

СООБЩЕНИЯ  
ОБЪЕДИНЕННОГО  
ИНСТИТУТА  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ

Дубна

P10-2000-315

М.А.Назаренко, Э.Г.Никонов, А.В.Старцев

**ИСПОЛЬЗОВАНИЕ OBJECTIVITY/DV  
ДЛЯ СОЗДАНИЯ  
ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ БАЗЫ ДАННЫХ  
ПО СУБМОДУЛЯМ TILE-КАЛОРИМЕТРА  
ПРОЕКТА ATLAS**

2000

# Введение

Одним из перспективных направлений в современном мире информационных технологий является создание приложений объектно-ориентированных баз данных (ООБД). Использование таких систем может быть особенно эффективным в тех областях, где есть необходимость работать с данными сложной структуры и большого объема. К таким областям относится и обработка данных физических экспериментов.

Именно ООСУБД, а точнее – Objectivity/DB, предполагается использовать для хранения и обработки данных эксперимента ATLAS. Эта работа демонстрирует процесс создания объектно-ориентированной базы данных на основе Objectivity/DB и показывает некоторые возможности использования данной технологии в области разработки программного обеспечения для физических экспериментов.

Пакет Objectivity/DB является коммерческим продуктом и его использование возможно лишь при наличии лицензии, работа выполнена благодаря лицензионному соглашению ЦЕРН-ОИЯИ для эксперимента ATLAS.

В задачи этой работы не входит рассмотрение основ ООБД и сравнительный анализ реляционных и объектных баз данных. Подобные вопросы довольно подробно изложены в многочисленных источниках [1-6]. Предполагается также, что читатель знаком с базовыми понятиями объектно-ориентированного программирования (ООП). Любому, кто собирается серьезно заниматься разработкой в Objectivity/DB, следует иметь под рукой документацию по этому пакету [7-10].

## Постановка задачи

Цель этой работы — продемонстрировать процесс создания объектно-ориентированной базы данных в Objectivity/DB и возможности объектно-ориентированных технологий на примере информационной системы хранения и обработки данных по субмодулям установки ATLAS.

Исходными данными являются паспорта субмодулей в формате файлов MS Word 97. Каждый файл создается по определенному шаблону (см. приложение А) и соответствует одному субмодулю. В файле содержатся данные о названии и типе субмодуля, дате и месте изготовления и отклонении от нормального положения, а также другие данные, не являющиеся значимыми в пределах данной задачи.

Поступающие данные необходимо объединить в единой информационной системе, которая позволит хранить данные по субмодулям, импортировать их из doc-файлов, выводить данные через WWW-интерфейс. Схема функционирования такой системы и ее взаимодействия с пользователями представлена на рис. 1.

Запрос администратора на добавление субмодуля содержит файл в формате MS Word 97, содержащий характеристики субмодуля. Ответом на этот запрос может быть сообщение об удачном добавлении объекта, о существовании объекта с таким заголовком в ООБД или о других ошибках (например, неверный формат файла). WWW-интерфейс обеспечивает возможность просмотра списка объектов, находящихся в базе данных, и динамически формируемого отчета по любому из них.

