МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Факультет систем управління літальними апаратами

Кафедра 302 «Інформаційні управляючі системи»

**Лабораторна робота № 2**

з дисципліни «Технологія створення програмного продукту»

на тему: «Створення тонкого клієнту»

Виконала: студентка 2 курсу групи 327 ст

напряму підготовки (спеціальності)

6.050101 Комп’ютерні науки

Крейзе М.В,

Прийняв: доц. каф 302 Момот М.О\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Національна шкала: \_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Харків – 2018

**Лабораторная работа № 2**

Тема: Створення тонкого клієнту

**Дополнительное задание:**

1. Главная форма содержит меню, в котором есть обязательные пункты File (Exit) Help(About –с информацией о создателях, Help context) – для связи с последующей лабораторной работой.
2. Приложение содержит несколько окон для отображения данных, причем некоторые из них модальные.
3. Формы отображения таблиц содержат управляющие элементы DBNavigator, DBGrid и др. из закладки компонентов Data Controls.
4. Заголовки полей в DBGrid – по-русски.
5. Помеченные \* таблицы имеют графическое поле – отображать содержимое на форме.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**Тонкий клиент**

*Тонкий клиент* (англ. *thin client*) в компьютерных технологиях — компьютер или программа-клиент в сетях с клиент-серверной или терминальной архитектурой, который переносит все или большую часть задач по обработке информации на сервер. Примером тонкого клиента может служить компьютер с браузером, использующийся для работы с веб-приложениями.

***Web-клиенты (англ. WEB clients)***

Web-клиент как программа — браузер. Web-клиент как устройство — устройство, основным приложением которого (с точки зрения разработчика устройства или маркетолога) является браузер.

***Тонкие клиенты, работающие в терминальном режиме***

Под термином «тонкий клиент» подразумевается достаточно широкий с точки зрения системной архитектуры ряд устройств и программ, которые объединяются общим свойством: возможность работы в терминальном режиме. Таким образом, для работы тонкого клиента необходим терминальный сервер. Этим тонкий клиент отличается от толстого клиента, который, напротив, производит обработку информации независимо от сервера, используя последний в основном лишь для хранения данных.

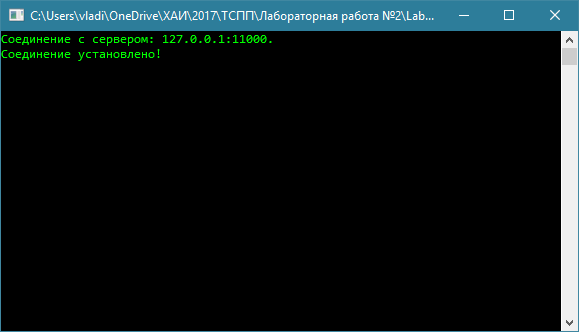
Сравнение размеров — тонкий клиент (справа) в сравнении с традиционным настольным ПК

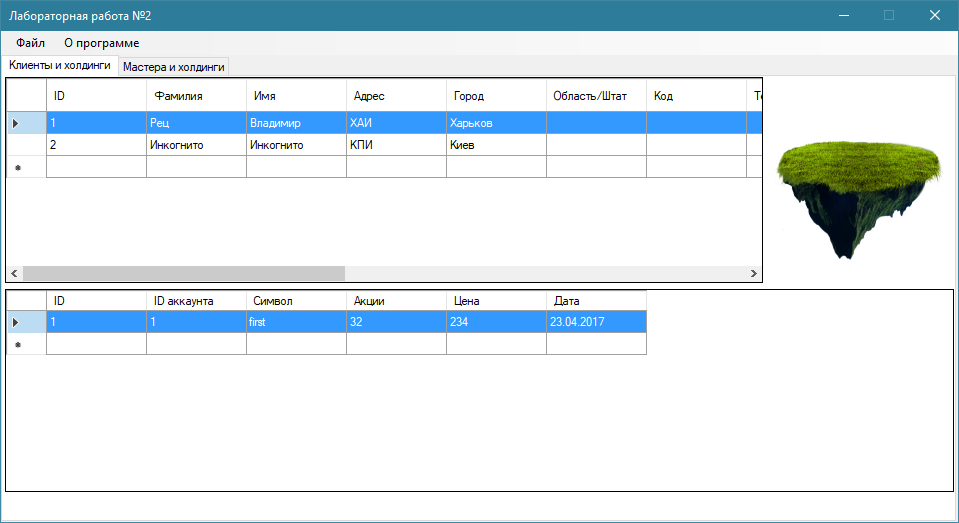
Кроме общего случая, следует выделить аппаратный тонкий клиент (например, Windows- и Linux-терминалы) — специализированное устройство, принципиально отличное от ПК. Аппаратный тонкий клиент не имеет жёсткого диска, использует специализированную локальную ОС (одна из задач которой организовать сессию с терминальным сервером для работы пользователя), не имеет в своём составе подвижных деталей, выполняется в специализированных корпусах с полностью пассивным охлаждением.

