная процедура предназначена для изменения данных, в сущности, товарная накладная |
| [dbo].[Tovarnaya\_Nakladnaya\_delete] | Данная процедура предназначена для удаления данных, в сущности, товарная накладная |
| [dbo].[Transportnaya\_Nakladnaya\_insert] | Данная процедура предназначена для добавления данных в сущность транспортная накладная |
| [dbo].[Transportnaya\_Nakladnaya\_update] | Данная процедура предназначена для изменения данных, в сущности, транспортная накладная |
| [dbo].[Transportnaya\_Nakladnaya\_delete] | Данная процедура предназначена для удаления данных, в сущности, транспортная накладная |
| [dbo].[Tovar\_insert] | Данная процедура предназначена для добавления данных в сущность товар |
| [dbo].[Tovar\_update] | Данная процедура предназначена для изменения данных, в сущности, товар |
| [dbo].[Tovar\_delete] | Данная процедура предназначена для удаления данных, в сущности, товар |

* + 1. Описание модулей работы с хранимыми процедурами

Таблица 2 – Описание модулей программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название модуля | Краткое описание | Пространства имён | Классы | Количество строк |
| Procedure\_Class.cs | Данный модуль предназначен для работы с хранимыми процедурами | System.Collections;  System.Data;  System.Data.SqlClient;  ING | DataTable, SqlCommand, SqlDependency, Configuration\_Class | 59 |

* + 1. Код программы

using System.Collections;

using System.Data;

using System.Data.SqlClient;

namespace ING

{

class Procedure\_Class

{

SqlCommand command = new SqlCommand("", Configuration\_class.connection);

/// <summary>

/// Мктод обращения к любой хранимой процедуре M SQL serever

/// </summary>

/// <param name="Procedure\_Name"> вводимое знание процедуры из бд</param>

/// <param name="filed\_value"> Не типизированная коллекция значений приложения</param>

public void procedure\_Execution(string Procedure\_Name, ArrayList filed\_value)

{

//запрос на вывод списка параметров, конкретной хранимой процедуры

//в зависимости от введённого разработчиком названия STORED pROCEDURE

Table\_Class table = new Table\_Class(string.Format("Select name from sys.parameters where " + "Object\_id =" +

"(select object\_id from sys.procedures where name = '{0}')", Procedure\_Name));

try

{

//Настройка SQLCommand для работы сс хранимыми процедурами

command.CommandType = CommandType.StoredProcedure;

//Присвоение в текст команды названия хранимой процеуры

command.CommandText = string.Format("[dbo].[{0}]", Procedure\_Name);

//Очистка параметров

command.Parameters.Clear();

for (int i = 0; i < table.table.Rows.Count; i++)

{

command.Parameters.AddWithValue(table.table.Rows[i][0].ToString(), filed\_value[i]);

}

Configuration\_class.connection.Open();

//Обьявление события на перехват сообщение из БД

Configuration\_class.connection.InfoMessage += Connection\_InfoMessage;

command.ExecuteNonQuery();

}

catch

{

}

finally

{

Configuration\_class.connection.Close();

}

}

/// <summary>

/// Обработчик событий о получении с сервера БД

/// </summary>

/// <param name="sender">Ссылка на объект</param>

/// <param name="e">Аргумент сообщения сервера</param>

private void Connection\_InfoMessage(object sender, SqlInfoMessageEventArgs e)

{

System.Windows.Forms.MessageBox.Show(e.Message);

Configuration\_class.connection.InfoMessage -= Connection\_InfoMessage;

}

}

}

* 1. Практическая работа № 30
     1. Описание пространств имён

Таблица 1 – Описание пространств имён

|  |  |
| --- | --- |
| Название элемента | Краткое описание назначения элемента |
| ING | Предназначен для работы в данном приложении и обозначения области видимости. |
| RegistryKey | Предназначен для работы с реестром |
| DataTable | Предназначен для предоставления одной таблицы данных в памяти |
| SqlCommand | Предназначен для работы с процедурами над базой данных |
| SqlDependency | Предназначен для установления зависимости уведомления запроса. |

* + 1. Описание классов

В данном пункте в виде таблицы необходимо описать название классов, и назначение данного пространства в рамках выполняемой работы.

