

## ЯЗЫКОВОЙ СЕРВИС FRIS ДЛЯ ЭФФЕКТИВНОЙ РАЗРАБОТКИ FORTRAN-ПРИЛОЖЕНИЙ

*И. С. Раткевич, Ю. Г. Бартнев, С. С. Касаткин*

Российский федеральный ядерный центр –  
Всероссийский НИИ экспериментальной физики, г. Саров

В настоящее время для большинства языков программирования существует множество интегрированных сред разработки, которые позволяют решать самые разнообразные задачи: начиная от написания текста, как правило, с использованием технологии быстрой разработки приложений (RapidApplicationDevelopment – RAD) и заканчивая отладкой и автоматическим тестированием полученного приложения.

Для программирования в операционной системе Windows наиболее распространенной и часто используемой средой разработки является MicrosoftVisualStudio (VS). Она поставляется с интеграциями для языков программирования, поддерживаемых Microsoft: C++, C#, VisualBasic и т. д. Отметим, что VS можно дополнить любым языком программирования, который следует определенному набору соглашений (реализует заданные интерфейсы). Однако наиболее распространенные интеграции FortranотIntel [1] и PGI [2] реализуют лишь часть из них. Именно такая ощутимая разница между Fortran и базовыми языками от Microsoft явилась основанием для разработки FRIS.

FRIS – это расширение для MicrosoftVisualStudio, которое вводит полную поддержку базовых возможностей технологии IntelliSense для языка программирования Fortran с использованием стандарта Fortran-2003 [3]. Его основной целью является поддержка быстрой разработки приложений.

**Ключевой технологией, которая используется в VisualStudio для поддержки быстрой разработки приложений, является IntelliSense.** IntelliSense – это обобщенный термин для обозначения возможностей поддержки [4]:

- списка членов типов данных (ListMembers);
- сведений о параметрах подпрограмм и функций (ParameterInfo);
- кратких сведений об элементе языка программирования (QuickInfo);
- завершения слова или авто дополнения (CompleteWord).

Они ускоряют процесс программирования, позволяя улучшить качество разрабатываемых программ и уменьшить время разработки за счет обеспечения широких возможностей контекстуальной помощи.

### Поддержка Fortran в VisualStudio

Рассмотрим, как они реализованы в наиболее распространенных интеграциях от Intel и PGI (табл. 1).

Таблица 1

Обзор реализации базовых возможностей IntelliSense у Intel и PGI

Возможность	Intel	PGI
Построение списка членов типов данных	Отсутствует	Отсутствует
Отображение сведений о параметрах подпрограмм и функций	Есть, за исключением методов связанных с типом данных, и перегруженных методов с использованием родового интерфейса	Только для встроенных подпрограмм и функций
Отображение кратких сведений об элементе языка программирования	Есть, за исключением полей и методов производных типов данных	Только для встроенных подпрограмм и функций
Автодополнение	Для имен модулей и подпрограмм и функций	Только для ключевых слов

Заметим, что ни одна возможность не реализована полностью. При этом именно возможности IntelliSense повсеместно используются при написании исходного кода программ.

**Основные возможности и особенности FRIS.** На данный момент FRIS (версия 0.9.6.0) интегрируется в Microsoft Visual Studio 2005/2008/2010. Для удобства выбора нужной конфигурации и интеграции FRIS в Visual Studio был создан MSI (Microsoft Installer) пакет установки.

К основным возможностям, реализованным во FRIS, относятся:

- подсветка синтаксиса;
- выделение структурных элементов кода;
- поддержка панели навигации в текущем документе;
- поддержка XML комментариев документирования.
- все функции технологии IntelliSense:
  - поддержка списка элементов производных типов данных;
  - поддержка списков автодополнения;
  - отображение сведений о параметрах подпрограмм и функций;
  - отображение кратких сведений об элементе языка программирования;
- поддержка перехода к определению элемента языка программирования;
- поддержка работы со сниппетами исходного кода;
- поддержка работы с панелью Список ошибок (ErrorList);
- расширенная поддержка библиотек программ, заключающаяся в выделении цветом элементов библиотек в файле исходного кода:
  - встроенных функций и подпрограмм Fortran-2003;
  - библиотек: UPC-ОФ [5] – расчет уравнений состояния, ЕФР [6] – параллельный ввод-вывод сеточных данных.

Отличительной особенностью FRIS является обработка программы, которую редактирует программист в Visual Studio, в реальном времени. В подавляющем большинстве случаев в момент редактирования текст программы является недопустимым с точки зрения лексических и синтаксических правил языка. Это представляет значительную сложность для реализации соответствующих анализаторов, поскольку они, как правило, ориентированы на работу с полными грамматиками, предполагая, что тексты анализируемых программ не содержат ошибок. При этом в случае обнаружения ошибки такие полные анализаторы прекращают дальнейший разбор файла исходного кода.