Для расширения функциональности тонкого клиента прибегают к его «утолщению», например, добавляют возможности автономной работы, сохраняя главное отличие — работу в сессии с терминальным сервером. Когда в клиенте появляются подвижные детали (жёсткие диски), появляются возможности автономной работы, он перестаёт быть тонким клиентом в чистом виде, а становится универсальным клиентом.

Тонкий клиент в большинстве случаев обладает минимальной аппаратной конфигурацией, вместо жёсткого диска для загрузки локальной специализированной ОС используется DOM (DiskOnModule) [модуль с разъёмом IDE, флэш-памятью и микросхемой, реализующей логику обычного жёсткого диска — в BIOS определяется как обычный жёсткий диск, только размер его обычно в 2-3 раза меньше]. В некоторых конфигурациях системы тонкий клиент загружает операционную систему по сети с сервера, используя протоколы PXE, BOOTP, DHCP, TFTP и Remote Installation Services (RIS).

**ХОД РАБОТЫ**





**Листинг программы:**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Net.Sockets;

using System.Net;

using System.IO;

using System.IO.Compression;

namespace Laba2Server

{

class Program

{

static void Main(string[] args)

{

byte[] msg = null;

string sql1 = "SELECT \* FROM[Clients]";

string sql2 = "SELECT \* FROM[Holdings]";

string sql3 = "SELECT \* FROM[Master]";

string connection = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\vladi\OneDrive\\2017\\  2\Lab2\Laba2Server\Laba2Server\Database1.mdf;Integrated Security=True";

using(SqlConnection sqlConnection = new SqlConnection(connection))

{

sqlConnection.Open();

SqlDataAdapter adapter1 = new SqlDataAdapter(sql1, sqlConnection);

DataSet dataSet = new DataSet();

adapter1.Fill(dataSet, "Clients");

SqlDataAdapter adapter2 = new SqlDataAdapter(sql2, sqlConnection);

adapter2.Fill(dataSet, "Holdings");

SqlDataAdapter adapter3 = new SqlDataAdapter(sql3, sqlConnection);

adapter3.Fill(dataSet, "Master");

Byte[] data;

MemoryStream mem = new MemoryStream();

GZipStream zip = new GZipStream(mem, CompressionMode.Compress);

dataSet.WriteXml(zip, XmlWriteMode.WriteSchema);

zip.Close();

data = mem.ToArray();

mem.Close();

msg = data;

}

IPHostEntry ipHost = Dns.GetHostEntry("localhost");

IPAddress ipAddr = ipHost.AddressList[0];

IPEndPoint ipEndPoint = new IPEndPoint(ipAddr, 11000);

Socket sListener = new Socket(ipAddr.AddressFamily, SocketType.Stream, ProtocolType.Tcp);

try

{

sListener.Bind(ipEndPoint);

sListener.Listen(10);

while (true)

{

Socket handler = sListener.Accept();

handler.Send(msg);

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("  : {0}.", ipEndPoint);

handler.Shutdown(SocketShutdown.Both);

handler.Close();

Console.WriteLine(" !");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

}

finally

{

Console.ReadLine();

}

Console.ReadLine();

}

}

}

**ВЫВОД:**

Во время выполнения лабораторной работы мы закрепили знания о тонких клиентах, web-клиентах, тонких клиентах, работающих в терминальном режиме, а также научились создавать «тонкие» клиенты.