МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ижевский государственный технический университет имени М. Т. Калашникова»

Институт «Информатика и вычислительная техника»

Кафедра «Программное обеспечение»

Работа защищена с оценкой

«\_\_\_\_\_\_\_\_\_\_»

Дата \_\_\_\_\_\_\_\_\_\_\_\_

Подпись\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по дисциплине «Исследование операций. Теория принятия решений»

на тему: «ИС анализа клиентской базы CRM систем»

|  |  |
| --- | --- |
| Выполнил:  студент группы Б18-191-2 | Гумметов Р. А. |
| Руководитель:  Старший преподаватель | Лугачев П. П. |

Рецензия:

степень достижения поставленной цели работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

полнота разработки темы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

уровень самостоятельности работы обучающегося\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

недостатки работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

СОДЕРДАНИЕ

[1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ 3](#_Toc73529989)

[2. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ 4](#_Toc73529990)

[3. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ 8](#_Toc73529991)

[3.1 Входные данные 8](#_Toc73529992)

[3.2 Выходные данные 9](#_Toc73529993)

[4. СХЕМЫ ПРОГРАММЫ 10](#_Toc73529994)

[5. СХЕМА ОБРАБОТКИ ДАННЫХ 11](#_Toc73529995)

[6. ТЕСТОВЫЙ ПРИМЕР 13](#_Toc73529996)

[7. БЕТА ТЕСТИРОВАНИЕ 18](#_Toc73529997)

[8. ЗАКЛЮЧЕНИЕ 19](#_Toc73529998)

[9. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 20](#_Toc73529999)

[ПРИЛОЖЕНИЕ 1 21](#_Toc73530002)

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 21](#_Toc73530003)

[ПРИЛОЖЕНИЕ 2 23](#_Toc73530004)

[РУКОВОДСТВО ОПЕРАТОРА 23](#_Toc73530005)

[РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 24](#_Toc73530006)

[ПРИЛОЖЕНИЕ 3 28](#_Toc73530007)

[ТЕКСТ ПРОГРАММЫ 28](#_Toc73530008)

# 1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Реализовать информационную систему, анализирующую клиентскую базу CRM систем.

# 2. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

В бизнесе необходимость [автоматизации процессов](https://worksection.com/ua/blog/business-process.html) стала привычным явлением. Она важна и в мегапроектах, и при ведении малого, среднего бизнеса. CRM-система — мощный инструмент, который используется для контроля, планирования и аналитики работы с клиентами, в том числе, и через автоматизацию отдельных рабочих моментов.

В малом бизнесе еще встречается ведение истории звонков и контактов в Excel-таблицах или использование отдельных приложений для аналитики эффективности работы. Это примеры «ручного» CRM, который, конечно же, всухую проигрывает автоматизации бизнес-процессов. Поэтому на рынке за последнее десятилетие появилось огромное количество CRM — комплексных систем с модулями, направленными на выполнение различных задач, единым хранилищем, и возможностью подключения дополнительных сервисов.

Клиентская база — ценный актив любой компании. Полнота и актуальность сведений в ней позволяют менеджеру наладить эффективное взаимодействие с заказчиками и увеличить объем повторных продаж. Помочь в создании клиентской базы может [CRM-система](https://megaplan.ru/crm/) — программное обеспечение для учета, контроля и анализа информации о клиентах.

Для создания клиентской базы построим гипер-куб решений. Параметры, по которым будут приниматься решения:

1) Местонахождение клиента

2) Сегмент деятельности клиента

3) Статут клиента

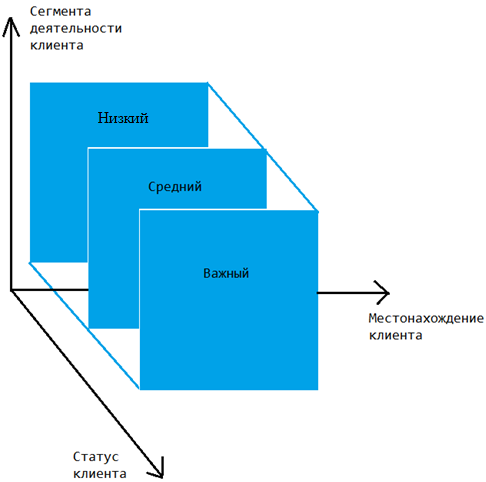


Рис. 1 - Гипер-куб решений

Гипер-куб можно вращать, а также делать изменения по измерениям.

Зачем нужна клиентская база

Специалисты выделяют пять основных преимуществ клиентской базы:

1. Анализ перспектив. Составление такой базы данных позволит оценить рынок по проценту дистрибьюции, принадлежащей вашей компании и лояльности клиентов к вашему продукту.
2. Сохранение данных. Клиентская база позволяет застраховаться руководителю компании от потери клиентов в случае увольнения торгового представителя или менеджера по продажам. Наличие подобной сводки позволит быстро и эффективно ввести в курс дела новых сотрудников.
3. Оценка клиентов и составление прогнозов. Клиентская база позволяет проводить более глубокий анализ вашего сегмента рынка. Вы можете узнать о личных предпочтениях своих клиентов, оценить их корзину покупки, частоту совершения заказов, а также их сезонность. Подобного рода анализ позволит проводить маркетинговые исследования и составлять прогнозы в отношении заказчиков и развития вашего бизнеса в целом.
4. Анализ других сегментов рынка с целью расширения компании. Клиенты зачастую интересуются другими товарами на рынке помимо вашей продукции. Благодаря сбору и обработке такой информации создаются перспективы для совершения попутных продаж или расширения бизнеса.
5. Индивидуализация подхода к каждому клиенту. Клиентская база помогает сегментировать заказчиков, изучить их потребности и подобрать личный подход в каждом отдельном случае.

Клиентская база также позволяет улучшить и оптимизировать работу предприятия благодаря функции обратной связи с клиентами. На основании клиентской базы возможно формирование системы поощрений и программ лояльности для клиентов. Знание персональных данных ваших заказчиков поможет поздравлять их с праздниками, организовывать специальные скидки и делать индивидуальные предложения тогда, когда это будет актуально для каждого конкретного человека. Подобный способ ведения бизнеса значительно стимулирует продажи и пользуется одобрением со стороны клиентов.