Во FRIS эту сложность преодолевает оригинальная стратегия лексического, синтаксического и семантического разбора. Анализаторы, разработанные и используемые во FRIS, построены с использованием генератора анализаторов ANTLR [7] и используют его особенности, в частности разбор, осуществляется при помощи стратегии LL(\*) [7], используется разбор с возвратами, сохранением обнаруженных альтернатив разбора и использованием предикатов.

При лексическом анализе неизвестные символы исключаются из входного потока и токены для них не формируются. При синтаксическом анализе, когда фактическая последовательность токенов не соответствует ни одному из возможных правил, возникает ошибка разбора, которую перехватывает FRIS и использует специально разработанное правило, которое может обработать такую недопустимую последовательность токенов. После этого разбор продолжается далее.

*Подсветка синтаксиса* (рис. 1) – это выделение цветом различных смысловых частей исходного кода: ключевых слов, строковых литералов, комментариев и т. п. Отличительной особенностью FRIS является предоставление выделения цветом элементов библиотек конечного пользователя, а также XML-комментариев документирования.

```

{} (глобальная область видимости)
globalsub1(a, b, c)
!!!<summary>
  !!!Модуль для тестирования работы FRIS
  !!!</summary>
module TestMod
  implicit none
  #Директива препроцессора
  !Комментарий
  !!!<summary>Комментарий документирования</summary>
  character(255) :: str = "Строка"
  !!!<summary>Пример BOZ констант</summary>
  integer(4) :: myInt = b'11' + o'87' + z'FA01'
  !!!<summary>Пример логической константы и оператора</summary>
  logical(4) :: myFlag = .false. .and. .true.

```

Рис. 1. Текст редактора с использованием языкового сервиса FRIS

*Выделение структурных элементов* (см. рис. 1) позволяет наглядно представить структуру исходного кода путем объединения в один блок парных конструкций, например, начало и конец подпрограммы, модуля, условного оператора или оператора цикла и т.п.

*Панель навигации* (см. рис. 1) позволяет быстро переходить к интересующим элементам, расположенным в текущем файле. Она состоит из двух выпадающих списков. Левый предназначен для отображения информации об областях видимости (модулях и типах), а правый – об их членах.

*Поддержка XML комментариев документирования* является одной из ключевых возможностей FRIS. Комментарий документирования – это комментарий специального вида, который используется для документирования программного кода, как правило, его содержимое включается в выводимые средой разработки контекстные подсказки.

Для того чтобы комментарий был распознан и включен FRIS в выводимые IntelliSense подсказки, он должен иметь следующий вид:

```

!!!<ИмяТэга Атрибут1 = «Значение атрибута»>
!!!Текст комментария...
!!!</ИмяТэга>

```

FRIS трактует любой комментарий, начинающийся с последовательности из 3-х восклицательных знаков – «!!!» как комментарий документирования, за которым может следовать один или несколько XML тэгов документирования. Например, тэг<summary></summary> – служит для описания краткой общей информации. Полный список поддерживаемых комментариев документирования приведен на рис. 2.

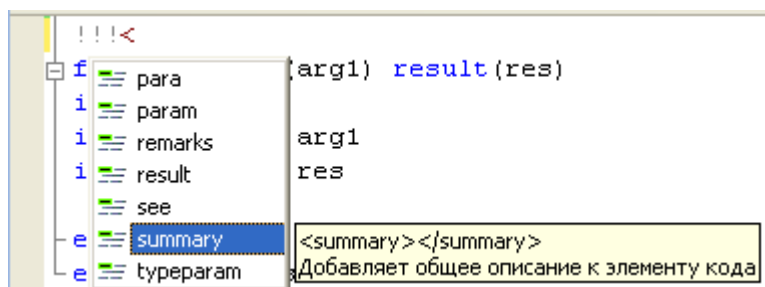


Рис. 2. IntelliSense для комментариев документирования

FRIS поддерживает возможность *построения списка элементов*, определенных в различных типах данных, с учетом механизма наследования типов. Поддерживается работа со всеми компонентами производных типов данных, включая связанные с типом процедуры. На рис. 3 показана

работа данной возможности для элементов данных производного наследуемого типа данных. При этом в список его членов включен особый компонент **SecondType** – это ссылка на базовый тип данных. На рис. 4 показан пример работы с процедурами, связанными с типом данных.