Таблица 2 – Описание классов

|  |  |
| --- | --- |
| Название элемента | Краткое описание назначения элемента |
| Cpnfiguration\_Class | Данный класс предназначен для подключения к серверу бд и выбору самой бд. |
| Table\_Class | Данный класс предназначен для работы с таблицами бд. |
| Procedure\_Class | Данный класс предназначен для работы с хранимыми процедурами бд. |

* + 1. Код программы

using System;

using System.Data;

using System.Data.Sql;

using System.Data.SqlClient;

using Microsoft.Win32;

namespace ING

{

class Configuration\_class

{

public event Action<DataTable> server\_Collection;

//Получение коллекцию доступных серверов

public event Action<DataTable> Data\_Base\_Collection;

//Получения коллекции доступных БД на сервере

public event Action<bool> connection\_checked;

//Статус подключения

public string DS = "Empty", //Переменная data Source

IC = "Empty";

public string ds = ""; //Проверка подключение data source

public static SqlConnection connection = new SqlConnection();

/// <summary>

/// Так можно отдать ссылку на метод

/// </summary>

public void SQL\_Server\_Configuration\_get()

{

RegistryKey registry = Registry.CurrentUser;

RegistryKey key = registry.CreateSubKey("Server\_Configuration");

try

{

DS = key.GetValue("DS").ToString();

IC = key.GetValue("IC").ToString();

}

catch

{

DS = "Empty";

IC = "Empty";

}

finally

{

//Обновление строки подключения

connection.ConnectionString = "Data Source = " + DS + "; Initial Catalog = " + IC + "; Integrated Security = true;";

}

}

/// <summary>

/// Метод обновения информации о подключение к источнику данных

/// по технологии ADO.NET

/// </summary>

public void SQL\_Server\_Configuration\_Set(string ds, string ic)

{

RegistryKey registry = Registry.CurrentUser;

RegistryKey key = registry.CreateSubKey("Server\_Configuration");

key.SetValue("DS", ds);//Запись значения в переменную реестра

key.SetValue("IC", ic);

SQL\_Server\_Configuration\_get();

}

/// <summary>

/// Метод возвращает список доустпных сервров в локалке

/// </summary>

public void SQL\_Server\_Enumurator()

{

//получить сведения доступных серверов

SqlDataSourceEnumerator sourceEnumerator = SqlDataSourceEnumerator.Instance;

//Присвоение Event Action списка серверов ввиде таблицы

server\_Collection(sourceEnumerator.GetDataSources());

}

/// <summary>

/// Метод проверки подключение к источнику данных

/// </summary>

public void SQL\_Data\_Base\_Checking()

{

connection.ConnectionString = "Data source = " + ds + "; " + "Initial Catalog = master; Integrated Security = true";

try

{

//Если подключение по источнику данных

//в приваивает true

connection.Open();

connection\_checked(true);

}

catch

//В противном случае false

{

connection\_checked(false);

}

finally

{

connection.Close();

}

}

public void SQL\_Data\_Base\_Collection()

{

//нахождение базы с названием Sale\_Data\_Base и не назвыается master,model и т.п

SqlCommand command = new SqlCommand("select name from sys.databases " + "where name not in ('master','tempdb','model','msdb')", connection);

try

{

connection.Open();

DataTable table = new DataTable();

table.Load(command.ExecuteReader());

Data\_Base\_Collection(table);

}

catch

{

}

finally

{

connection.Close();

}

}

public static string Organiztion\_Name, Save\_Files\_Path, Machine\_Name;

public static Int32 doc\_Left\_Merge, doc\_Right\_Merge, doc\_Top\_Merge, doc\_Bottom\_Merge;

public void Document\_Configuration\_Get()

{

RegistryKey registry = Registry.CurrentUser;

RegistryKey key = registry.CreateSubKey("Server\_COnfiduration");

try

{

Organiztion\_Name = key.GetValue("Organization\_Name").ToString();

doc\_Left\_Merge = Convert.ToInt32(key.GetValue("doc\_Left\_Merge").ToString());

doc\_Right\_Merge = Convert.ToInt32(key.GetValue("doc\_Right\_Merge").ToString());

doc\_Top\_Merge = Convert.ToInt32(key.GetValue("doc\_Top\_Merge").ToString());

doc\_Bottom\_Merge = Convert.ToInt32(key.GetValue("doc\_Bottom\_Merge").ToString());

}

catch

{

Organiztion\_Name = "Empty";

doc\_Left\_Merge = 0;

doc\_Right\_Merge = 0;

doc\_Top\_Merge = 0;

doc\_Bottom\_Merge = 0;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ING

{

static class Program

{

public static string intID = "";

public static Int32 intDropStatis = 0;

public static string strStatus = "Null";

public static bool connect = false;

private static Mutex \_instanse;

private const string \_app\_name = "INGPROMTORG";

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

bool Create\_app;

\_instanse = new Mutex(true, \_app\_name, out Create\_app);

if (Create\_app)

{

try

{

Configuration\_class configuration = new Configuration\_class();

configuration.SQL\_Server\_Configuration\_get();

Configuration\_class.connection.Open();

connect = true;

}

catch

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Connection\_Form conection = new Connection\_Form();

conection.ShowDialog();

}

finally

{

Configuration\_class.connection.Close();

switch (connect)