CRM система как способ ведения клиентской базы

#### На CRM стоит обратить внимание, потому что:

1. это мощный инструмент для привлечения новых и удержаний старых клиентов — недостаточно просто выпускать хороший продукт; в условиях жёсткой конкуренции важны высокие индексы LTV и ROI. Усилия в виде маркетинговых кампаний, наработки клиентской базы, найма опытных сотрудников сами по себе ничего не значат без достижения целей бизнеса. Для этого и существует автоматизированная система CRM;
2. CRM учит тому, как работать с клиентом — в системе заложен комплекс готовых инструментов, использование которых выводит работу на новый конкурентоспособный уровень. Например, интеграция CRM-системы и IP-телефонии обеспечит фиксацию звонков и новых контактов, анализ качества работы отдела продаж с лидами.
3. улучшенный опыт работы с клиентами — ключевое преимущество работы с CRM. Пользователи точнее и легче сегментируются по разным критериям, определяются их потребности, и благодаря этому компании взаимодействуют с каждым клиентом, превращая отношения из «компания» → «группа клиентов» в «компания» → «конкретный клиент».

Мы привыкли, что наши желания предугадываются, а постпродажное сопровождение доступно в любое время суток. Все это практически невозможно реализовать без CRM-системы, ведь в ней на каждого клиента заведена отдельная карточка с полной информацией о покупателе:

* контакты;
* данные о клиенте (марка авто, день рождения, наличие детей, предпочтения и т. д.);
* записи звонков и письма;
* файлы и документы;
* история покупок и платежей;
* примечания менеджеров и история задач.

Эти данные представлены в хронологии — можно за пару минут просмотреть всю историю работы с клиентом. Еще один плюс внедрения CRM в том, что база данных, наконец, станет единой.

# 3. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

## 3.1 Входные данные

Входными данными в программе являются сведения о клиентах в клиентской базе, находящиеся в xml файле ClientBase.xml:

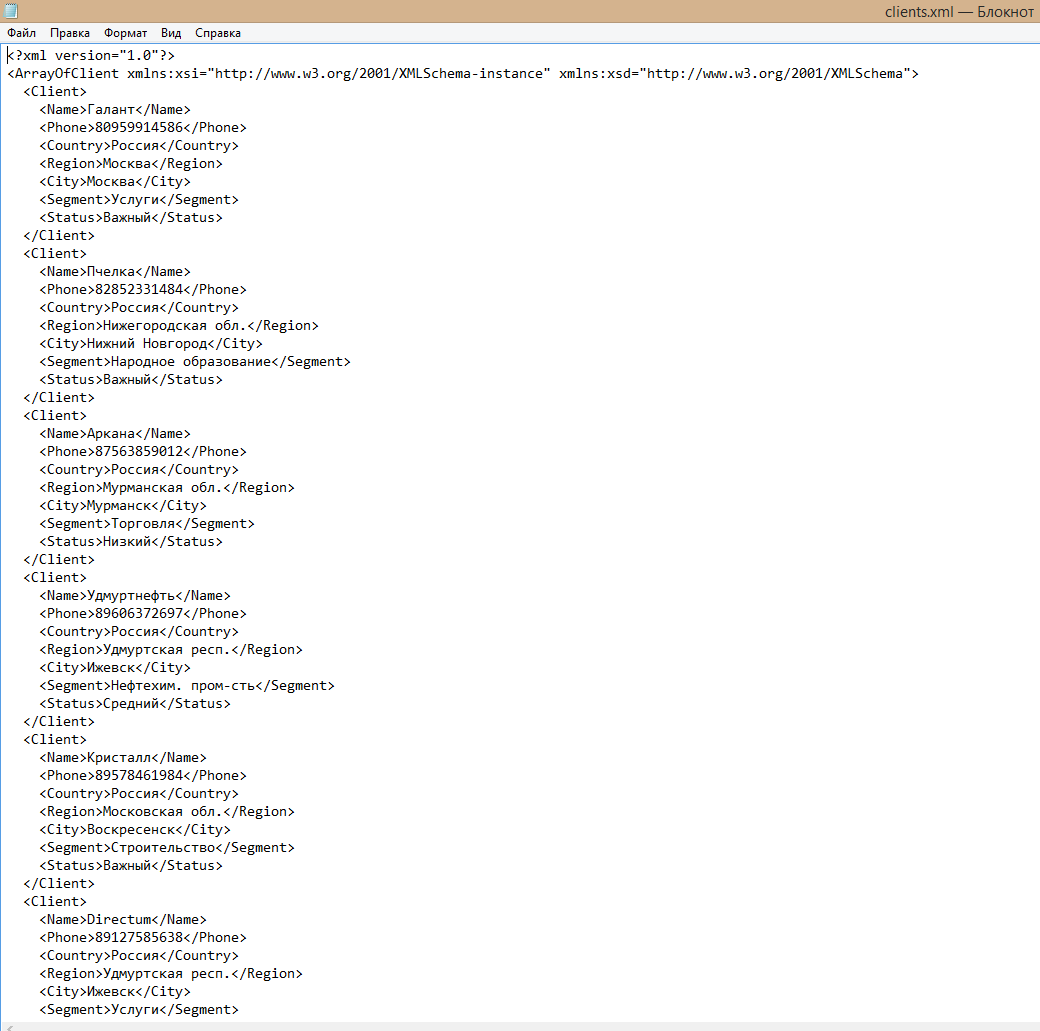


Рис. 2 - xml файл ClientBase.xml с данными о клиентах

Пользователь может добавлять в клиентскую базу новых клиентов, а также удалять клиентов из базы. Чтобы добавить нового клиента, нужно ввести название компании, телефон, страну, регион, город, сегмент деятельности и статус. Чтобы удалить клиента, нужно ввести название компании.

## 3.2 Выходные данные

По завершению работы программы обновленная клиентская база сохраняется в файл ClientBase.xml.

Также пользователь может делать выборку клиентов по определенным параметрам. Например, можно вывести только клиентов, имеющих одинаковое местонахождение, одинаковый сегмент деятельности или одинаковый статус.

