

УДК 004.413

Единый планировщик событий для Windows, OS X, Android

Гурин В.И., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Автономные информационные и управляющие устройства»*

*Научный руководитель: Вишневская Т.И. к.ф.-м.н.,
доцент кафедры «Программное обеспечение ЭВМ и информационные технологии»
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
bauman@bmstu.ru*

1. Введение

При создании нативных (родных) приложений для современных компьютеров и мобильных устройств используются среды разработки и языки программирования, предназначенные для целевой платформы. Возможность использования приложения на различных платформах порождает проблему переноса его с одной платформы на другую платформу. Конечно, существует достаточное количество способов портирования приложений, но всё они предполагают, может и не существенное, но в любом случае изменение кода. Главной же нашей задачей является создание приложения, использующего один исходный код, который одинаково функционирует на разных платформах.

В работе рассмотрено создание нативного приложения Единый планировщик событий для Windows, OS X и Android. Данное приложение – это модельный пример программы – органайзер [1] для описания этапов и особенностей разработки кроссплатформенного приложения в инструментальной среде Embarcadero Rad Studio XE7 [2].

2. Описание приложения Единый планировщик событий

Назначение приложения: приложение предназначено для своевременного оповещения пользователя о наступлении события из заранее подготовленного списка событий.

Входные и выходные данные приложения.

Событие - это заметка или напоминание. Оно содержит следующую информацию: тему, описание события, дату и время создания, а также дату и время наступления события, устанавливаемую пользователем. Если событие содержит время для наступления события, то будем называть его напоминанием, в противном случае заметкой. Напоминание может быть одиночным или повторяющимся. Для повторяющегося напоминания можно

настроить период, через который будет происходить извещение о событии. В качестве извещения, при наступлении события, проигрывается несколько звуковых сигналов и выводится сообщение, содержащее информацию о событии.

Функциональные возможности приложению [3]: Приложение Единый планировщик событий позволяет пользователю:

- создавать заметки и напоминания;
- редактировать события;
- удалять события;
- контролировать время наступления события и своевременно оповещать об этом пользователя.

Особенности реализации приложения:

- приложение разработано в инструментальной среде Embarcadero Rad Studio XE7;
- событие представлено в виде структуры данных запись;
- данные о списке событий (записей) записываются и хранятся в текстовом файле;
- приложение имеет один исходный код, одинаково функционирующий на платформах Windows, OS X и Android .

Интерфейс приложения имеет главную форму с меню и списком событий (Рис.1) и дополнительную форму для ввода данных о событиях (Рис. 2).

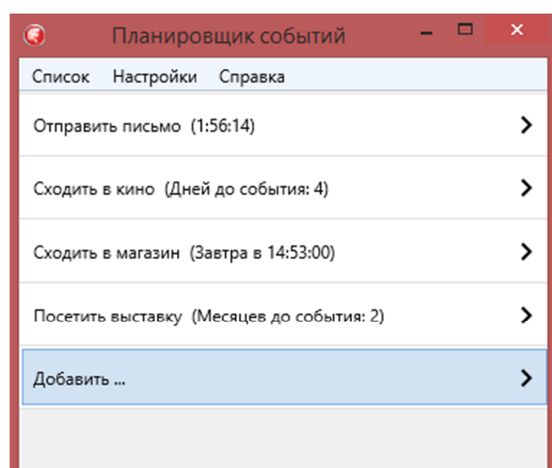


Рис. 1. Окно приложения в Windows

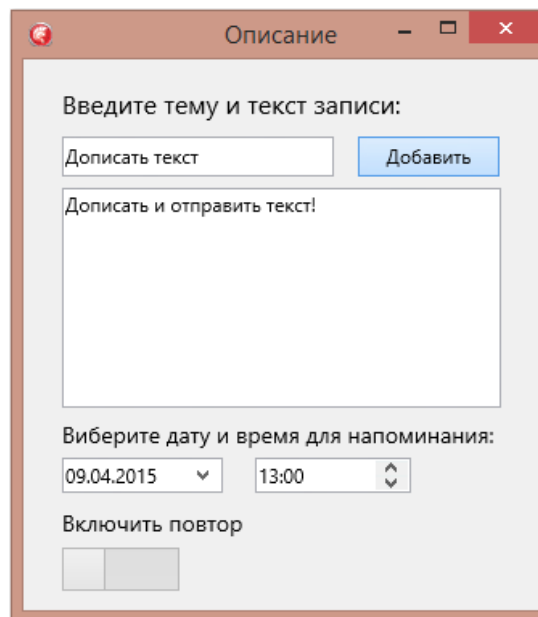


Рис. 2. Окно ввода данных о событии

В приложении есть экран настроек, в котором можно устанавливать параметры отображения, например, длину списка записей, тип извещения о событии, время за которое будет происходить напоминание, разрешение на удаление прошедших событий (Рис.3).