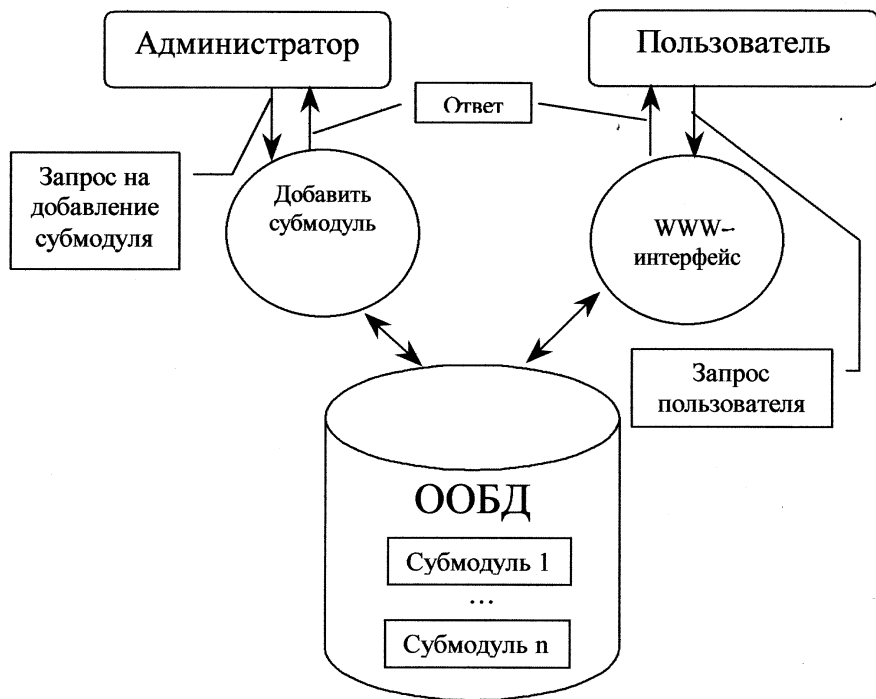


Рис. 1.Схема функционирования создаваемой информационной системы

## Реализация

Этот раздел поэтапно описывает процесс создания ООБД. На сайте «<http://atlasinfo.jinr.ru>» содержится пример, в котором реализованы все шаги, описываемые ниже. Для его корректной работы требуется установить операционную систему Windows NT 4.0, среду программирования Visual C++ 5.0, пакет Objectivity/DB 5.0, Objectivity/C++ 5.0. Для просмотра содержимого базы данных через WWW-интерфейс достаточно иметь интернет-браузер Internet Explorer 4.0 или выше.

Порядок использования демонстрационного примера.

Скопировать файл `smsample.exe` в корневую директорию `Objectivity/DB` и запустить его. После этого появятся файл `runme.bat` и новая поддиректория `SMSAMPLE`.

Запустить `runme.bat`, и в поддиректории `SMSAMPLE` будут созданы федеративная ООБД и файлы доступа к ней. В качестве примера во вновь созданную ООБД будут помещены два объекта, импортированные из файлов `qsJIN002.doc` и `qsJIN040.doc`, также входящих в данный пример.

Ниже описаны этапы создания ООБД и пользовательского интерфейса к ней.

## 1. Создание федеративной базы данных

Для создания федеративной базы данных (ФБД) в `Objectivity/DB` достаточно выполнить одну команду:

```
oonewfd -fdfilehost wnct126  
        -fdfilepath D:\ObjDB\Subm\sm01.fdb  
        -lockserverhost wnct126 smfdb01,
```

где `fdfilehost` — имя машины, на которой расположен файл базы данных (`wnct126.jinr.ru`),

`fdfilepath` — полный путь к файлу федеративной базы данных (`D:\ObjDB\Subm\sm01.fdb`),

`lockserverhost` — имя машины, на которой запущен `Objectivity Lock Server`, к которому будет обращаться данная ФБД (`wnct126.jinr.ru`),

`smfdb01` — системное имя ФБД, которое совпадает с именем ее загрузочного файла.

Перед созданием ФБД и выполнением других команд Objectivity/DB следует убедиться в том, что на машине, имя которой указано в качестве *lockserverhost*, запущен Objectivity Lock Server (LS). Для этого можно воспользоваться командой *oocheckls*, а для запуска LS – командой *oolockserver*. Подробно о командах Objectivity/DB можно узнать в [8] или запустив соответствующую команду с параметром *help*.

С помощью команды *oonewfd* создана ФБД – наиболее высокий уровень в структуре Objectivity/DB. Структура создаваемой ФБД представлена на рис. 2. Федеративная база данных объединяет БД, расположенные в общем случае на различных машинах. В данном случае ФБД будет содержать одну БД. Каждая БД в составе ФБД состоит из контейнеров. Объекты при записи в БД попадают в



определенный контейнер, если он указан, или в контейнер по умолчанию в противном случае. В демонстрационном примере в БД создается отдельный контейнер для хранения данных по субмодулям.