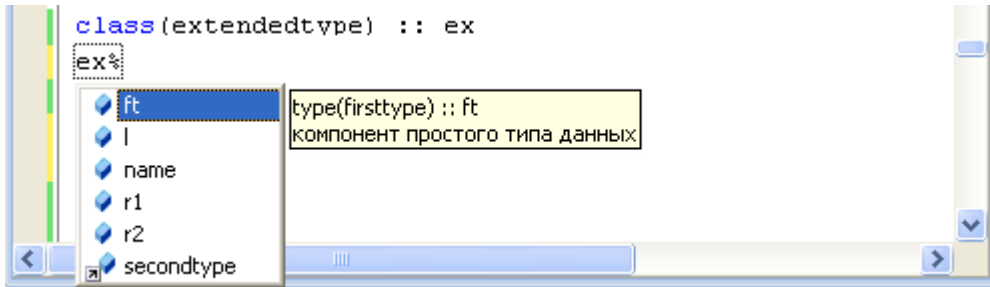


Рис. 3. Список компонентов наследуемого производного типа

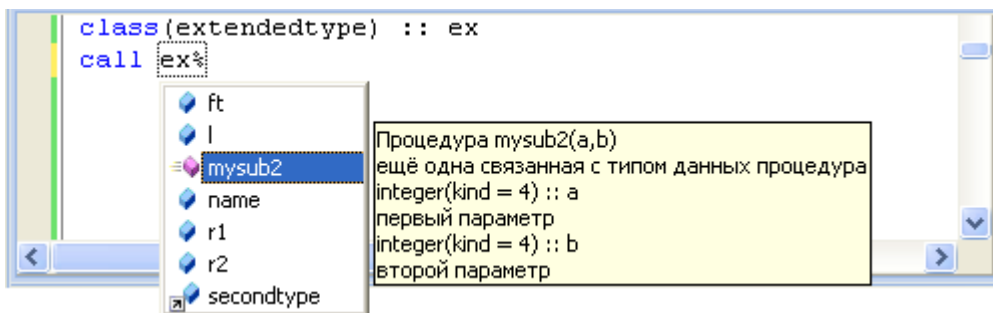


Рис. 4. Работа со связанными с типом данных процедурами

**Автодополнение** является основополагающей возможностью для быстрой разработки приложений. Для отображения списков автодополнения необходимо нажать комбинацию клавиш **<Ctrl>+<Space>**. Во FRIS реализованы следующие возможности автодополнения:

- для имен модулей;
- для имен производных типов данных;
- для имен переменных, доступных в текущей области видимости (см. рис. 5);
- для имен подпрограмм.

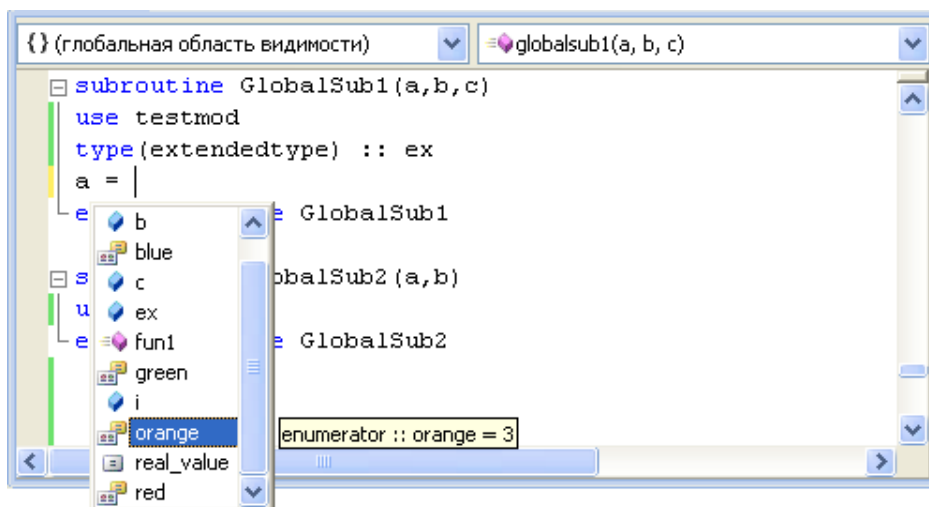






Рис. 5. Автодополнение имен переменных, доступных в текущей области видимости, для правой части оператора

В правой части оператора могут появиться переменные, константы, в том числе члены перечислений (enum), и функции. Значения иконок:  – переменная,  – функция,  – член перечисления,  – именованная константа.

