

УДК 681.5

Программный модуль подготовки данных баллистической информации

*Телегин Е.А., студент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»*

*Научный руководитель: Черненко М.В., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
chernen@bmstu.ru*

Введение

В настоящее время околоземные космические аппараты (ИСЗ) широко используются для научных исследований, прикладных задач (в метеорологии, для навигации, связи, в военных целях), а также в образовании.

Важным аспектом управления полетом современных космических аппаратов (КА) является то, что контроль полета КА предполагает оценку значительного объема весьма разнородных параметров состояния. Для того чтобы справиться с этим потоком информации, возникает необходимость в разработке автоматизированных комплексов управления [1].

Одним из ключевых типов данных, определяющих состояние КА, являются баллистические параметры, которые характеризуют орбиту аппарата. Для того чтобы непрерывно контролировать положение спутника в пространстве, требуются средства, позволяющие максимально автоматизировать процесс сбора, хранения и обработки этих параметров. Разработке такого средства и посвящена данная работа.

Архитектура разрабатываемой системы

В качестве основных компонентов разрабатываемой системы возьмем базу данных, обеспечивающую накопление информации о характеристиках КА, его орбите (начальных условиях – НУ), и приложение, позволяющее получать доступ к этим данным и осуществлять расчеты. Разработанная структурная схема системы представлена на рис.1.

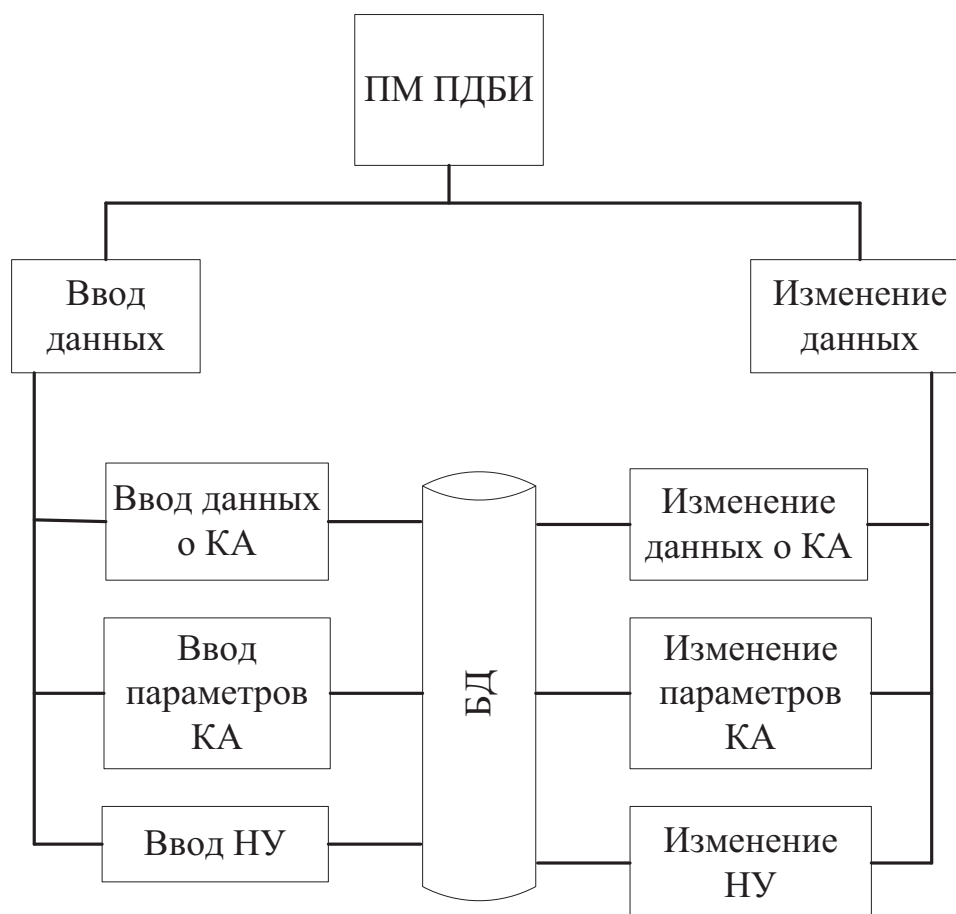


Рис. 1. Структурная схема системы

Выбор СУБД

Ввиду того, что для нашей системы существенной характеристикой является быстрота обращения к данным, наиболее подходящей для решения задачи является реляционная СУБД. В качестве альтернатив рассмотрим широко распространенные объектно-реляционные СУБД Oracle XE 11g, PostgreSQL 9.3.2 и реляционную Oracle MySQL 5.6.15. Результаты сравнения приведены в таблице 1.