Рис. 2. Структура ООБД

## 2. Описание класса

В ООБД могут храниться объекты, принадлежащие только заранее описанным классам. Прежде чем создавать новый объект, необходимо описать класс, представителем которого он будет. Для этой цели используется Data Definition Language (DDL) и соответствующая утилита Objectivity/DB – DDL-процессор. Класс описывается в файле с расширением `ddl` (в примере - `subm.ddl`). Все классы в ФБД должны быть наследниками класса `ooObj` из библиотеки Objectivity/C++:

```
class SubModule:public ooObj
```

Класс `SubModule` содержит атрибуты, соответствующие метке (`label`) и типу субмодуля, массив отклонений, а также конструкторы и методы, обеспечивающие доступ к значениям атрибутов и импорт данных. Полный текст этого и других файлов можно найти в демонстрационном примере. Язык DDL описан в [9].

Для добавления описанного класса в структуру ФБД используется DDL-процессор:

```
ooddlx subm.ddl new01,
```

где `subm.ddl` — файл с описанием класса на DDL,

`new01` — системное имя ФБД.

DDL-процессор добавляет описание класса в схему ФБД и создает три файла: `subm.h`, `subm_ref.h`, `subm_ddl.cpp`. `Subm.h` содержит все описания, необходимые для работы с объектами-экземплярами описанного класса. Он также содержит директиву `#include <oo.h>`, так что при написании приложений для работы с ООБД на Objectivity/C++ достаточно будет включить `subm.h` для получения доступа к Objectivity/C++ API. `Subm_ref.h` содержит описания, необходимые для

использования итераторов, ссылок на созданный класс (см. [10]) и т.д. `Subm_ddl.cpp` реализует методы, необходимые для работы с объектами-экземплярами класса.

При создании приложений, использующих объекты-экземпляры описанного класса, при помощи Objectivity/C++ (см. ниже) необходимо включить в исходный текст программы строку следующего вида:

```
#include "subm.h"
```

### 3. Разработка интерфейса пользователя

Интерфейс пользователя ООБД реализуется набором исполняемых файлов, которые обеспечивают импорт данных (`write.exe`), просмотр содержимого ООБД в виде текста (`read.exe`) и CGI-скрипт для работы с ООБД через WWW.

При создании пользовательского интерфейса используется язык программирования Objectivity/C++, т.е. для разработки собственно интерфейса применяется язык C++ и его расширение (Objectivity/C++) – для открытия и закрытия ФБД, БД, получения объекта из ООБД, выполнения транзакций и т.п.

Исходные тексты для `write.exe` и `read.exe` (`write.cpp` и `read.cpp` соответственно) можно найти в примере на сайте «<http://atlasinfo.jinr.ru>». Язык Objectivity/C++ описан в [10]. Реализация WWW-интерфейса в пример не вошла, но ознакомиться с таким интерфейсом можно на <http://wnct126.jinr.ru/objdb> (см. приложение Б).

Запуск `write.exe` с именем файла в качестве параметра вызывает добавление нового объекта в ООБД, если файл с таким именем существует и представляет собой паспорт submodule в формате MS Word 97, соответствующий заданному шаблону.



При запуске *read.exe* все данные по всем субмодулям, хранящимся в ООБД, будут выведены на экран и записаны в текстовый файл *subm.log*. Просмотреть содержимое ООБД можно также при помощи утилиты *oobrowse.exe*, входящей в пакет Objectivity/DB (см. приложение В).

## **Заключение**

В результате работы создан действующий образец информационной системы на базе Objectivity/DB, обеспечивающий хранение данных по субмодулям проекта ATLAS. Прделанная работа может послужить примером для начинающих разработчиков в Objectivity/DB, а созданная система – основой для дальнейших разработок в этой области, т.к. средства Objectivity/DB дают возможность достаточно гибко расширять и модифицировать уже существующие ООБД.

Одним из возможных путей дальнейшего развития созданной системы может стать реализация возможности изменения имеющихся характеристик хранимых объектов (субмодулей) и добавления новых характеристик.

