

метром от 400 до 700 метров с центром в полученных в п. 1.2 точках (рис. 2).

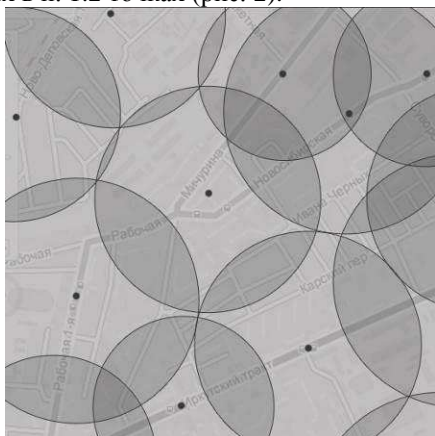


Рис. 2. Пример зонирования города

1.4 Находим список жилых домов, расположенных в полученных зонах также путем отправления HTTP-запросов в формате: адрес дома, этажность, координаты данного дома.

1.5 Вычисляем площадь каждого жилого дома с учетом его этажности. На основе этой информации будем вычислять среднее количество людей, проживающих в доме.

1.6 Производим расчет среднего количества проживающих в найденных домах людей. Согласно данным Росстата на 2013 год на одного человека в РФ приходится 24,4 м<sup>2</sup> жилой площади. Таким образом, делим общую площадь дома, полученную в п. 1.5 на среднюю величину жилой площади, приходящейся на одного человека в РФ.

1.7 Рассчитываем количество людей, вероятно использующих общественный транспорт. Для этого из общего количества людей в каждой зоне исключаем пользующихся личным автомобилем и детей младше 16 лет. Согласно данным Росстата на 2013 год процент населения, не имеющего в личном пользовании автомобиль, составляет 75 %; процент взрослого населения (лиц старше 16 лет) от общего числа жителей РФ – 84 %.

Таким образом, полученные данные о количестве проживающих людей в каждой зоне и потен-

циально использующих общественный транспорт, и является массивом отправок, элементами которого является количество людей, которым необходимо уехать с данного остановочного пункта.

Предлагаемый алгоритм позволяет существенно снизить трудозатраты на получение подобных данных и сократить количество ошибок, возникающих в результате человеческого фактора.

### Заключение

Результатом работы является программное приложение, реализующее предлагаемый алгоритм нахождения массива отправок пассажиров на основе данных свободной электронной карты OpenStreetMap.org.

Данное приложение в первую очередь направлено на получение массива пассажирских отправок с использованием современных информационных средств получения и обработки данных. Разработанное программное приложение позволяет построить массив для любой местности, в зависимости от введенных значений долготы и широты.

Также благодаря выполнению работы, были внесены новые данные по объектам для карты Города Томска, что позволило сделать карту ещё более актуальной и полезной.

### Литература

1. Понятие о пассажиропотоках. Изложить цели и методы их изучения [Электронный ресурс]. – Режим доступа: <http://www.transportguide.ru>, свободный.

2. Методика определения пассажиропотоков [Электронный ресурс]. – Режим доступа: <http://gorodtransport.ru>, свободный.

3. Общие сведения о географических картах. Типы географических карт. [Электронный ресурс]. – Режим доступа: <http://www.bygeo.ru>, свободный.

4. Электронная карта OpenStreetMap.org [Электронный ресурс]. – Режим доступа: <http://www.openstreetmap.org>, свободный.

## СРЕДСТВА ПОСТРОЕНИЯ РАСШИРЕНИЙ ДЛЯ ПАКЕТА MICROSOFT OFFICE

Верхотурова А.Э.

Томский политехнический университет  
634050, Россия, г. Томск, пр-т Ленина, 30  
E-mail: [verkhoturouva@gmail.com](mailto:verkhoturouva@gmail.com)

### Введение

Microsoft Office – наиболее распространённый офисный пакет приложений для работы с различными типами документов является открытой расширяемой платформой, использующей преимущества мощной коллекции интегрированных программ, локальных серверов, служб, инструментов и платформенных технологий [1]. Благодаря ис-

пользованию Office в качестве интерфейса для создания собственных решений разработчики могут воспользоваться преимуществами хорошо знакомых пользователям интерфейсов системы Microsoft Office и таких средств, как обработка текстов в программе Word, функции анализа данных программы Excel или функции управления электронной почтой программы Outlook.