Таблица 1

Сравнение СУБД

	Вес. коэф.	Oracle Database XE 11g	PostgreSQL 9.3.2	Oracle MySQL 5.6.15
Скорость обработки	0,4	Отлично	Хорошо	Удовлетворительно

больших объемов данных				
Трудоемкость разработки	0,2	Хорошо	Хорошо	Отлично
Возможность распараллелива ния запросов по ядрам	0,3	Да	Нет	Нет
Удобство администриро- вания	0,1	Хорошо	Отлично	Удовлетворительно
Итог		0,88	0,75	0,68

Таблица 2

Шкала перевода качественных характеристик в количественные

Качественная характеристика	Количественная характеристика
Удовлетворительно	0,6
Хорошо	0,8
Отлично	0,9
Да	1
Нет	0,6

Выбор технологии разработки пользовательского интерфейса

Создаваемая система ориентирована на работу в ОС семейства Windows, поэтому целесообразно будет воспользоваться возможностями платформы .NET Framework.

Рассмотрим две основные технологии для создания оконного интерфейса, входящие в состав платформы: Windows Forms и Windows Presentation Foundation (WPF) [2].

WPF – система для построения клиентских приложений [Windows](#) с визуально привлекательными возможностями взаимодействия с пользователем, использующая язык [XAML](#). В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. Графической технологией, лежащей в основе WPF, является [DirectX](#), в отличие от [Windows Forms](#), где используется [GDI+](#). Производительность WPF выше, чем у [GDI+](#), за счёт использования аппаратного ускорения графики через [DirectX](#). Преимущество WPF состоит еще и в том, что пользовательский интерфейс полностью отделяется от кода, и дизайнеры графики могут использовать профессиональные инструменты для редактирования файлов XAML, улучшая внешний вид всего приложения. Результаты сравнения технологий приведены в таблице 3.

Таблица 3

Сравнение технологий разработки

Параметр	Вес. коэф.	WPF	WinForms
Системные требования	0,2	Высокие	Небольшие
Возможности графического дизайнера	0,5	Отлично	Удовлетворительно
Аппаратное ускорение графики	0,3	Да	Нет
Итого		0,86	0,64

Таблица 4

Шкала перевода качественных характеристик в количественные

Качественная оценка	Количественная оценка	Качественная оценка	Количественная оценка	Качественная оценка	Количественная оценка
Возможности графического дизайнера		Системные требования		Аппаратное ускорение графики	

Удовлетворительно	0,6	Высокие	0,6	Да	0,8
Отлично	1	Небольшие	0,8	Нет	0,6

В качестве среды разработки воспользуемся Microsoft Visual Studio, включающей в себя редактор исходного кода с поддержкой технологии [IntelliSense](#), встроенный отладчик и редактор форм для упрощения создания графического интерфейса приложения. Разработку логики работы программы будем вести на языке C#.

Взаимодействие приложения с БД

Для связи с данными в приложении используются источники данных, которые могут быть получены из баз данных (включая локальные файлы баз данных). Visual Studio предоставляет средства разработки для создания и редактирования источников данных, используемых в приложении [2].

Прежде чем добавить в приложение источник данных, необходимо произвести подключение к БД. При этом требуется выбрать поставщик данных. Поставщик данных .NET Framework используется для установления соединения с базой данных, выполнения команд и получения результатов. В нашем случае это будет Oracle Data Provider for .NET (ODP.NET), входящий в состав пакета Oracle Data Access Components (ODAC) [3].

Основным компонентом архитектуры .NET, обеспечивающим взаимодействие с базой данных, является набор данных. Набор данных представляет собой объект, содержащий коллекцию таблиц данных, в которых можно временно хранить данные для их использования в приложении. Данные для работы приложения в этом случае будут храниться в локальной памяти. Можно работать с данными в наборе данных, даже если приложение отключается от базы данных. Таблицы данных заполняются путем выполнения запросов адаптера таблиц. Набор данных хранит информацию об изменениях своих данных. Так что обновления можно отслеживать и отправлять обратно в базу данных, когда приложение вновь подключится к ней.

Структура набора данных схожа со структурой реляционной базы данных: она представляет собой иерархическую объектную модель таблиц, строк, столбцов, ограничений и связей.

Наборы данных могут быть типизированными или нетипизированными. Типизированные наборы данных получают свою схему (таблицу и структуру столбцов) из

файлов XSD и проще программируются. Нетипизированный набор данных не имеет соответствующей встроенной схемы. Как и типизированный набор данных, он содержит такие структуры, как таблицы, столбцы и др., однако они доступны только как коллекции.

В этой модуле будем использовать типизированный набор данных. т.к. для поддержки наборов этого типа в Visual Studio предусмотрено больше инструментов, которые упрощают процесс программирования и уменьшают вероятность возникновения ошибки.