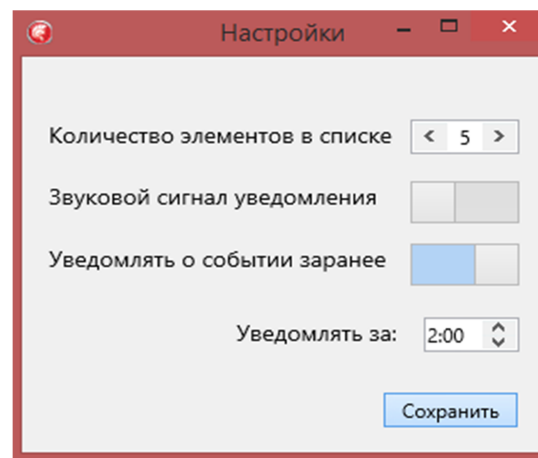


Рис. 3. Окно настройки приложения

3. Разработка Windows приложения

Для разработки приложения Единый планировщик событий для Windows, OS X и Android использовали библиотеку FireMonkey в инструментальной среде Embarcadero Rad Studio XE7 [4]. Кроссплатформенные классы этой библиотеки позволили нам создать приложения для платформы Windows и затем откомпилировать их на другие платформы.

Меню приложения реализовано классом TMenuBar. Для ввода текстовой информации (темы и текста сообщения) применили классы TEdit и TMemo (Рис. 4) . Для задания даты и времени напоминания использовали компоненты TDateEdit и TTimeEdit. Дату и время создания или изменения записи пользователь не вводит, в этом нет необходимости, так как считываются системные дата и время.