# 4. СХЕМЫ ПРОГРАММЫ

На рис. 3 показана схема программы:

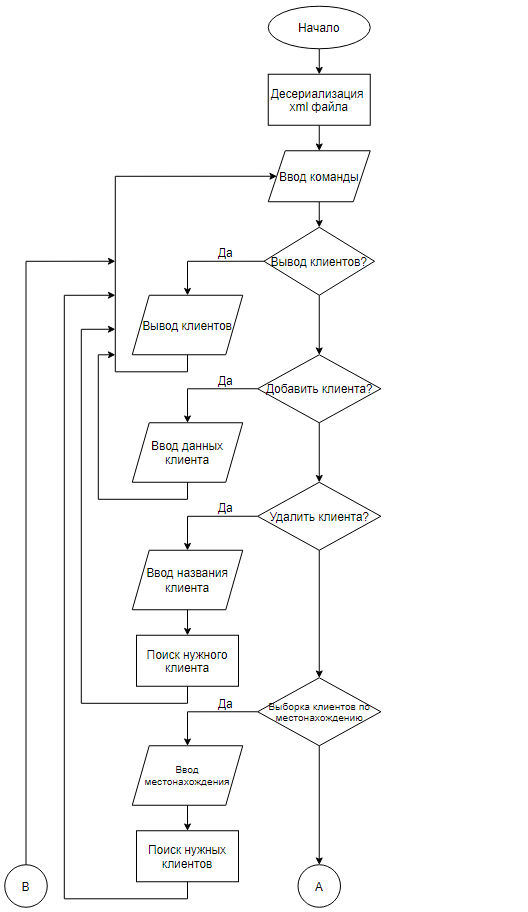


Рис. 3

# 

Рис.4

# 5. СХЕМА ОБРАБОТКИ ДАННЫХ

На рис. 4 показана схема обработки данных:

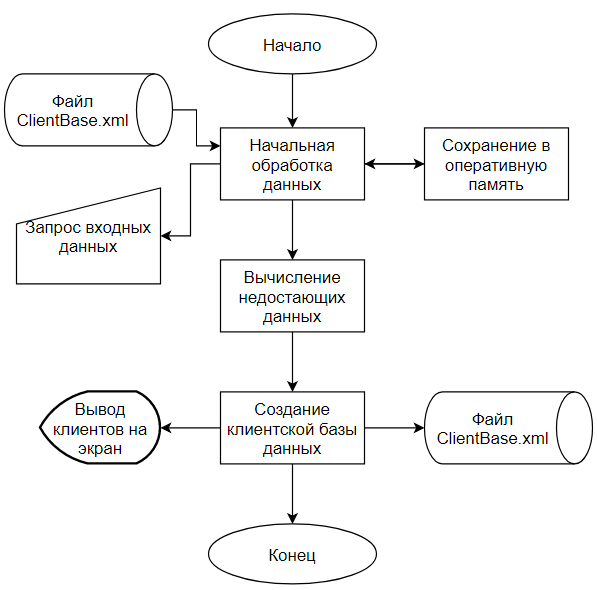


Рис. 5

Программа сохраняет начальные данные в оперативную память, после чего вычисляет недостающие данные. Далее создается клиентская база, которая записывается в xml файл ClientBase.xml, а также выводится на экран.

# 6. ТЕСТОВЫЙ ПРИМЕР

При запуске программа предоставляет пользователю меню и предлагает выбрать действие:

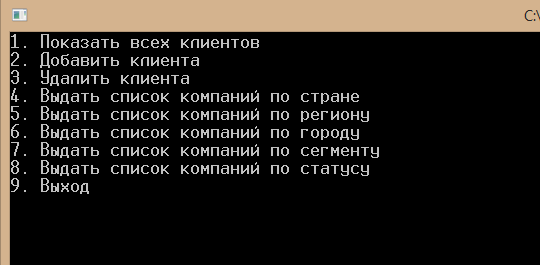


Рис. 6 - Меню программы

Выберем первый пункт, чтобы программа показала список всех клиентов:

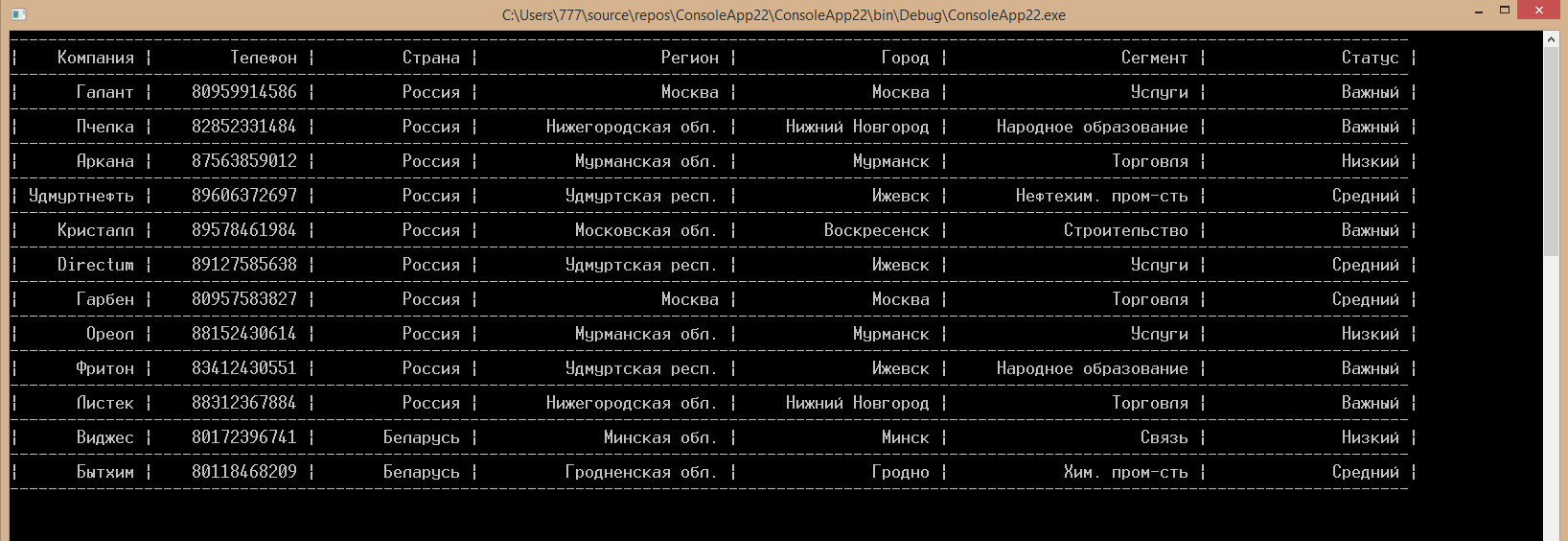


Рис. 7

Вернемся в меню и выберем второй пункт для добавления нового клиента. Программа запрашивает данные о клиенте:

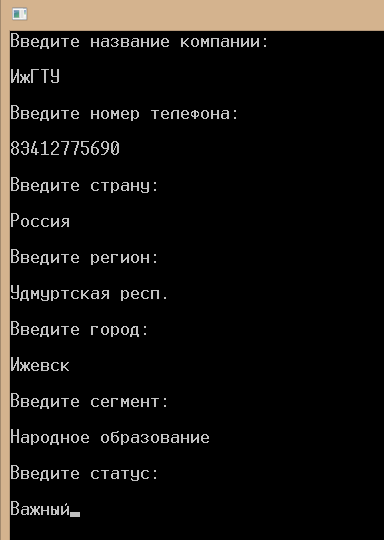


Рис. 8

Клиент добавился в список клиентов:

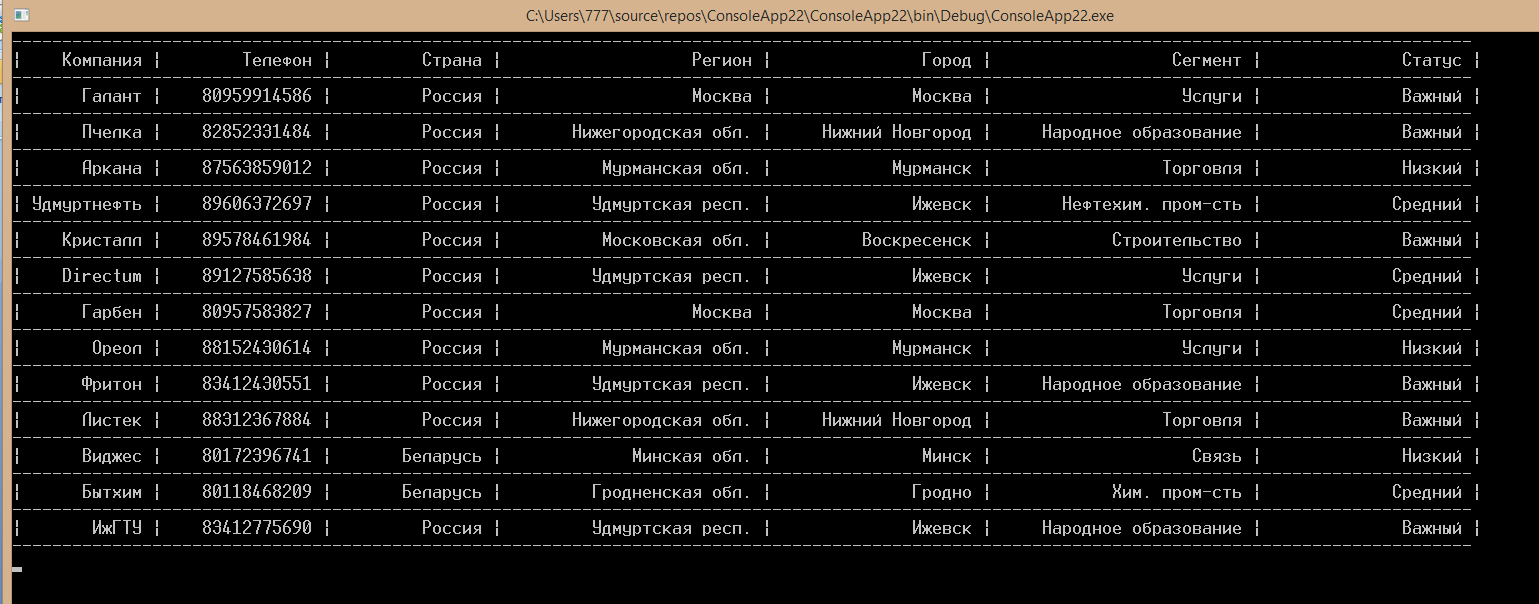


Рис. 9

Выберем пункт 3 и удалим клиента, которого только что добавили. Программа запрашивает название клиента, которого требуется удалить из клиентской базы:

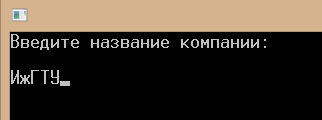


Рис. 10

Теперь данного клиента нет в списке клиентов:

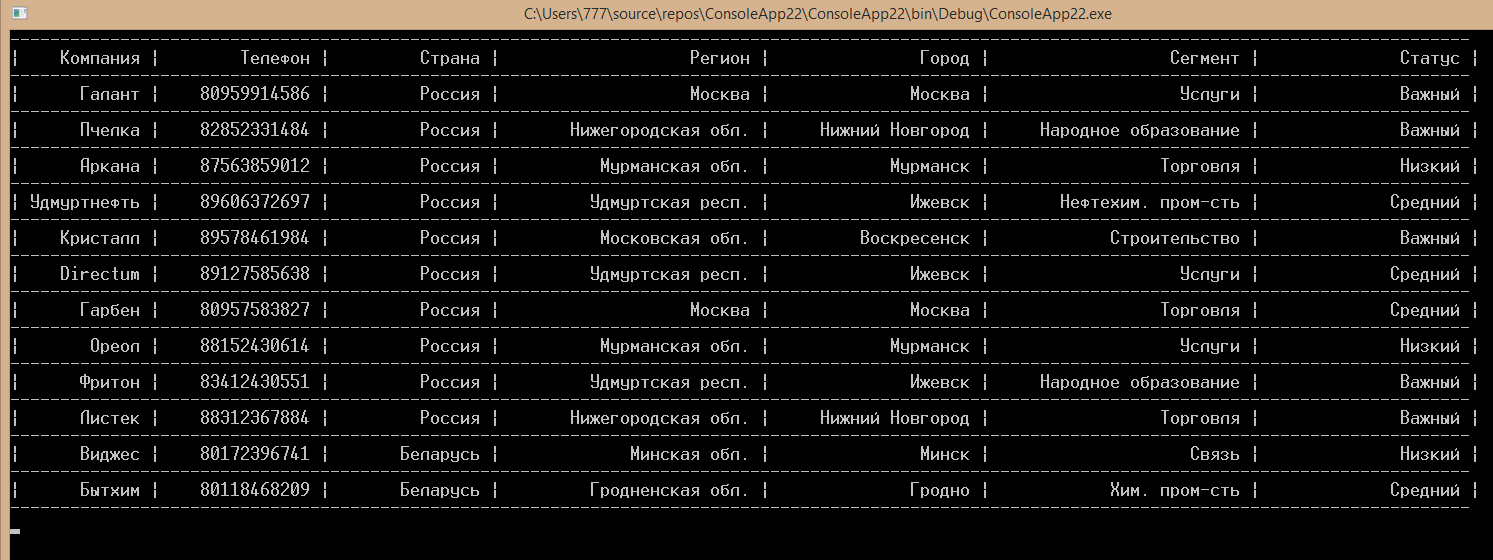


Рис. 11

Выберем пункт 4 и запросим у программы вывод всех клиентов из Беларуси:

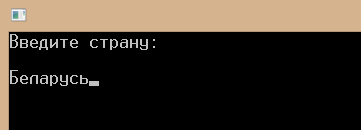


Рис. 12

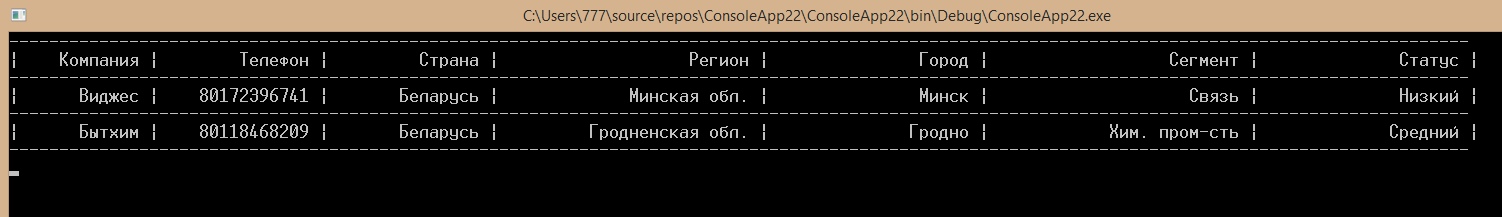


Рис. 13

Теперь выберем пункт 5 и запросим у программы вывод всех клиентов из Мурманской области:

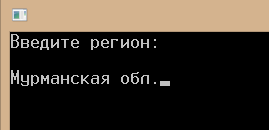


Рис. 14

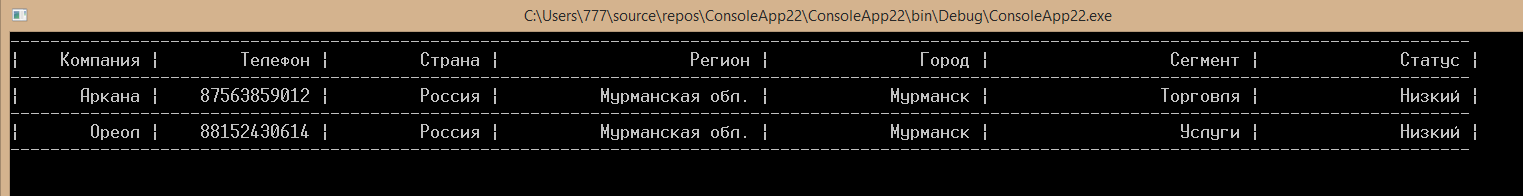


Рис. 15

Выберем пункт 6 и запросим у программы вывод всех клиентов из Ижевска:

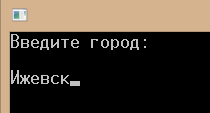


Рис. 16

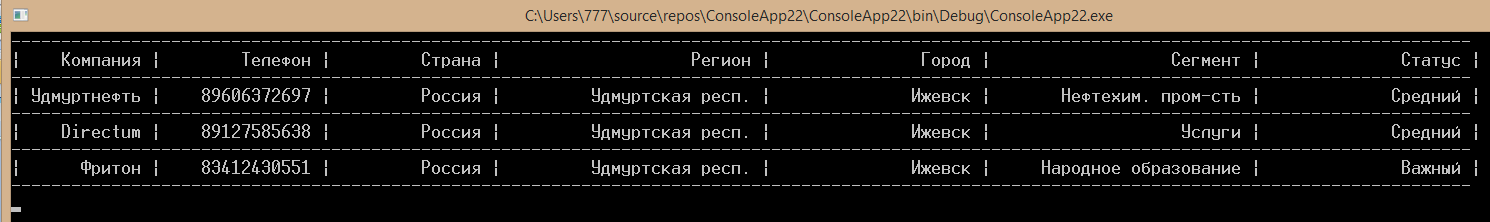


Рис. 17

Выберем пункт 7 и запросим у программы вывод всех клиентов из сегмента "Услуги":

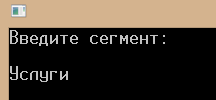


Рис. 18

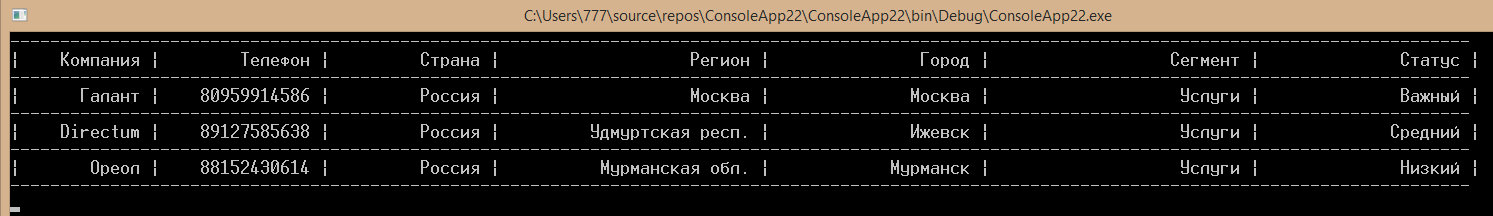


Рис. 19

Выберем пункт 8 и запросим у программы вывод всех клиентов со статусом "Важный":