**Отображение сведений о параметрах** подпрограмм и функций (методов) позволяет работать с методами, точную сигнатуру которых Fortran программист может и не знать. Вся необходимая информация о наличии перегруженных версий метода, о его аргументах, их количестве, названиях и типах данных, будет отображена во всплывающей подсказке (рис. 6). FRIS поддерживает работу с родовыми интерфейсами, когда для группы подпрограмм или функций объявляется одно имя (рис. 7).

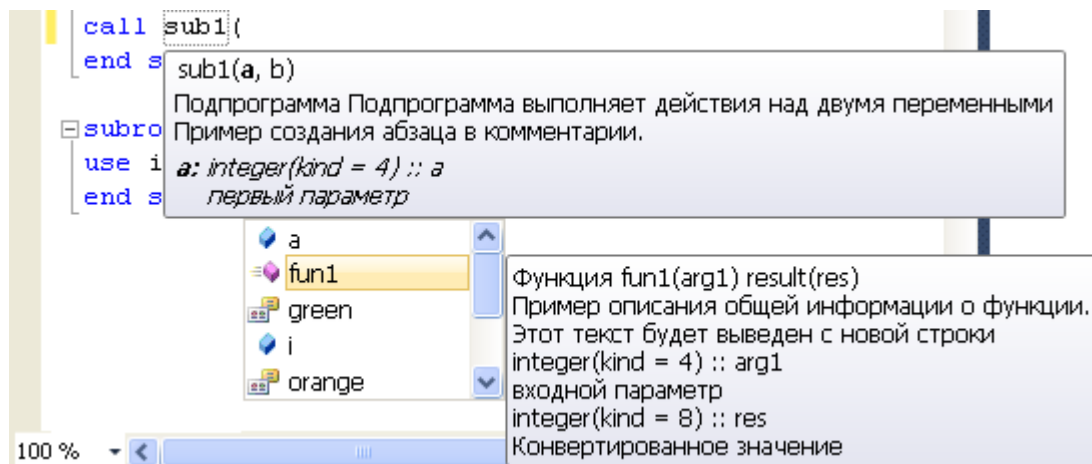


Рис. 6. Отображение параметров подпрограммы и автодополнение ее аргументов

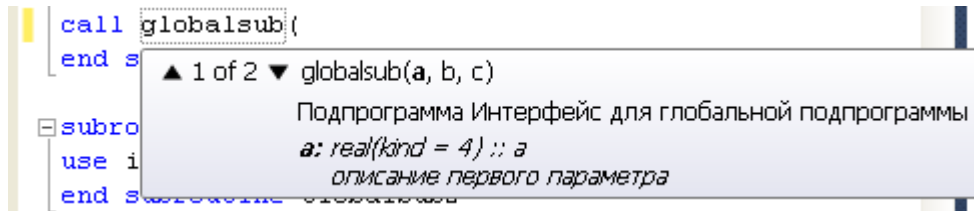


Рис. 7. Отображение параметров для родового (обобщенного) интерфейса

К **отображению кратких сведений об элементе языка программирования**, относится любая контекстная помощь, отображаемая в виде подсказок. В контекстные подсказки, выводимые FRIS, включаются не только информация, полученная из определения элемента языка программирования, но и информация, полученная из комментариев документирования (рис. 8).

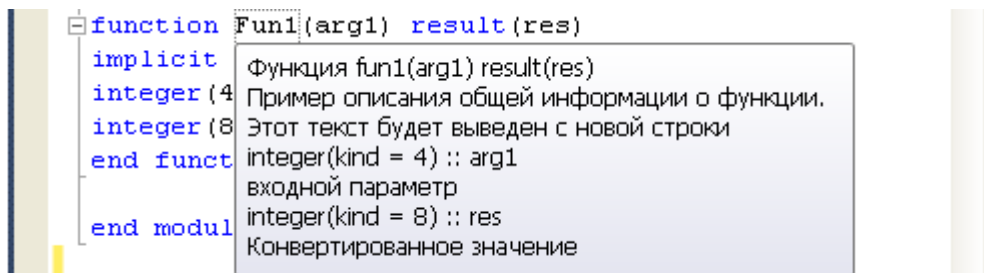


Рис. 8. Отображение информации о функции

Довольно часто встречающейся на практике задачей является *переход к определению* того или иного элемента языка программирования для получения наиболее полной информации о его внутреннем устройстве (рис. 9).