Для настройки приложений Office и добавления новых функциональных возможностей, необходимых для решения различных задач, можно разрабатывать расширения с использованием нескольких технологий поддерживаемых Microsoft.

### Платформа Visual Basic for Applications

Эта мощная технология является наиболее полезной при необходимости автоматизации повторяющихся действий, таких как форматирование частей документа, внесение редакторских правок и т.д.

Так же можно расширить возможности взаимодействия с пользователем: уведомлять о событиях, или о необходимости выполнить определенное действие.

При работе с VBA используется объектная модель приложения Office представляющая собой систему классов, обеспечивающих определенную функциональность с помощью свойств и методов. Объектные модели для разных приложений Office отличаются. К примеру, в приложении Word объектом верхнего уровня является Application, содержащий коллекцию объектов Document, у которых в свою очередь есть коллекция объектов Paragraph. Ко всем объектам можно обращаться, изменять их свойства и инициировать связанные с ними события [2].

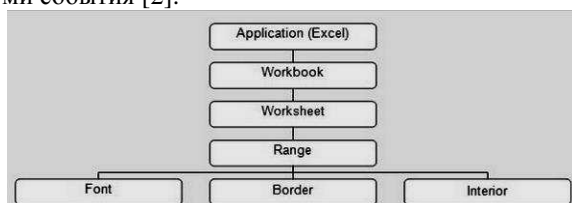


Рис. 1. Объектная модель Excel

Пример разработки в среде VBA – надстройка (макрос) обрабатывающая файл, в соответствии с форматом приказа и отправляющая его на сервер в виде xml-файла.



Рис. 2. Макрос сохранение приказов в базу

Для того чтобы записать значение в приказ необходимо выделить текст, относящийся к какому-либо значению и нажать на соответствующую кнопку.

Так же в VBA имеется инструмент построения форм пользовательского интерфейса, что облегчает взаимодействие пользователя с надстройками.



Рис. 3. Форма пользовательского интерфейса

Макросы сохраняются в файле с расширением .dotm и запускаются через панель “Разработчик”.

### Платформа Visual Studio Tools for Office

Если требуется создать решение, глубоко интегрируемое с Office и использующее полную функциональность платформы .NET Framework, надстройки (add-ins) могут быть хорошим выходом. Надстройки выполняются при запуске приложения Office, например Excel, и могут работать с любым документом, открытым пользователем.

Платформа разработки решений для Office позволяет создавать два типа надстроек: надстройки, связанные с конкретным документом, и надстройки расширяющие функциональность всего приложения.

При построении надстройки уровня приложения с помощью средств VSTO создается управляемый код сборки, загружаемый приложением Microsoft Office. После загрузки сборки, надстройка может инициировать обработку событий, возникающих в приложении (например, выбор пункта меню).

В надстройке также поддерживается обращение к объектной модели и любым классам .NET Framework, что позволяет автоматизировать и расширить функциональные возможности приложения. Для того чтобы обратиться к объектной модели необходимо установить на компьютере Primary Interop Assemblies – основные сборки взаимодействия. Они позволяют управляемому коду в решении взаимодействовать с объектной моделью приложения Office. Например, чтобы обратиться к объекту Application приложения Word необходимо будет использовать ссылку Microsoft.Office.Interop.Word.Application.

При создании проекта решения для Office в visual studio автоматически создается один или несколько классов, которые предоставляют точку входа для кода. В проектах уровня приложения это класс ThisAddIn, представляющий саму надстройку, а также элементы, которые можно использовать для доступа к объектной модели основного приложения и к другим функциям, которые доступны для надстроек.

Все созданные классы в проектах надстроек содержат обработчики событий Startup и

Shutdown. Для инициализации надстройки можно добавить код в обработчик событий Startup. Для очистки ресурсов, используемых надстройкой, можно добавить код в обработчик событий Shutdown [4].

Средства VSTO поддерживает несколько способов работы с графическим интерфейсом:

- построение пользовательского интерфейса с помощью форм WindowsForms;
- построение пользовательского интерфейса с помощью ленты Ribbon UI;
- построение пользовательского интерфейса с помощью настраиваемой панели задач custom task pane.

