

**БИБЛИОТЕКА КЛАССОВ И АЛГОРИТМОВ ДЛЯ ОБРАБОТКИ РЯДОВ
ГЕОФИЗИЧЕСКИХ НАБЛЮДЕНИЙ**

**LIBRARY OF CLASSES AND ALGORITHMS FOR GEOPHYSICAL OBSERVATION
SERIES PROCESSING**

С.Э. Смирнов¹, А.В. Поляруш²

¹*Институт космических исследований и распространения радиоволн ДВО РАН,*
²*Камчатский государственный университет имени Витуса Беринга*

In IKIR a long series of geophysical observations are stored as a bank of data composed of a set of text files of a certain format. In order to use this data service classes were developed to solve the following tasks:

- I application of uniform presentation of geophysical data in computing environment;
- I determination of procedures to handle the groups of geophysical original data;
- I determination of procedures to separate the data from the mass for the further processing;
- I realization of algorithms for mathematical processing of geophysical data;
- I transformation of data from one format to another one.

The developed class system represents the original data on the required level of abstraction to perform mathematical modeling of geophysical processes.

We offer the developed procedure for a wide discussion and revision in order to form a bank of algorithms for geophysical data processing.

В данной работе была продолжена создание иерархической системы классов [1] низкоуровневой обработки данных.

Исходные требования к разрабатываемым классам были следующие: не зависимость от операционной системы (Unix, Windows), не зависимость от типа компиляторов (т.е. стандартный C++).

Имена файлов архива формируется следующим образом:

Расшифровка имени файла `ikggmmdd.abb` (в файле 1 сутки)

`ik` - символы "ik"

`gg` - год

`mm` - месяц

`dd` - день

`a` - номер станции

`bb` - идентификатор канала

Расшифровка имени файла `ikggmm.abb` (в файле 1 месяц)

`ik` - символы "ik"

`gg` - год

`mm` - месяц

`a` - номер станции

`bb` - идентификатор канала

Номера станций:

1 - Паратунка (лаборатория ГФП)

2 - Карымшина (лаборатория ГФП)

3 - Карымшина (лаборатория АИ)

4 - ст. Забайкальское (Хабаровск)

5 - Магадан

6 - Сахалин

7 - Микижа

Сами данные представлены в виде наборов строк:

2007.05.30 00:12:00 14.8 32.0 0.00 4.33 200.0 990.0

Первые 2 столбца представляют дату и время по UT, далее записаны сами значения. Соответственно базовый класс данных класс даты и массив данных одной строки:

```
class MGDData
{
private:

public:
MGDData();
SDate date; // класс даты
int Count; // всего каналов в файле
double x[MaxChan];
~MGDData();
};
```

Таким образом все геофизические данные имеют привязку к мировому времени. Поскольку в стандартный набор классов C++ не входит класс работы с датой, был разработан такой класс, арифметическими и булевыми операциями:

```
class SDate
{
MYDATE d_year, /* Year (calendar year) */
d_mon, /* Month (1--12) */
d_mday, /* Day of month (1--31) */
d_hour, /* Hour (0--23) */
d_min, /* Minutes */
d_sec, /* Seconds */
d_msec, /* миллисекунды */
double JD; // юлианская дата
...
public:
bool operator == (const SDate& D); // проверка на равенство
bool operator != (const SDate& D); // проверка на неравенство
bool operator > (const SDate& D) // проверка (больше)
bool operator >= (const SDate& D); // проверка (больше или равно)
bool operator < (const SDate& D) // проверка (меньше)
bool operator <= (const SDate& D); // проверка (меньше или равно)
friend SDate operator + ( SDate&, Sdate&); // сложение дат
friend SDate operator + (SDate&, const double&); // прибавить Юлианскую дату
friend SDate operator + (SDate&, const long&); // прибавить секунды
friend SDate operator - (SDate&, const double&); // SDate-JD
friend SDate operator - (SDate&, const long&); // Sdate-sec
};
```

Следующим шагом было создание класса чтения данных из файлов в массив MGDData.

```
class MGDReadData // фрагмент объявления класса
{
bool ret; // флаг ошибки
char sError[100]; // сообщение о последней ошибке
public:
long Count; // количество считанных записей
bool ReadData(MGDData& dan);
};
```

Этот класс абстрагируется от разбиения данных на отдельные файлы, их местоположения. В программе заполнение массива выполняется всего в несколько строк:

```
for(;;)
{
if(!RD.ReadData(dan) // заполнение массива MGDData dan[]
{
RD.SetLastError(sError); // диагностика ошибки
break;
}
}
```

Наиболее востребованная операция над данными – это выборка данных. Для этой процедуры был разработан класс ChoiseData. Главный метод этого класса – выборка от начальной даты до конечной с заданным интервалом:

```
if(!ChD.RunDist(d_start,d_end))
{
ChD.SetLastError(str);
cout << str << endl;
}
```

Для доступа к данным этот класс обращается к классу MGDReadData. Метод вывода результата работы в файл позволяет использовать различные форматы представления файлов.

Для работы с данными произвольного объема был разработан класс MGDDataFile. Его особенностью является то, что данные не загружаются целиком в оперативную память, а считываются последовательно из файла данных. Поэтому объем данных ограничиваются только объемом носителя.

Для этого класса были реализованы арифметические операции с самим классом, арифметические операции с константой, операции с потоком ввода/вывода. А так же разработаны методы обработки данных:

```
bool Dispersion(); - вычисление дисперсии;
bool Smooth(); - сглаживание рядов данных;
bool DFT() - дискретное преобразование Фурье;
bool IDFT() - обратное дискретное преобразование Фурье.
```

В результате была создана библиотека методов первичной обработки данных. Достигнут высокий уровень абстрагирования от рутинной работы с файловой системой и форматами данных, что позволяет быстро приступить непосредственно к математическому моделированию физических процессов представленных банками данных.

Список литературы

1. Смирнов С.Э. Основные классы С++ информационного комплекса геофизической обсерватории Паратунка // Судовое оборудование, судовождение, безопасность мореплавания и жизнедеятельности. КамчатГТУ. 2005. С. 81-86.