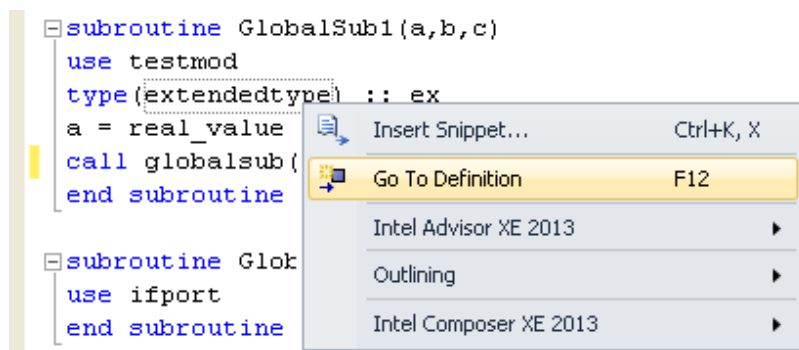


Рис. 9. Переход к определению производного типа данных

FRIS поддерживает *работу со сниппетами (фрагментами) исходного кода*. Сниппет представляет собой некоторый шаблонный фрагмент исходного кода, в котором могут содержаться пустые, предназначенные для заполнения места. Создание и использование таких шаблонов существенно ускоряет процесс разработки программного обеспечения.

Использовать сниппеты исходного кода можно несколькими способами. Самый простой – поместить курсор на новую строку и нажать **<Ctrl>+<Space>**. В результате этого будет выведен список автодополнения содержащий все доступные сниппеты (рис. 10).

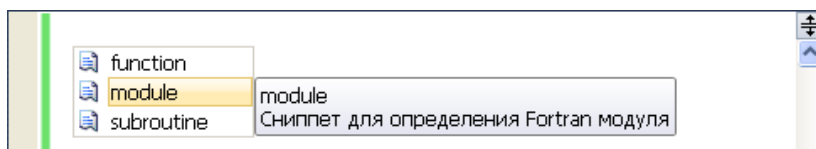


Рис. 10. Сниппеты исходного кода в списке автодополнения

Результат работы вставки сниппета FRIS для модуля показан на рис. 11. Отметим, что сниппеты FRIS снабжены заготовками для комментариев документирования.

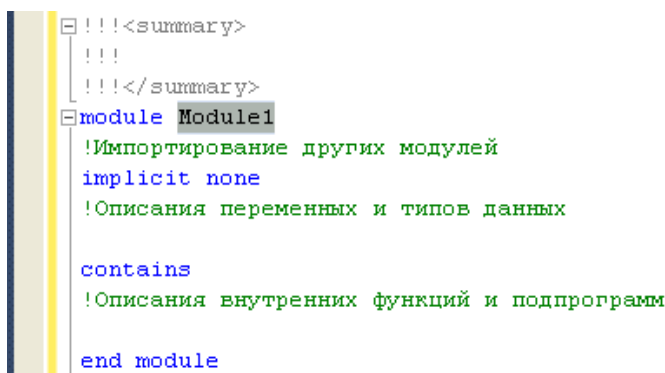


Рис. 11. Вставка сниппета FRIS для модуля

Сразу после вставки сниппет находится в режиме редактирования. В этом режиме пользователь может заполнить шаблонные части нужными значениями. Переход между шаблонными частями осуществляется при помощи клавиши **<Tab>**, ввод шаблонной части заканчивается нажатием клавиши **<Enter>**.

FRIS поддерживает работу с панелью *Список ошибок (ErrorList)*, который предназначен для вывода сообщений об обнаруженных ошибках. Сейчас в него выводятся предупреждения, относящиеся к работе с комментариями документирования, а также грубые синтаксические ошибки (рис. 12).

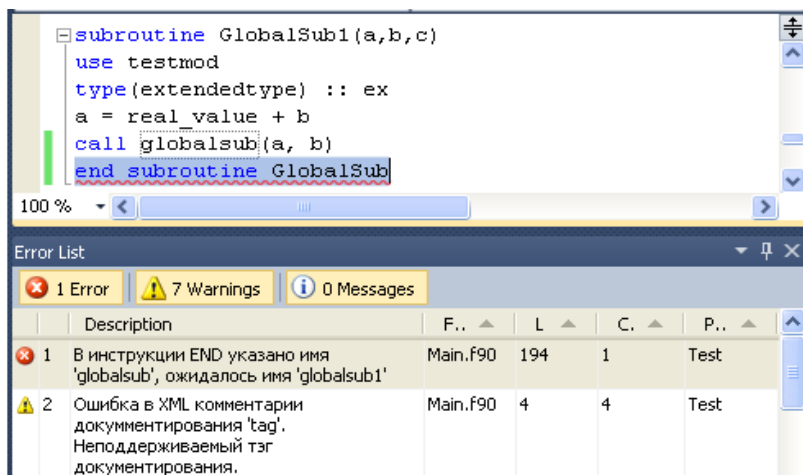


Рис. 12. Поддержка работы со Списком ошибок

**Общий механизм поддержки произвольных библиотек во FRIS.** Особенностью FRIS является возможность поддержки произвольных библиотек, которая состоит из:

- 1) предоставления контекстно-зависимой помощи для элементов библиотеки;
- 2) обозначение цветом элементов библиотеки.