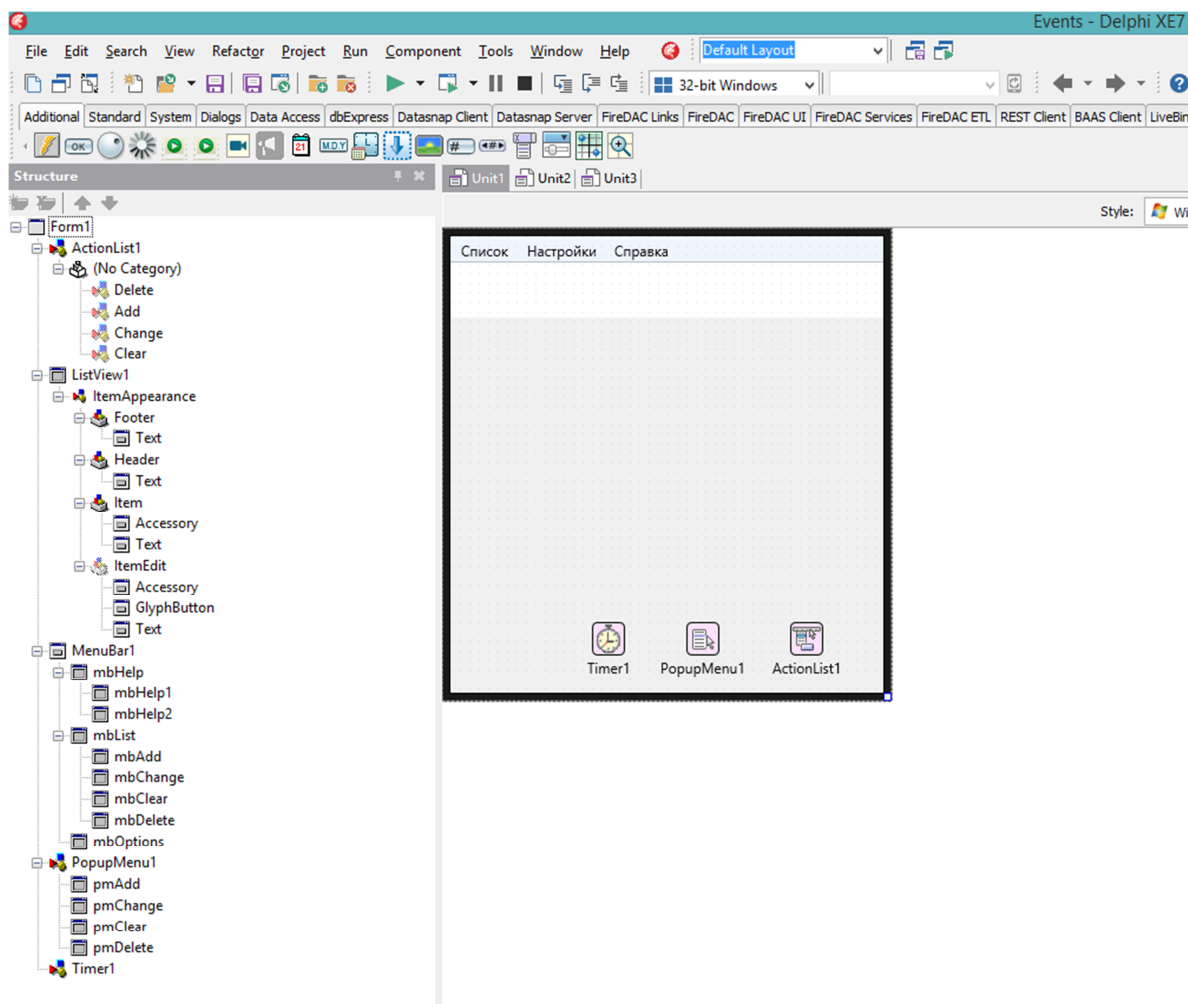


Рис. 4. Визуальная среда разработки Delphi XE7. Главная форма приложения.

Данные о списке событий сохраняются в текстовом файле. Рассматривался вопрос о хранении записей в типизированном файле. Для Windows приложений это наиболее эффективно. Но при создании релиза для Android OS возникала проблема, заключающаяся в том, что компилятор выделял ошибку в объявлении типа записей, в том месте, где элементам типа String указывается размерность. А если её не указывать, то не удастся записать данные в файл. Проблема с типизированными файлами возникала и на Mac OS, но уже другого характера, файл создавался, но в него не удавалось записывать информацию. Кроме этого, в приложении предусмотрена проверка наличия файла перед работой с ним,

и если же его нет, то он создаётся. Но Android OS вводит некоторые ограничения, которые не позволяют нам просто так создать файл, и при каждом обращении к нему будет всплывать сообщение об ошибке. Решить данную проблему можно таким способом. Мы не будем пытаться создать что-либо. Так необходимый нам файл мы включим в ресурсы программы. Это делается в окне Deployment, которое открывается с панели Project. Далее следует указать путь (remote path): assets/internal. И теперь программно обратиться к файлу можно присвоив его пути строку: `TPath.GetDocumentsPath + PathDelim + '*.txt'`, но при этом необходимо подключить модуль `System.IOUtils` (добавить в `uses System.IOUtils`).

Вывод информации производится на главной форме приложения. Создаётся список (компонент `TListView`) и в него добавляются элементы записей, сохранённые в файле. Каждый элемент списка содержит в себе текст темы записи и таймер обратного отсчёта до даты и времени, указанных в этой записи. При двойном клике по элементу всплывает сообщение, в котором указаны тема, текст записи, а также дата и время её создания или изменения.

4. Описание процесса переноса приложения на OS X и Android

Процесс компиляции приложения на OS X подразумевает наличие двух компьютеров: один – с операционной системой Windows, на котором приложение разрабатывается, и другой – с OS X, на котором приложение компилируется и запускается. Хотя на самом деле необязательно наличие обоих компьютеров, вполне достаточно одного, но в таком случае на нём необходимо запустить виртуальную машину с целевой операционной системой.

Для запуска OS X на одном компьютере с Windows можно использовать виртуальные машины VMWare Workstation, Oracle VM VirtualBox и другие. Так же вполне допустима такая связь: компьютер на OS X, с запущенной в виртуальной машине операционной системой Windows, в которой открыт Delphi XE7.

Но в любом случае далее необходимо установить утилиту PAServer в OS X, установочный файл которой есть в папке Embarcadero. Запустив PAServer в OS X, проверяем IP адрес машины, после чего в Delphi XE7 создаём профиль, в котором указываем имя профиля, IP адрес компьютера с OS X и порт. Осталось выбрать OS X в списке платформ разрабатываемого приложения и запустить проект. На Рис.5 представлен интерфейс приложения на платформе Mac OS X.

