

ЧТО ТАКОЕ IDL?

М.А. Болсуновский (Совзонд)

В 1990 г. окончил Киевское высшее инженерное радиотехническое училище. После окончания училища служил в рядах ВС РФ. С 2000 г. работал в ООО «Гео Спектрум», а с 2002 г. — в ФГУП ВО «Техмашимпорт». В 2004 г. получил степень «Мастер делового администрирования в области стратегического планирования» (Master of Business Administration) во Всероссийской академии внешней торговли МЭРИТ РФ. С 2004 г. работает в компании «Совзонд», в настоящее время — заместитель генерального директора.

В.О. Скрипачев (ФГУП «Научный центр космических информационных систем и технологий наблюдения»)

В 2003 г. окончил факультет оптического приборостроения МИИГАиК по специальности «оптико-электронные приборы и системы». После окончания университета работает научным сотрудником в ФГУП «Научный центр космических информационных систем и технологий наблюдения».

Язык программирования IDL (Interactive Data Language) зародился в 1970-х гг. в Лаборатории атмосферной и космической физики (США), сотрудники которой совершенствовали существующие и разрабатывали новые программы для обработки и визуализации научных данных. Работу в этом направлении поручили Дэвиду Стерну (David Stern), который создал язык программирования Rufus. В 1977 г. Дэвид Стерн организовал собственную фирму Research Systems, Inc. (RSI), которая стала заниматься разработкой языка программирования для решения научных задач IDL. В 1977 г. вышла первая версия языка IDL. В 1992 г. в язык было введено понятие «виджет» (widget — элемент графического интерфейса), благодаря чему в 1994 г. появилось программное обеспечение ENVI, предназначенное для обработки данных ДЗЗ.

Язык IDL является кроссплатформенным языком программирования, что позволяет создавать приложения на различных платформах в различных опера-

ционных системах, таких как Microsoft Windows, Linux и другие.

При проведении исследований, как правило, используется большой объем данных, которые необходимо быстро обработать. Это требование стало ключевым для языка IDL. Оно воплотилось в IDL в виде простого синтаксиса, т. е. нет необходимости в составлении циклов и оптимизации времени выполнения функций для работы с массивами. Приведем простой пример: при обработке снимка ДЗЗ были получены двухмерные массивы данных R и NIR, в которых содержатся данные, полученные по данным красного и инфракрасного каналов, соответственно. В двухмерном массиве NDVI будут храниться значения вегетационного индекса NDVI. Различия в программной записи действий над массивами NIR и R для вычисления значений NDVI языков программирования C/C++ и IDL показаны в таблице, где i, j — индексы элементов массивов.

Язык IDL предлагает пользователям большое количество

разнообразных математических преобразований и функций, способных помочь решить достаточно сложные задачи. Для более детального анализа полученных результатов IDL обладает мощными средствами визуализации данных.

Визуализация данных в IDL поддерживает два типа рендеринга (rendering — создание трехмерного изображения с учетом теней, отражений и прочих световых эффектов): программный рендеринг и рендеринг с использованием аппаратных 3D-ускорителей, которые поддерживают библиотеку OpenGL. В большинстве случаев визуализация данных через OpenGL обладает лучшим качеством изображения, чем визуализация через программный рендеринг. Но следует учесть следующий аспект программного рендеринга: он может быть единственным доступным решением для систем, в которых нет аппаратных 3D-ускорителей, например, при использовании несовместимых OpenGL X серверов этот тип рендеринга строит одноразовую визуализацию быстрее, чем OpenGL и позволяет избежать возможных ошибок, возникающих с драйверами устройств.

IDL позволяет работать с файлами различных форматов,

Синтаксические отличия вычисления NDVI

C/C++

$$NDVI[i, j] = (NIR[i, j] - R[i, j]) / (NIR[i, j] + R[i, j])$$

IDL

$$NDVI = (NIR - R) / (NIR + R)$$

в том числе с графическими, например, BMP, JPEG, PNG, форматами, разработанными специально для научных задач, например, HDF, CDF, NCDF и несколькими распространенными форматами хранения данных, например, XML, ASCII. Существует возможность работы с данными в формате DICOM (Digital Imaging and Communications in Medicine). Такой большой набор поддерживаемых форматов файлов дает возможность программисту сконцентрироваться непосредственно на разработке проекта, не задумываясь о чтении или записи данных.