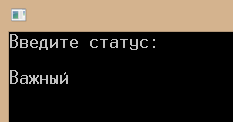


Рис. 20

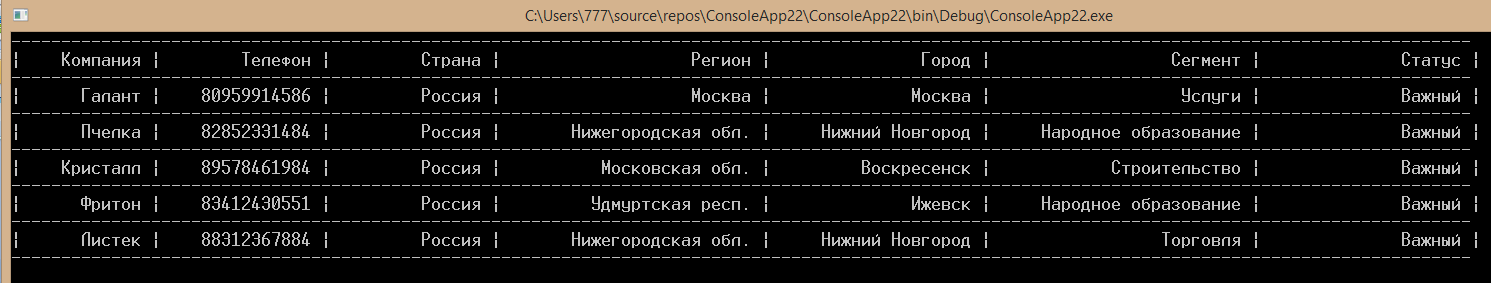


Рис. 21

# 7. БЕТА ТЕСТИРОВАНИЕ

Тестирование программы было произведено на различных ОС, ошибок выявлено не было. Главное неудобство программы в том, что если ввести слишком длинную строку в какой-либо параметр клиента, то строка сдвигается и таблица выглядит не очень красиво.

Программа предоставляет возможность ввода данных о клиентах, производит проверку на правильность ввода и вывода данных.

# 8. ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы изучены все преимущества CRM систем в работе с клиентскими базами, построен гипер-куб решений, а также реализована информационная система, анализирующая клиентскую базу CRM систем.

# 9. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Корнеев, А. М. Методы принятия решений : методические указания к проведению практических занятий по курсу «Теория принятия решений» / А. М. Корнеев. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2012. — 19 c. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/22892.html (29. 05. 2021). — Режим доступа: для авторизир. пользователей

2. Мхитарян, С. В. Системы управления взаимоотношениями с клиентами : учебное пособие / С. В. Мхитарян, М. В. Маркова. — Москва : Евразийский открытый институт, 2011. — 150 c. — ISBN 978-5-374-00518-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/10826.html (29. 05. 2021). — Режим доступа: для авторизир. пользователей

# 3. Ведение клиентской базы в CRM [электронный курс]. URL: https://megaplan.ru/news/articles/crm-klientskaya-baza/ (29. 05. 2021)

# 4. CRM и его виды, типы, предназначение [электронный курс]. URL: https://worksection.com/blog/crm-theory.html (29. 05. 2021)

5. Зачем нужна база клиентов? [электронный курс]. URL: https://alenashefina.com/zachem-nuzhna-baza-klientov/ (29. 05. 2021)

# ПРИЛОЖЕНИЕ 1

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1. Введение

В современном бизнесе необходимость автоматизации различных процессов стала уже привычным явлением. CRM-система — это — прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путем сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов.

2. Назначение

Программа предназначена для создания и анализа клиентской базы.

3. Требования к программе

3.1 Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения следующих функций:

* Возможность вывести клиентскую базу на экран
* Возможность добавить нового клиента в клиентскую базу
* Возможность удалить клиента из клиентской базы
* Возможность вывести список клиентов с заданными характеристиками на экран

3.2 Требования к надежности

Программа должна проверять правильность ввода и вывода данных

3.3 Требования к составу и параметрам технических средств

Система должна работать на IBM-совместимых персональных компьютерах.

Минимальная конфигурация:

* тип процессора Intel с частотой около 1000Гц и выше
* объем ОЗУ 512 Мб и более
* тип монитора VGA и выше

3.4 Требования к информационной и программной совместимости

Программа должна работать на всех ОС в приложении Visual Studio

4. Требования к программной документации

В состав сопровождающей документации должны входить:

1) расчетно-пояснительная записка;

2) руководство оператора;

3) руководство пользователя

# ПРИЛОЖЕНИЕ 2

# РУКОВОДСТВО ОПЕРАТОРА

1. Назначение программы

Программа написана на языке высокого уровня C# и предназначена для создания и анализа клиентской базы.

2. Условия применения программы

Программа работает на всех операционных системах в приложении Visual Studio. Для функционирования данной программы может быть использован IBM-совместимый персональный компьютер с процессором частотой 1000Гц или выше.

3. Характеристика программы

Объем исполнимого файла программы 154Kb.

Программа представляет собой 2 файла:

Program.cs - содержит все необходимые методы для работы программы;

Client.cs - содержит класс клиента с его характеристиками.

Программа сохраняет данные о клиентах в файл ClientBase.xml.

4. Обращение к программе

Для запуска программы необходимо запустить код файла program.cs.

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

После запуска программы появляется меню:

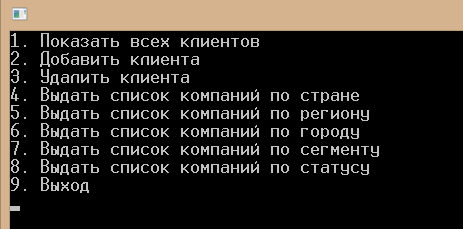


Рис. 22

После выбора пункта 1 показывает список клиентов:

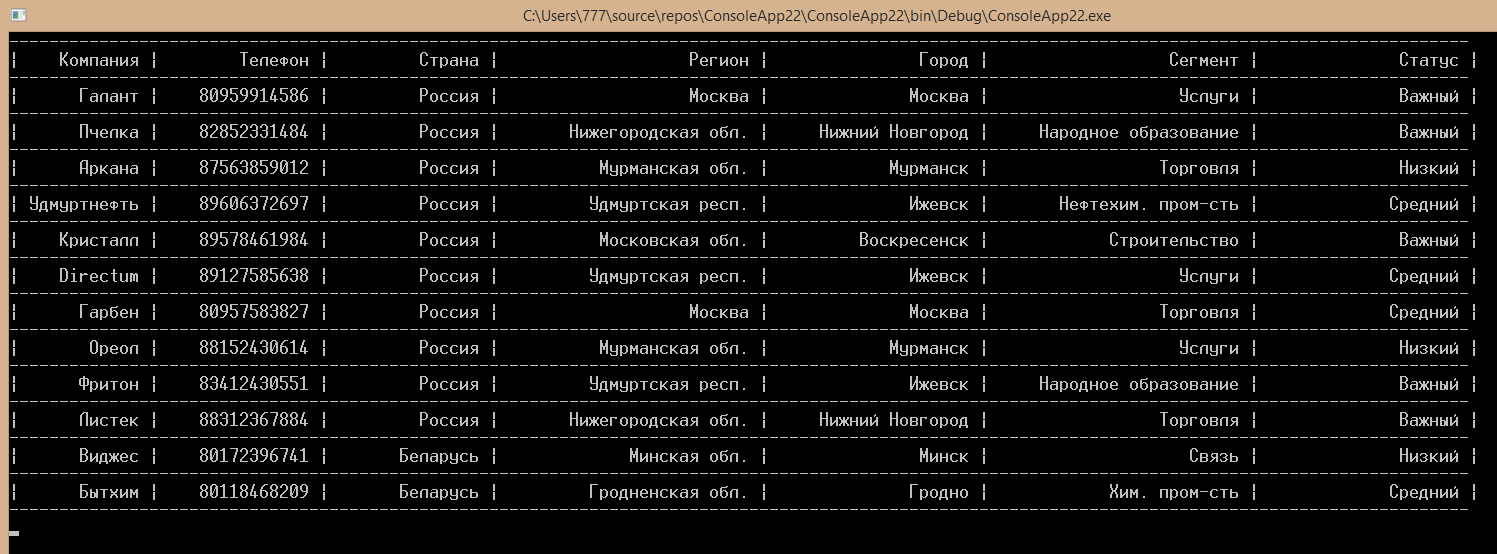


Рис. 23

После выбора пункта 2 просит ввести данные о клиенте:

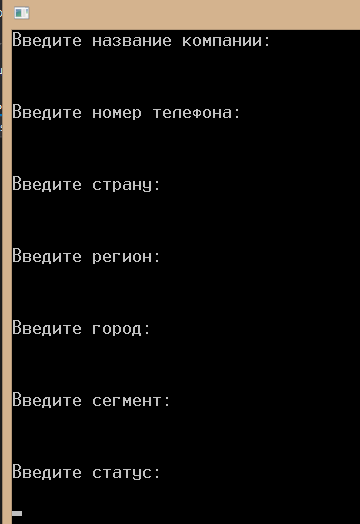


Рис .24

После выбора пункта 3 просит ввести название клиента, которого нужно удалить:

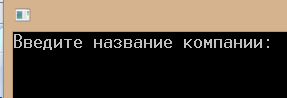


Рис. 25

После выбора пункта 4 просит ввести страну, по которой выдать клиентов:

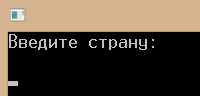


Рис. 26

После выбора пункта 5 просит ввести регион, по которому выдать клиентов:

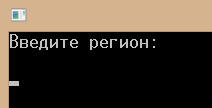


Рис. 27

После выбора пункта 6 просит ввести город, по которому выдать клиентов:

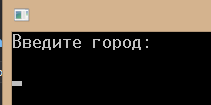


Рис. 28

После выбора пункта 7 просит ввести сегмент деятельности, по которому выдать клиентов:

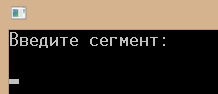


Рис. 29

После выбора пункта 8 просит ввести статус, по которому выдать клиентов:

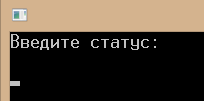


Рис. 30

Если клиентов по введенным параметрам нет, то программа сообщает об этом:

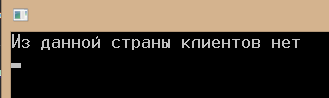


Рис. 31

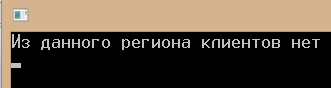


Рис. 32

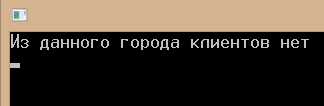


Рис. 33

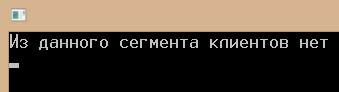


Рис. 34

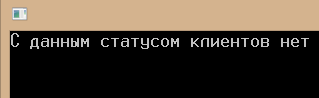


Рис. 35

После работы программы создается файл ClientBase.xml, в котором сохраняются данные о клиентах.



Рис. 36

# ПРИЛОЖЕНИЕ 3

# ТЕКСТ ПРОГРАММЫ

Файл Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Serialization;

using System.IO;

using System.Runtime.Serialization;

namespace ConsoleApp22

{

class Program

{

static XmlSerializer xmlFormatter = new XmlSerializer(typeof(List<Client>));

static public void Serialization()

{

string path = @"C:\Users\777\source\repos\ConsoleApp22\ConsoleApp22\bin\Debug\clients.xml";

FileInfo fileInf = new FileInfo(path);

fileInf.Delete();

using (var file = new FileStream("clients.xml", FileMode.OpenOrCreate))

{

xmlFormatter.Serialize(file, Client.Clients);

}

}

static public void Deserialization()

{

using (var file = new FileStream("clients.xml", FileMode.OpenOrCreate))

{

Client.Clients = xmlFormatter.Deserialize(file) as List<Client>;

}

}

static public void CreateClient()

{

Console.Clear();

Console.WriteLine("Введите название компании:\n");

string name = Console.ReadLine();

Console.WriteLine("\nВведите номер телефона:\n");

string phone = Console.ReadLine();

Console.WriteLine("\nВведите страну:\n");

string country = Console.ReadLine();

Console.WriteLine("\nВведите регион:\n");

string region = Console.ReadLine();

Console.WriteLine("\nВведите город:\n");

string city = Console.ReadLine();

Console.WriteLine("\nВведите сегмент:\n");

string segment = Console.ReadLine();

Console.WriteLine("\nВведите статус:\n");

string status = Console.ReadLine();

Client client = new Client()

{

Name = name,

Phone = phone,

Country = country,

Region = region,

City = city,

Segment = segment,

Status = status

};

Client.Clients.Add(client);

Console.Clear();

Console.WriteLine("Клиент успешно добавлен");

}

static public void Menu()

{

Console.Clear();

Console.WriteLine("1. Показать всех клиентов");

Console.WriteLine("2. Добавить клиента");

Console.WriteLine("3. Удалить клиента");

Console.WriteLine("4. Выдать список компаний по стране");

Console.WriteLine("5. Выдать список компаний по региону");