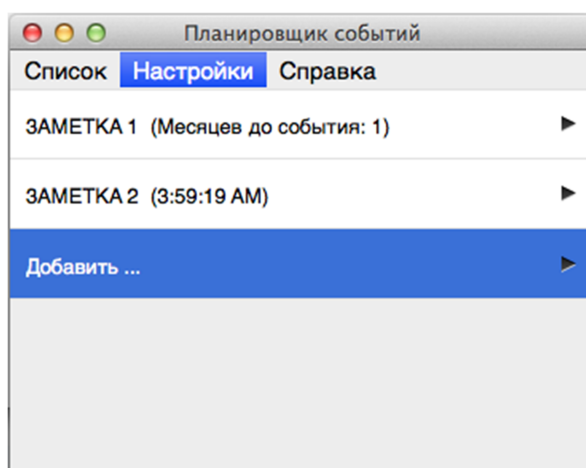


Рис. 5. Интерфейс приложения на платформе Mac OS X.

Процесс трансляций на Android устройства выглядит значительно проще. Достаточно в списке выбора платформ приложения указать Android и подсоединить устройство через USB-кабель к компьютеру, заранее включив на нём режим отладки, после чего запустить компиляцию. И через некоторое время, приложение, при отсутствии ошибок, запустится на подключенном устройстве (Рис. 6).

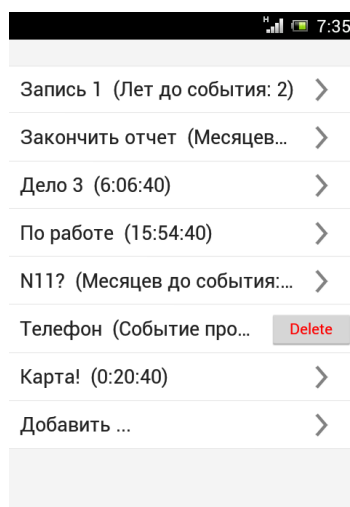


Рис. 6. Интерфейс приложения на платформе Android.

Наличие Android-устройства не обязательно. Даже если его не подключать к компьютеру, процесс компиляции пройдёт успешно, только появится извещение об ошибке, сообщающей о том, что для отладки необходимо подключить Android-устройство или эмулятор. После этого арк файл можно будет найти в директории проекта по пути: ...\\Android\\Debug*\\bin

5. Достоинства и недостатки кроссплатформенной разработки приложений

Главной особенностью кроссплатформенной разработки приложений является то, что приложения имеют единый исходный код, вследствие чего упрощается процесс создания приложений для различных платформ, а также намного повышается скорость разработки, исправления ошибок и последующего редактирования. Несмотря на то, что приложение исходит из единого кода, мы можем без проблем изменять различные параметры и свойства компонентов для разных платформ в процессе редактирования интерфейса приложения для целевой платформы.

Безусловное достоинство кроссплатформенного приложения - это возможность совместного использования его на разных платформах с возможностью обмена общими данными. Например, для приложения Единый планировщик событий для Windows, OS X и Android рассмотрена возможность обмена файлами между устройствами, на которых установлено данное приложение для возможности планирования событий сразу на нескольких устройствах.

К недостаткам можно отнести следующие моменты, например, при подобном способе разработки не получится пользоваться особыми свойствами каждой системы, придётся выбирать тот вариант, который будет одинаково работать везде. Так же к недостаткам следует добавить большой объём исполнительных файлов, который является достаточно важным для устройств под управлением Android OS. Например, размер исполняемого арк файла пустого откомпилированного приложения составляет порядка 7МБ.

Заключение

Рассмотрев способ разработки кроссплатформенного приложения в среде Embarcadero Rad Studio XE7, можно прийти к следующим выводам. Во-первых, данная среда даёт отличную возможность вести разработку нативных приложений для OS X и Android без подробного изучения программирования на конкретной целевой платформе. Во-вторых, значительно упрощается процесс переноса приложений на вышеперечисленные платформы, до такой степени, что можно обойтись без изменения кода. Однако, приложения, созданные в среде Embarcadero Rad Studio XE7, по некоторым характеристикам могут уступать приложениям, созданным на целевой платформе.

Список литературы

1. Путеводитель по органайзерам. Режим доступа:

<http://www.ixbt.com/soft/organizers.shtml> (дата обращения 5.03.2015)

2. Delphi XE7. Режим доступа: <http://www.embarcadero.com/ru/products/delphi/> (дата обращения 12.03.2015).
3. Вишневская Т.И., Романова Т.Н. Технология программирования: методические указания к лабораторному практикуму. В 3 ч. Ч.2. М.: Изд-во МГТУ им. Н.Э. Баумана, 2011. 71с.
4. Осипов Д.Л. Делфи. Программирование для Windows, iOS и Android. СПб.: БХВ-Петербург, 2014. 464с.
5. Вишневская Т.И. Тестирование ПО – как учебная дисциплина // Вестник Московского университета С.Ю. Витте. Сер. Образовательные ресурсы и технологии. 2014. № 1(4). С. 83-89.