Работать с IDL можно в интерактивном или компилируемом режиме. Интерактивный режим используется для решения простых задач. В этом режиме предусмотрена командная строка IDL, в которой вводятся и выполняются команды сразу после нажатия клавиши «Enter». Интерактивный режим целесообразно использовать при разработке функций или процедур, чтобы мгновенно увидеть реакцию системы на выполненные действия. Компилируемый режим предназначен для разработки сложных приложений. Ярким примером применения компилируемого режима является программное обеспечение ENVI, которое представляет собой программу, написанную на языке IDL; при этом возможности ENVI не ограничены встроенными функциями. Задачи, решаемые с помощью ПО ENVI, можно расширять за счет разработки собственных дополнительных приложений на языке IDL, которые могут быть легко встроены в ENVI.

IDL располагает средством IDL GUIBuilder, позволяющим создавать графический интерфейс, и на его основе формировать исходный код IDL-программы, которая описывает интерфейс и события, возникающие в нем. IDL GUIBuilder со-

здан только для Windows версии IDL. Однако исходный код, образованный IDL GUIBuilder, является кроссплатформенным и работает на платформах, поддерживаемых IDL. IDL GUIBuilder обладает стандартными для подобного программного обеспечения (Microsoft Visual Studio, JBuilder) элементами построения графического интерфейса. Примеры построения изолиний, поверхностей и их совмещения с помощью интерфейса IDL GUIBuilder приведены на рис. 1 и 2.

Программы, разработанные на языке IDL, можно вызывать из внешних программ, созданных на других языках программирования или в IDL-программах использовать функции, реализованные на C/C++, FORTRAN, Java. Такое взаимодействие IDL с языками программирования позволяет гибко подходить к решению сложных задач. У IDL существуют особенности, которые явно отличают его от других языков программирования, таких как C/C++, Java.

Язык IDL не является сильно типизированным языком. Поэтому при выполнении программы тип переменной может изменяться в зависимости от значения, хранящегося в ней. Язык не чувствителен к регистру символов, т. е. переменные **A** и **a** являются одной и той же переменной. Нагляднее эти особенности продемонстрированы в приведенном ниже фрагменте программы:

«PRO PROPERTIES

```
A=1
HELP,A
A=-3.14
HELP,A
A='Строка'
Help,a
```

END».

В результате выполнения этой программы можно убедиться, что переменная **A** в начале программы после присваивания ей значения, равного «1»,

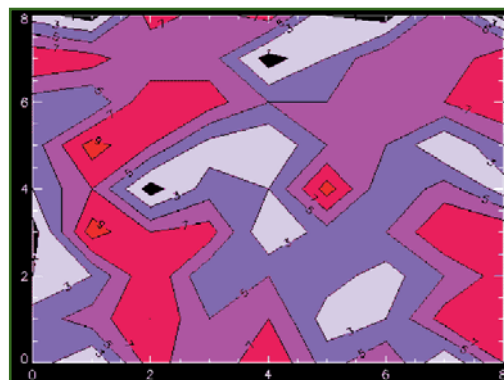


Рис. 1
Построение изолиний

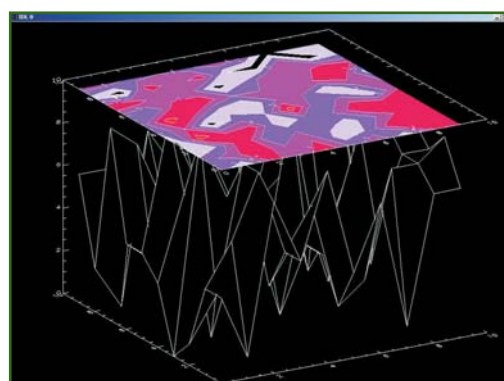


Рис. 2
Совмещение поверхности и изолиний

принадлежит целочисленному типу INT. Однако далее видно, что тип INT был изменен на FLOAT и STRING после присвоения соответствующих значений. Кроме того, на этом примере показано, что регистр для имен переменных, процедур и функций, а также ключевых слов (в данном случае только процедура **help** и ключевое слово **END**) значения не имеет.

RESUME

It is marked that the ENVI software package capabilities for remotely sensed data processing can be complemented with the software modules written in the IDL programming language. The article introduces the IDL programming language capabilities as well as shows its main distinctions from the other programming languages. IDL significantly simplifies programming for remotely sensed data processing.