{

case false:

MessageBox.Show("Ошибка подключения", "Продажа товара", MessageBoxButtons.OK, MessageBoxIcon.Error);

Environment.Exit(0);

break;

case true:

try

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Authorization\_From());

}

catch

{

}

break;

}

}

}

}

}

}

* + 1. Результат работы

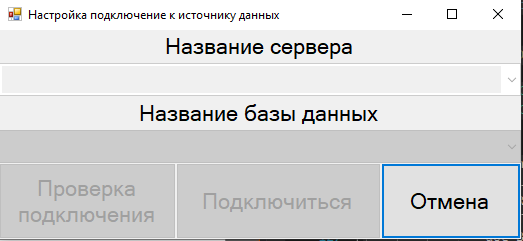


Рисунок – от 1 этап

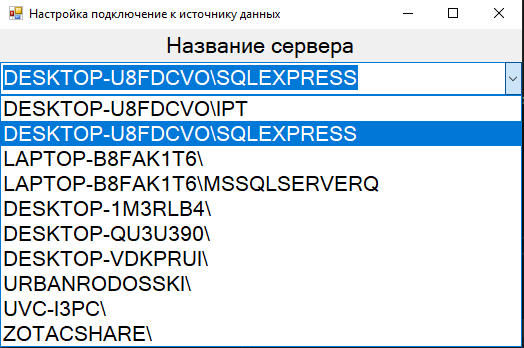


Рисунок – 2 этап

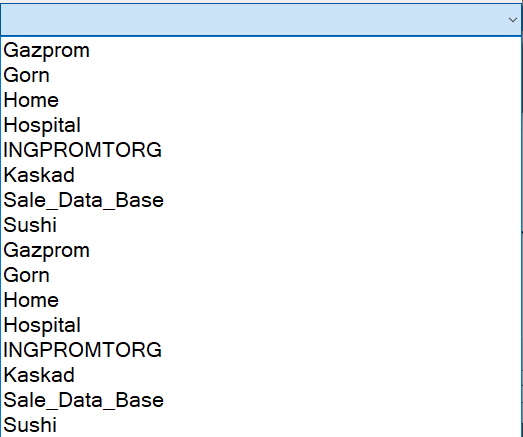


Рисунок – 3 этап



Рисунок – 4 этап

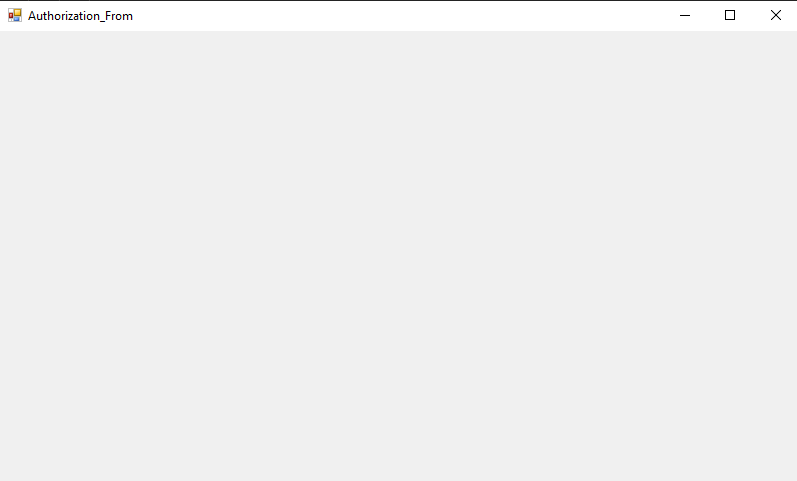


Рисунок - до

* 1. Практическая работа № 31
     1. Описание окон приложения

В данном пункте в виде таблицы описать разрабатываемые окна справочных таблиц, с указанием названия и описанием тех таблиц, которые в них будут выводить.

Таблица 1 – Описание окон приложения

|  |  |
| --- | --- |
| Название окна | Функциональное описание окна |
|  |  |

* + 1. Описание классов

В данном пункте в виде таблицы необходимо описать название классов, и назначение данного пространства в рамках выполняемой работы.

Таблица 2 – Описание классов

|  |  |
| --- | --- |
| Название элемента | Краткое описание назначения элемента |
|  |  |
|  |  |

* + 1. Описание визуальных компонентов

В данном пункте необходимо описать требования к визуальным компонентам, с помощью которых будут производиться манипулирования данных в приложении БД. Данный пункт является эталоном создания и реализации интерфейса, т.е. описывается обязательно внешний вид элементов, каких они будут размеров, их функциональное назначение (UI). Во второй таблице описать предполагаемые реакции элементов на взаимодействие с пользователями (UX).