Адаптер таблиц подключается к базе данных, выполняет запросы или хранимые процедуры и либо возвращает новую заполненную таблицу данных, либо заполняет существующую таблицу возвращаемыми данными [4]. Адаптеры таблиц также используются для отправки обновленных данных из приложения обратно в базу данных. Адаптеры таблиц могут содержать несколько запросов для заполнения связанных таблиц. При этом каждый запрос возвращает данные, которые соответствуют той же схеме, что и связанная таблица данных. Это позволяет загружать данные, которые удовлетворяют различным критериям.

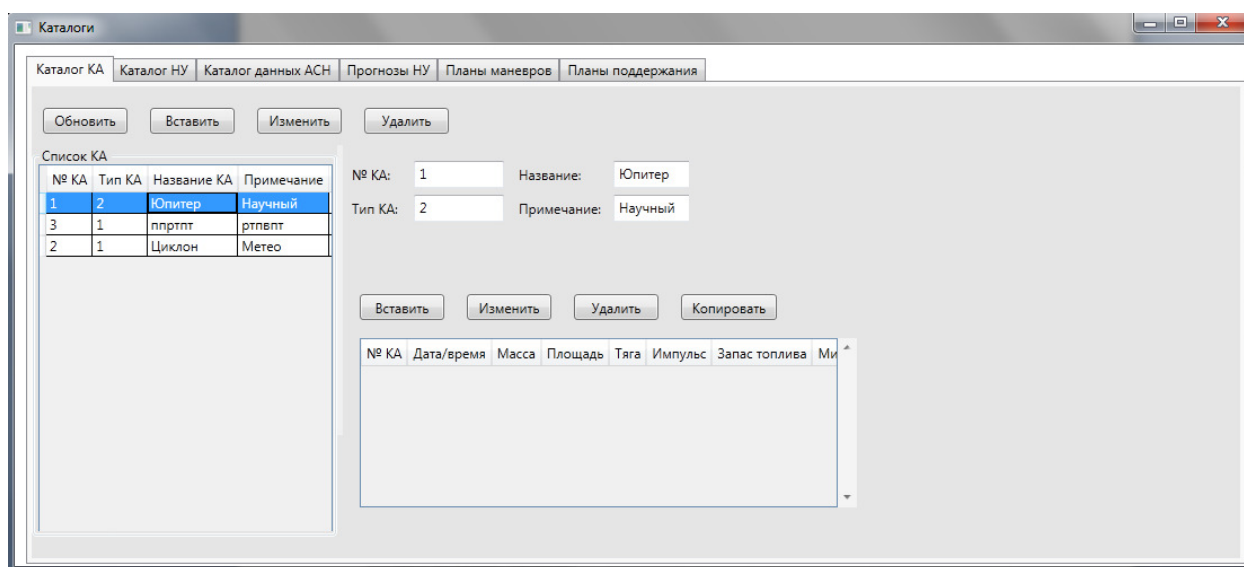


Рис.2. Работа с КА и их параметрами

Для организации взаимодействия интерфейса с набором данных будем использовать встроенный в WPF механизм привязки данных (Binding). Привязка данных WPF обеспечивает простой и согласованный способ представления данных и взаимодействия с ними в приложениях. Если привязка имеет правильные параметры и данные предоставляют правильные уведомления, то при изменении значений данных в элементах,

которые привязаны к данным, автоматически отражаются изменения. Привязка к данным также означает, что, если внешнее представление данных в элементе изменяется, то базовые данные автоматически обновляются. Например, связав расположенное на форме поле ввода с полем таблицы набора данных, получим возможность изменять значение ячейки таблицы прямо из формы.

Реализация программы

Parameter	Value
КА	1
Дата/время	27.02.2014 13:08:37
Масса	492.4
Площадь поверхности:	17.3
Тяга:	28.4
Импульс:	96.5
Запас топлива:	184.6
Максимальное время маневра:	3.9
Минимальное время маневра:	2.6
Поперечное сечение:	4.2
Коэффициент отражения:	0.17
Коэффициент сопротивления:	0.03
Максимальный угол:	12.4
Примечание:	начальные

Рис.3. Форма ввода параметров КА

При запуске приложения открывается главное окно, содержащее вкладки. Первая вкладка позволяет пользователю получить доступ к редактированию параметров КА. Она обеспечивает возможность нажатием соответствующих кнопок вставлять, изменять и удалять КА и их характеристики. При выборе КА из списка отображается таблица со

списком его характеристик. Внешний вид вкладки представлен на рисунке 2, формы для ввода параметров КА – на рисунке 3.