Console.WriteLine("6. Выдать список компаний по городу");

Console.WriteLine("7. Выдать список компаний по сегменту");

Console.WriteLine("8. Выдать список компаний по статусу");

Console.WriteLine("9. Выход");

int command = Convert.ToInt32(Console.ReadLine());

switch (command)

{

case 1:

{

ShowClients();

Menu();

break;

}

case 2:

{

CreateClient();

Menu();

break;

}

case 3:

{

DeleteClient();

Menu();

break;

}

case 4:

{

ShowCountries();

Menu();

break;

}

case 5:

{

ShowRegions();

Menu();

break;

}

case 6:

{

ShowCities();

Menu();

break;

}

case 7:

{

ShowSegments();

Menu();

break;

}

case 8:

{

ShowStatuses();

Menu();

break;

}

case 9:

{

break;

}

}

}

static public void ShowClients()

{

Console.Clear();

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", "Компания", "Телефон", "Страна", "Регион", "Город", "Сегмент", "Статус");

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

for (int i = 0; i < Client.Clients.Count; i++)

{

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", Client.Clients[i].Name, Client.Clients[i].Phone, Client.Clients[i].Country, Client.Clients[i].Region, Client.Clients[i].City, Client.Clients[i].Segment, Client.Clients[i].Status);

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

}

Console.ReadLine();

}

static public void ShowCountries()

{

Console.Clear();

Console.WriteLine("Введите страну:\n");

string country = Console.ReadLine();

bool found = false;

Console.Clear();

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", "Компания", "Телефон", "Страна", "Регион", "Город", "Сегмент", "Статус");

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

for (int i = 0; i < Client.Clients.Count; i++)

{

if (Client.Clients[i].Country == country)

{

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", Client.Clients[i].Name, Client.Clients[i].Phone, Client.Clients[i].Country, Client.Clients[i].Region, Client.Clients[i].City, Client.Clients[i].Segment, Client.Clients[i].Status);

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

found = true;

}

}

if (!found)

{

Console.Clear();

Console.WriteLine("Из данной страны клиентов нет");

}

Console.ReadLine();

}

static public void ShowRegions()

{

Console.Clear();

Console.WriteLine("Введите регион:\n");

string region = Console.ReadLine();

bool found = false;

Console.Clear();

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", "Компания", "Телефон", "Страна", "Регион", "Город", "Сегмент", "Статус");

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

for (int i = 0; i < Client.Clients.Count; i++)

{

if (Client.Clients[i].Region == region)

{

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", Client.Clients[i].Name, Client.Clients[i].Phone, Client.Clients[i].Country, Client.Clients[i].Region, Client.Clients[i].City, Client.Clients[i].Segment, Client.Clients[i].Status);

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

found = true;

}

}

if (!found)

{

Console.Clear();

Console.WriteLine("Из данного региона клиентов нет");

}

Console.ReadLine();

}

static public void ShowCities()

{

Console.Clear();

Console.WriteLine("Введите город:\n");

string city = Console.ReadLine();

bool found = false;

Console.Clear();

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", "Компания", "Телефон", "Страна", "Регион", "Город", "Сегмент", "Статус");

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

for (int i = 0; i < Client.Clients.Count; i++)

{

if (Client.Clients[i].City == city)

{

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", Client.Clients[i].Name, Client.Clients[i].Phone, Client.Clients[i].Country, Client.Clients[i].Region, Client.Clients[i].City, Client.Clients[i].Segment, Client.Clients[i].Status);

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

found = true;

}

}

if (!found)

{

Console.Clear();

Console.WriteLine("Из данного города клиентов нет");

}

Console.ReadLine();

}

static public void ShowSegments()

{

Console.Clear();

Console.WriteLine("Введите сегмент:\n");

string segments = Console.ReadLine();

bool found = false;

Console.Clear();

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", "Компания", "Телефон", "Страна", "Регион", "Город", "Сегмент", "Статус");

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

for (int i = 0; i < Client.Clients.Count; i++)

{

if (Client.Clients[i].Segment == segments)

{

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", Client.Clients[i].Name, Client.Clients[i].Phone, Client.Clients[i].Country, Client.Clients[i].Region, Client.Clients[i].City, Client.Clients[i].Segment, Client.Clients[i].Status);

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

found = true;

}

}

if (!found)

{

Console.Clear();

Console.WriteLine("Из данного сегмента клиентов нет");

}

Console.ReadLine();

}

static public void ShowStatuses()

{

Console.Clear();

Console.WriteLine("Введите статус:\n");

string status = Console.ReadLine();

bool found = false;

Console.Clear();

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", "Компания", "Телефон", "Страна", "Регион", "Город", "Сегмент", "Статус");

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

for (int i = 0; i < Client.Clients.Count; i++)

{

if (Client.Clients[i].Status == status)

{

Console.WriteLine("|{0,12} |{1,15} |{2,15} |{3,25} |{4,20} |{5,25} |{6,20} |", Client.Clients[i].Name, Client.Clients[i].Phone, Client.Clients[i].Country, Client.Clients[i].Region, Client.Clients[i].City, Client.Clients[i].Segment, Client.Clients[i].Status);

Console.WriteLine("--------------------------------------------------------------------------------------------------------------------------------------------------");

found = true;

}

}

if (!found)

{

Console.Clear();

Console.WriteLine("С данным статусом клиентов нет");

}

Console.ReadLine();

}

static public void DeleteClient()

{

Console.Clear();

Console.WriteLine("Введите название компании:\n");

string name = Console.ReadLine();

bool found = false;

for (int i = 0; i < Client.Clients.Count; i++)

{

if (Client.Clients[i].Name == name)

{

found = true;

Client.Clients.RemoveAt(i);

break;

}

}

Console.Clear();

if (found)

Console.WriteLine("Клиент успешно удален");

else

Console.WriteLine("В клиентской базе нет компании с таким названием");

Console.ReadLine();

}

static void Main(string[] args)

{

Deserialization();

Menu();

Serialization();

}

}

}

Файл Client.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Serialization;

using System.IO;

using System.Runtime.Serialization;

namespace ConsoleApp22

{

[Serializable]

public class Client

{

public static List<Client> Clients = new List<Client>();

public string Name { get; set; }

public string Phone { get; set; }

public string Country { get; set; }

public string Region { get; set; }

public string City { get; set; }

public string Segment { get; set; }

public string Status { get; set; }

}

}