Первая задача решается автоматически, если определения модулей, производных типов данных, интерфейсов, подпрограмм и функций библиотеки содержатся в файлах исходного кода, непосредственно подключаемых к программному проекту. Применительно к библиотекам ИТМФ таким способом подключаются библиотеки УРС-ОФ и ЕФР.

Если для подключения библиотек используются не файлы исходного кода, а например, двоичные \*.mod и \*.lib файлы, то используются специальные XML файлы, в которых содержится описание необходимых программных сущностей: модулей, подпрограмм, функций, типов данных и т. п. Типичным примером подобного подключения являются встроенные в язык подпрограммы и функции.

```

<?xmlversion="1.0" encoding="utf-8" ?>
<doc>
<!--Указываем все элементы, для которых есть документация-->
<members>
    <!--Документация для одного элемента-->
<!--13.7.2 ACHAR (I [, KIND])-->
<membername="ACHAR(I, [KIND])">
<summary>
    Возвращает символ, находящийся в указанной позиции в схеме упорядочения ASCII. Это инверсия значения функции IACHAR.
</summary>
<paramname="I">
    Номерпозиции, типinteger
</param>
<paramname="KIND">
    Необязательный параметр, размерность типа возвращаемого значения
</param>
<result>Символ character(len=1,kind=[kind])</result>
</member>
</members>
</doc>

```

Одними из наиболее часто используемых библиотек в комплексах программ ИТМФ являются библиотеки УРС-ОФ и ЕФР. Целью FRIS является обеспечение их расширенной поддержки.

**Поддержка библиотеки УРС-ОФ.** Библиотека УРС-ОФ используется многими программными комплексами ИТМФ для расчета заданного уравнения состояния вещества в каждой точке при моделировании различных изделий.

Для обеспечения поддержки библиотеки УРС-ОФ во FRIS разработчиками библиотеки УРС-ОФ был модифицирован способ подключения библиотеки, и задокументированы с использованием комментариев документирования FRIS основные элементы библиотеки.

Со стороны FRIS был реализован расширенный механизм подсветки синтаксиса для библиотеки УРС-ОФ (рис. 13).

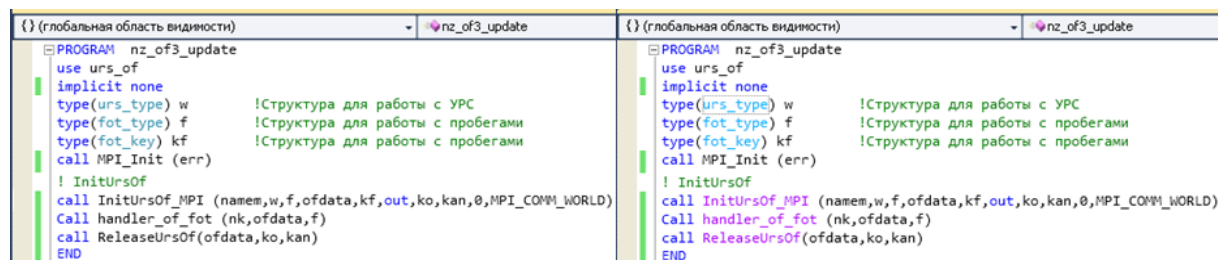


Рис. 13. Подсветка синтаксиса УРС-ОФ (слева отсутствует, справа присутствует)

**Поддержка библиотеки ЕФР.** Библиотека ЕФР также является «общей частью» многих программных комплексов ИТМФ, обеспечивая обмен с внешней памятью ЭВМ распределенных сеточных данных в параллельном режиме.

Для обеспечения поддержки библиотеки ЕФР во FRIS, разработчиками библиотеки ЕФР были задокументированы, с использованием комментариев документирования FRIS, основные элементы библиотеки (рис. 14).