Таблица 3 – UI описание

|  |  |
| --- | --- |
| Скриншот элемента | Описание свойств элемента |
|  |  |

Таблица 4 – UX описание

|  |  |
| --- | --- |
| Скриншот элемента | Описание реакции элемента |
|  |  |

* + 1. Код программы

В данном пункте даётся полный код программы модуля с системными и прочими комментариями по программному коду, стилистика программирования должна быть выдержана.

* + 1. Результат работы программы

В данном пункте продемонстрировать скриншоты работы «от и до» окна подключения к источнику данных.

* 1. Практическая работа № 32
     1. Код программы
        1. Добавление данных

В данном подпункте описать код программы методов добавления данных в базу данных через приложение, с полним описанием кода программы с использованием системных и обыкновенных комментариев.

* + - 1. Изменение данных

В данном подпункте описать код программы методов изменения данных в базу данных через приложение, с полним описанием кода программы с использованием системных и обыкновенных комментариев.

* + - 1. Удаление данных (Физическое)

В данном подпункте описать код программы методов физического удаления данных из базы данных через приложение, с полним описанием кода программы с использованием системных и обыкновенных комментариев.

* + - 1. Удаление данных (Логическое)

В данном подпункте описать код программы методов логического удаления данных из базы данных через приложение, с полним описанием кода программы с использованием системных и обыкновенных комментариев.

* + - 1. Фильтрация данных

В данном подпункте описать код программы методов фильтрации данных в базе данных через приложение, с полним описанием кода программы с использованием системных и обыкновенных комментариев.

* + - 1. Поиск данных

В данном подпункте описать код программы методов поиска данных в базе данных через приложение, с полним описанием кода программы с использованием системных и обыкновенных комментариев.

* + 1. Результат работы программы

В данном пункте продемонстрировать скриншоты работы «от и до» окон манипулирования над данными.

* 1. Практическая работа № 33
     1. Описание блоков функций бизнес-процессов

В данном пункте необходимо описать порядок выполнения бизнес-процессов в приложении на основе бизнес-процессов DFD, в виде нумерованного списка на против каждый блок процесса.

Таблица 1- Описание логики реализации бизнес процессов

|  |  |
| --- | --- |
| Блок DFD | Порядок выполнения |
|  |  |

* + 1. Код программы

В данном пункте даётся полный код программы модуля с системными и прочими комментариями по программному коду, стилистика программирования должна быть выдержана.

* + 1. Результат работы программы

В данном пункте продемонстрировать скриншоты работы «от и до» окон манипулирования над данными.

* 1. Практическая работа № 34
     1. Требования к экспорту

В данном пункте в виде таблицы описать требования к реализации выходных документов на основе модели EPC.

Таблица 1 – Описание выходных документов

|  |  |
| --- | --- |
| Скриншот нотации EPC | Требования к содержанию выходного документа |
|  |  |

* + 1. Код программы

В данном пункте даётся полный код программы модуля с системными и прочими комментариями по программному коду, стилистика программирования должна быть выдержана.

* + 1. Результат работы программы

В данном пункте продемонстрировать скриншоты работы «от и до» окон манипулирования над данными.

* 1. Практическая работа № 35
     1. Описание принципа аутентификации
        1. Авторизация

В данном подпункте описать в вольной форме требования к авторизации в системе, с указанием либо статических или динамических прав доступа к системе. Указать все роли, с описанием прав доступа.

* + - 1. Регистрация

В данном подпункте указать требования к регистрации в системе, с перечислением, каким образом будет происходить регистрация, кто будет выдавать права и как данный интерфейс будет реализован по функциональным характеристикам.

* + 1. Код программы

В данном пункте даётся полный код программы модуля с системными и прочими комментариями по программному коду, стилистика программирования должна быть выдержана.

* + 1. Результат работы программы

В данном пункте продемонстрировать скриншоты работы «от и до» окон манипулирования над данными.

* 1. Практическая работа № 36
     1. Описание персонального репозитория
        1. Создание репозитория

В данном подпункте в виде скриншотов показать пошаговое создание удалённого репозитория к текущему проекту.

* + - 1. Настройка среды разработки

В данном пункте показать в виде скриншотов пошаговую настройку репозитория в среде разработки.

* + 1. Отправка проекта на репозиторий

В данном пункте показать в виде скриншотов механизм Push проекта на удалённый репозиторий, через локальный Commit.

* + 1. Скачивание проекта с репозитория

В данном пункте показать в виде скриншотов механизм Pull проекта из удалённого репозитория.

1. ВЫВОД

В данном разделе на основе целей к практическим работам и проведённой оценке пишется полный вывод о проделанной работе, с полной выкладкой и спецификой индивидуальной темы. Так же делается заключение о программном продукте или ресурсе.

1. СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

В данном разделе перечисляются все источники, которые использовались при реализации решений.