# Приложения

## Приложение А. Пример паспорта субмодуля (фрагмент)

**ATLAS TILECAL SUBMODULES QUALITY CONTROL PROTOCOL**

<b>Submodule Label: JIN040</b>	<b>Institute: Joint Institute for Nuclear Research</b>
--------------------------------	--

### **THICKNESS MEASUREMENTS**

*According to drawing ATLLBMS\_0002*

**Submodule Type:**                    **STANDARD:**                    **Design height = 291.7mm**  
*(Check 1)*

**SPECIAL:**                                **Design height = 341.2mm**

<b>Point</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>Height deviation (mm)</b>	-0.40	+0.04	-0.20	-0.17	-0.34	-0.30	-0.60

<b>Point</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>
<b>Height deviation (mm)</b>	-0.30	-0.03	-0.20	-0.19	-0.21	-0.32

<b>Point</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Height deviation (mm)</b>	-0.42	-0.11	-0.25	-0.17	-0.29	-0.36	-0.70

Приложение Б. Иллюстрация работы с системой через  
WWW-интерфейс (отчет, генерируемый при выборе одного  
из объектов)

This file was generated by Objectivity/DB application programm 11/15/2000 15:59:50

**ATLAS TLECAL SUBMODULES QUALITY CONTROL PROTOCOL**

Submodule Label: qcJIN040.doc Institute: JINR, Dubna, Russia

**THICKNESS MEASUREMENTS**

Submodule type:

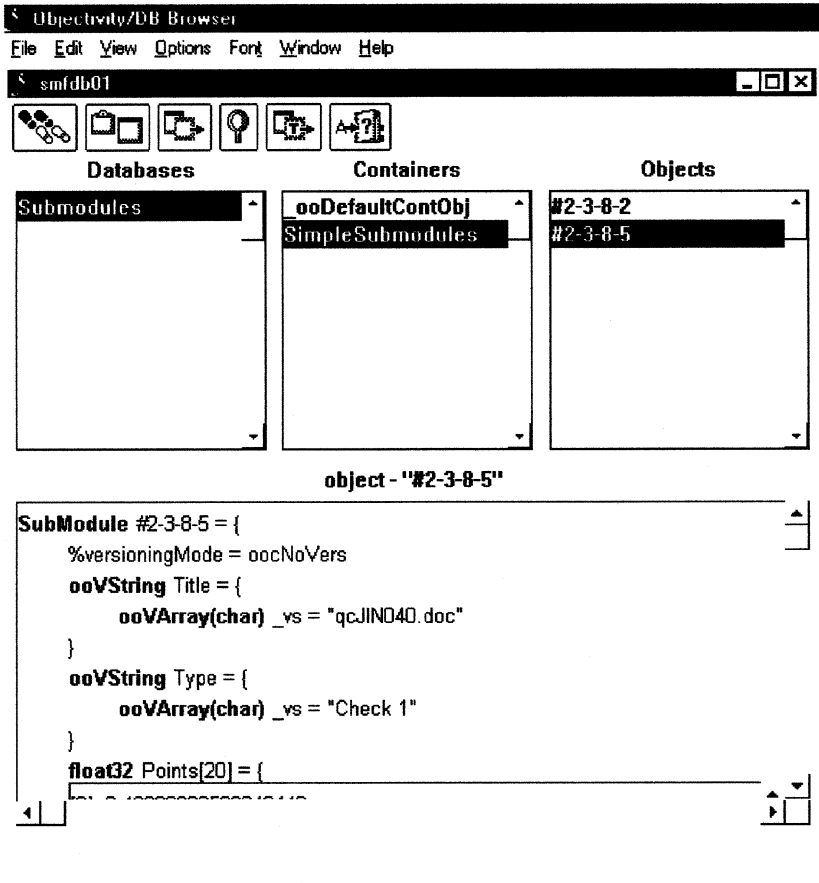
*(Check 1)*

Point	1	2	3	4	5	6	7
Height	-0.40	0.04	-0.20	-0.17	-0.34	-0.30	-0.60
Point	8	9	10	11	12	13	14
Height	-0.70	-0.36	-0.29	-0.17	-0.25	-0.11	-0.42
Point	15	16	17	18	19	20	21
Height	-0.30	-0.03	-0.20	-0.19	-0.21	-0.32	0.00