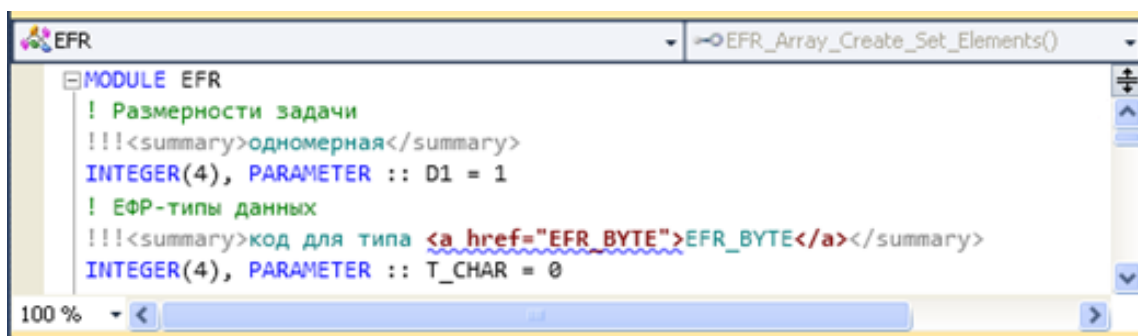


Рис. 14. Использование комментариев документирования FRIS в ЕФР

Далее приведены примеры работы различных возможностей FRIS для ЕФР:

- на рис. 15 приведен пример автодополнения функций библиотеки ЕФР;
- на рис. 16 приведен пример отображения аргументов функции;
- на рис. 17 приведен пример работы с перегруженной подпрограммой.



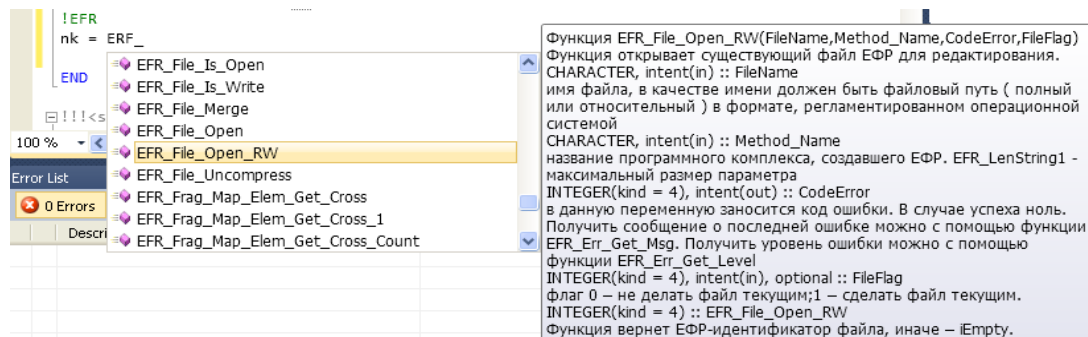


Рис. 15. Автодополнение имени функции ЕФР

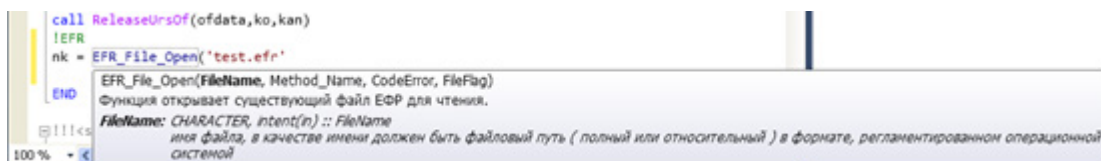


Рис. 16. Сведения об аргументах функции ЕФР по мере их ввода

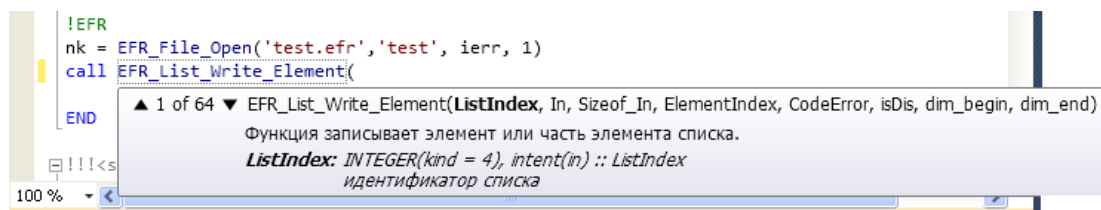


Рис. 17. Работа с перегруженными подпрограммами ЕФР

## Поддержка параллельного программирования: стандарты MPI и OpenMP

В настоящее время для создания параллельных программ наиболее часто используются следующие стандарты (или их комбинации):

- MPI – для распараллеливания на распределенной памяти;
- OpenMP – для распараллеливания на общей памяти.

Во FRIS планируется реализовать поддержку их самых современных версий: MPI 3.0 [8] и OpenMP 4.0 [9]. Это особенно важно ввиду расширяющего совместного внедрения обеих технологий параллельного программирования, существенно усложняющих структуру программы.