Вторая вкладка предоставляет доступ к работе с начальными условиями (НУ). Сверху располагается выпадающий список для выбора КА, ниже отображается список НУ для выбранного аппарата. Вставка, изменение и удаление их осуществляются по нажатию соответствующих кнопок. Форма для ввода НУ изображена на рисунке 4.

ID НУ	1	Примечание	
№ КА	1	ОЗ	
№ НУ	10		
Время НУ	13.04.2013 01:01:01		
Доли секунд	0.85038		
Виток	30		
БК, м2/кг	0.004506		
КСД	0.065333333		
Минимальная высота:	0		
Максимальная высота:	0		
Оскулирующие элементы			
u	0	аргумент широты, градусы	
a	6657.4130846730	большая полуось, км	
w	104.6067	аргумент перигея, градусы	
e	0.0031299	эксцентриситет	
i	51.67326	наклонение, градусы	
omega	70.5953	долгота восходящего узла, градусы	
<input type="button" value="Пересчитать в ГПСК и АПСК"/>			
ГПСК			
X	-5526628.97405088	м	
Y	-3721115.45882199	м	
Z	0.0000000000	м	
Vx	460672.785133291	м/с	
Vy	-164977.586669743	м/с	
Vz	6065.42316494	м/с	
<input type="button" value="Пересчитать в ОЗ и АПСК"/>			
АПСК			
X	2213574.482099629	м	
Y	6284140.043546131	м	
Z	0.0000000000	м	
Vx	-4071.95457806	м/с	
Vy	1409.48975724	м/с	
Vz	6065.42316494	м/с	
<input type="button" value="Пересчитать в ОЗ и ГПСК"/>			
<input type="button" value="OK"/> <input type="button" value="Отмена"/>			

Рис. 4. Форма ввода НУ

Отличительной особенностью формы ввода НУ является то, что она позволяет преобразовывать введенные данные между тремя системами координат по щелчку на соответствующих кнопках.

Технология выполнения преобразований

Разрабатываемое приложение позволяет вводить данные о космических аппаратах (название, тип, назначение), их характеристиках (масса, запас топлива, импульс и т.п.) и характеристиках их орбит (эксцентриситет, большая полуось, наклонение и т.п.), которые являются исходными данными для расчета корректирующих маневров. Эти исходные

данные, называемые также начальными условиями, с помощью вызываемых функций преобразований могут быть представлены в трех системах координат: оскулирующей орбиты (ОО), гринвичской прямоугольной (ГПСК) и астрономической прямоугольной (АПСК).

Указанные преобразования реализованы с помощью обмена данными со вспомогательным вычислительным модулем через файлы. Для выполнения пересчета из одной системы координат в другую исходные параметры записываются во входной файл и запускается модуль пересчета. Он, в свою очередь, записывает результаты вычислений в выходной файл, из которого их забирает приложение.

Заключение

В результате проделанной работы достигнуты следующие цели:

- Разработана и реализована реляционная БД
- Разработан и реализован пользовательский интерфейс
- Реализован механизм взаимодействия пользовательского приложения с БД
- Реализован механизм взаимодействия приложения с внешним

вычислительным модулем.

Список литературы

1. Соловьев В.А., Лысенко Л.Н., Любинский В.Е. Управление космическими полетами: учебное пособие. В 2 ч. Ч. 2. М.: Изд-во МГТУ им. Н.Э. Баумана, 2010. 426 с.
2. Нейгел К., Ивсен Б., Глинн Д., Уотсон К., Скиннер М. С# 4.0 и платформа .NET 4 для профессионалов: пер. с англ. М.: Вильямс, 2011. 1440 с. [Nagel C., Evjen B., Glynn J., Watson K., Skinner M. Professional C# 4 and .NET 4. Wrox, 2010. 1536 p.]
3. Гринвальд Р., Стаковьяк Р., Стерн Д. Oracle 11g. Основы: пер. с англ.. М.: Символ-Плюс, 2013. 464 с. [Greenwald R., Stackowiak R., Stern J. Oracle Essentials: Oracle Database 11g. 4th ed. O'Reilly, 2007. 408 p.]
4. Рихтер Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#. 3-е изд.: пер. с англ. М.: Мир книг, 2013. 928 с. [Richter J. CLR via C#. 4th ed. Microsoft Press, 2012. 864 p.]
5. Григорьев Ю.А., Ревунков Г.И. Банки данных: учебник для вузов. М.: МГТУ им. Н.Э.Баумана, 2002. 320 с.

6. Виноградова М.В., Игушев Э.Г. Принципы организации структуры данных с произвольным доступом и быстрой операцией вставки // Вестник МГТУ им. Н.Э.Баумана. Сер. Приборостроение. 2012. № 5. С. 20-30.