Пример надстройки в Office Excel –надстройка, автоматически выполняющая соединение 2 таблиц из разных файлов в одну, по значению одного из столбцов. Пользовательские элементы управления (user controls) расположены на ленте.

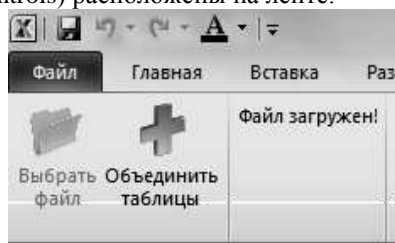


Рис. 4. Интерфейс надстройки для программы Excel

Для развертывания решений, созданных с помощью средств разработки Office, предоставляемых Visual Studio, можно использовать технологию ClickOnce или установщик Windows [5]. Для данной надстройки был разработан установщик с использованием библиотеки Windows Installer 3.1.

## Заключение

Надстройки и макросы в Office дают большие возможности для решения самых разнообразных задач от простой автоматизации рутинных действий до улучшения бизнес-процессов предприятия. Платформа VBA по-прежнему актуальна, с помощью нее легче выполнить задачи записи макросов или настройки документа, однако она тесно интегрирована только с приложениями Office и сильно ограничена в области безопасности и развертывания на предприятии.

Средства visual studio для Office в свою очередь предоставляют доступ не только к объектной модели Office, но и к .NET Framework, обеспечивают высокий уровень поддержки написания кода, безопасность использования надстроек, и масштабируемость в условиях предприятия.

## Литература

1. OpenOffice.org vs. Microsoft Office vs. Moore's Law [Электронный ресурс]. – Режим доступа: <http://www.oooninja.com/2008/05/openofficeorg-microsoft-office-moores.html>, свободный.
2. VBA Programming [Электронный ресурс]. Режим доступа: [http://msdn.microsoft.com/en-us/library/ee814735\(v=office.14\).aspx#odc\\_office14\\_AddingVBAProgrammingToYourOfficeToolkit\\_VBAProgrammingInOffice](http://msdn.microsoft.com/en-us/library/ee814735(v=office.14).aspx#odc_office14_AddingVBAProgrammingToYourOfficeToolkit_VBAProgrammingInOffice), свободный.
3. Общие сведения о разработке решений Office [Электронный ресурс]. Режим доступа: <http://msdn.microsoft.com/ru-ru/library/hy7c6z9k.aspx>
4. Visual Studio Tools for Office 2007: VSTO for Excel, Word, and Outlook Eric Carter, Eric Lippert. – изд. «Pearson Education» 2009 г.-1120с.
5. От VBA к VSTO – Dr. Gerard M. Verschuuren изд. «Holy Macro! Books» .2006 г. – 206с.

## СОВРЕМЕННЫЕ СТАНДАРТЫ ПОСТРОЕНИЯ СЕРВЕРА

Видяев И.Г.

Томский политехнический университет  
634050, Россия, г. Томск, пр-т Ленина, 30  
E-mail: vig@tpu.ru

### Введение

Широкое развитие в последнее время получила технология кластеризации серверов. Кластер – это объединение серверов в группу, которая с точки зрения пользователя воспринимается как единый сервер. То есть запросы пользователя поступают в единую точку входа, а дальше кластер решает – какой конкретно физический сервер будет его обрабатывать. Кластеризация серверов обычно преследует две цели – повышение надежности (отказоустойчивости) и повышение вычислительной мощности (или балансировка нагрузки) сервера.

Отказоустойчивость достигается за счет того, что при отказе аппаратных или программных компонент одного сервера, работающие процессы

без приостановки работы автоматически «переползают» с него на работоспособные серверы. После чего администраторы могут приступить к устранению неисправности, даже отключив сервер – пользователи ничего не заметят. Отказоустойчивое кластерное решение более ресурсоэффективно, чем обычные сервера, за счет того, что кластер использует совместно дисковые массивы и ленточные хранилища – отдельному серверу не требуются собственные большие жесткие диски. Таким образом уменьшается совокупная стоимость владения и потребляемая системой электроэнергия. Так же общее дисковое или ленточное хранилище более равномерно загружается, более легко резервируется, и поддерживает «горячую»