Maximum deviation from nominal (+):0.04

Minimum deviation from nominal (-):-0.70

# Приложение В. Пример работы утилиты ooBrowse



## Список литературы

1. Кузнецов С. Тенденции в мире систем управления базами данных ([http://www.citforum.ru/database/articles/art\\_25.shtml](http://www.citforum.ru/database/articles/art_25.shtml)).
2. Кузнецов С. Что было, то и теперь есть, и что будет, то уже было... (<http://www.citforum.ru/database/articles/oozinder.shtml>).
3. Кузнецов С. Объектно-ориентированные базы данных — основные концепции, организация и управление: краткий обзор ([http://www.citforum.ru/database/articles/art\\_24.shtml](http://www.citforum.ru/database/articles/art_24.shtml)).
4. Андреев А.А., Березкин Д.В., Кантонистов Ю.А. Объектные СУБД: состояние и перспективы (доклад на семинаре московской секции ACM SIGMOD, по материалам сервера издательства «Интелтек», <http://www.inteltec.ru/publish/articles/objtech/objcond.shtml>).
5. Андреев А.А., Березкин Д.В., Кантонистов Ю.А. Объектные СУБД на российских просторах. Компьютерная хроника, N11, 1997.
6. Медников А.Ю., Соловьев А.Ю. Объектно-ориентированные базы данных сегодня или завтра? Открытые системы, N4, 1994.
7. Objectivity/C++ Active Schema Release 5.2.
8. Objectivity/DB Administration Release 5.2.
9. Objectivity/C++ Data Definition Language Release 5.2.
10. Using Objectivity/C++ Version 4.
11. Шилдт Г., Дюссельдорф. Теория и практика С++. СПб.: ВНВ-С, 1999.
12. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. М.: Бином, 1998.
13. Страустрап Б. Что такое объектно-ориентированное программирование (<http://infocity.kiev.ua/mttitle.phtml?id=110>).
14. Кузнецов С. Обзор статьи Modeling Object/Relational Databases ([http://www.citforum.ru/database/digest/dig\\_1803.shtml](http://www.citforum.ru/database/digest/dig_1803.shtml)).
15. Березкин Д.В. Объектное проектирование и объектные базы данных. Компьютерная хроника, N4, 1999.
16. Андреев А.М., Березкин Д.В., Кантонистов Ю.А., Мальцев С.А. Web-интерфейс объектной СУБД. Компьютерная хроника, N4, 1999.
17. Андреев А.М., Березкин Д.В., Кантонистов Ю.А. Объектно-ориентированные базы данных: среда разработки программ плюс хранилище объектов. Мир ПК, N4, 1998.

Рукопись поступила в издательский отдел  
27 декабря 2000 года.

Назаренко М.А., Никонов Э.Г., Старцев А.В.

P10-2000-315

## Использование Objectivity/DB

для создания объектно-ориентированной базы данных  
по субмодулям tile-калориметра проекта ATLAS

Создан действующий образец информационной системы на базе Objectivity/DB, обеспечивающий хранение данных по субмодулям проекта ATLAS. Прделанная работа может служить примером для начинающих разработчиков в Objectivity/DB, а созданная система — основой для дальнейших разработок в этой области, т.к. средства Objectivity/DB дают возможность достаточно гибко расширять и модифицировать уже существующие объектно-ориентированные базы данных.

Objectivity/DB является коммерческим продуктом, все работы были выполнены благодаря лицензионному соглашению ЦЕРН–ОИЯИ для эксперимента ATLAS.

Работа выполнена в Лаборатории информационных технологий и в Лаборатории ядерных проблем им. В.П.Джелепова ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна, 2000

### Перевод авторов

Nazarenko M.A., Nikonov E.G., Startsev A.V.

P10-2000-315

## Using Objectivity/DB to Create Object-Oriented

ATLAS Tilecal Submodules Database

A working instance of information system based on Objectivity/DB was developed to perform ATLAS submodules data storage. The work accomplished may serve as an example for starting Objectivity/DB developers and the system created may become a base for further development in this field, since Objectivity/DB environment allows flexible extension and modification of existing object-oriented databases.

Objectivity/DB is a commercial product, all the work was done thanks to CERN–JINR licence agreement for ATLAS project.

The investigation has been performed at the Laboratory of Information Technologies and at the Dzhelepov Laboratory of Nuclear Problems, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna, 2000

Редактор Е.Ю.Шаталова. Макет Н.А.Киселевой

Подписано в печать 14.02.2001

Формат 60 × 90/16. Офсетная печать. Уч.-изд. листов 1,32

Тираж 300. Заказ 52499. Цена 1 р. 58 к.

Издательский отдел Объединенного института ядерных исследований  
Дубна Московской области