**Поддержка стандарта MPI 3.0.** Стандарт MPI служит для обеспечения передачи сообщений между параллельно работающими процессами при решении какой-либо задачи высокопроизводительных вычислений и предоставляет четко определенный в тексте стандарта набор подпрограмм, разделенных по следующим классам:

- 1) Программы для коммуникаций типа «точка-точка».
- 2) Программы для коллективных коммуникаций.
- 3) Служебные программы для работы с коммутаторами и группами процессов.
- 4) Программы для создания топологий.
- 5) Программы для работы с окружением.
- 6) Программы для порождения и управления процессами.
- 7) Программы для односторонних коммуникаций.
- 8) Программы ввода/вывода.

А также ряд производных типов данных и именованных констант. Соответственно, для поддержки MPI 3.0 со стороны FRIS необходимо:

- 1) создать XML файл с описанием подпрограмм, функций, типов данных и именованных констант, руководствуясь текстом стандарта;
- 2) определить набор цветов для визуального выделения подпрограмм и функций, а так же типов данных MPI.

**Поддержка стандарта OpenMP 4.0.** Стандарт OpenMP 4.0 предназначен для обеспечения параллельной работы выделенных блоков программы в рамках одного процесса в многоядерной среде.

Отличительной особенностью стандарта OpenMP является то, что основное его применение выполняется при помощи написания директив – специального рода комментариев, имеющих строго определенный синтаксис, а подпрограммы и функции в нем выполняют скорее дополнительную роль. Следовательно, для поддержки данного стандарта со стороны FRIS необходимо:

- 1) Реализовать лексический и синтаксический анализатор директив OpenMP во FRIS. Это, безусловно, наиболее трудоемкая часть реализации поддержки OpenMP.
- 2) Создать XML файл с описанием подпрограмм и функций, а также именованных констант, руководствуясь текстом стандарта.
- 3) Определить набор цветов для визуального выделения подпрограмм, функций и директив OpenMP.

## Заключение

Языковой сервис FRIS предназначен для поддержки технологии быстрой разработки Fortran приложений. Отличительной особенностью FRIS является ориентированность на удобство конечного пользователя. Разработанная во FRIS расширенная поддержка XML-комментариев документирования позволяет получать контекстно-зависимую помощь об используемых элементах языка программирования без необходимости обращения к специализированным справочным системам. Механизм работы со сниппетами исходного кода позволяет пользователю создавать и повторно использовать полноценные библиотеки шаблонов кода, а их включение в списки автодополнения обеспечивает мгновенный доступ к ним.

Особым видом адаптации к нуждам конечного пользователя является обеспечение расширенной поддержки пользовательских библиотек программ. В частности, FRIS может предоставлять возможности визуального выделения тех или иных элементов в файле исходного кода для акцентирования внимания пользователя. Для этого реализовано два универсальных подхода: первый из них заключается в автоматической обработке файлов исходного кода на языке Fortran, а второй – в создании специальных XML файлов с описанием программных компонентов экспортируемых библиотек.

В ИТМФ такая расширенная поддержка со стороны FRIS реализована для библиотек УРС-ОФ и ЕФР. В дальнейшем будет добавлена поддержка стандартов параллельного программирования MPI 3.0 и OpenMP 4.0.

Необходимо отметить, что FRIS активно используется в ИТМФ при разработке Fortran-приложений и существенно повышает эффективность работы за счет предоставления различных видов контекстной помощи. Ранее Fortran-программист был вынужден помнить имена всех используемых им функций, модулей, переменных и типов данных наизусть либо проводить полнотекстовый поиск для их уточнения. Использование возможностей, реализованных во FRIS особенно полезно при командной работе над одним проектом, когда множество программистов должны взаимодействовать между собой. В этом случае автоматизация выбора необходимых программных элементов (функций, подпрограмм, переменных и т. п.) существенно упрощает работу, т.к. не нужно обращаться к их определениям или системе справочной документации для их корректного использования.

### Литература

1. Intel® Fortran Composer XE 2013 SP1 Release Notes. [Electronic resource]. Mode of access: <http://software.intel.com/en-us/articles/intel-fortran-composer-xe-2013-sp1-release-notes>.
2. PGI Visual Fortran. [Electronic resource]. Mode of access: <http://www.pgroup.com/products/pvf.htm>.
3. Information technology – Programming languages – Fortran – Part 1: Base Language ISO/IEC 1539-1:2004. [Electronic resource].
4. Using IntelliSense. [Electronic resource]. Mode of access: [http://msdn.microsoft.com/en-us/library/hcwl1s69b\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/hcwl1s69b(v=vs.110).aspx).
5. Гордеев Д. Г., Голубкова Е. Ф., Гударенко Л. Ф. и др. Современное состояние пакета программ УРС-ОФ для расчета термодинамических и механических свойств веществ / Под ред. Р. М. Шагалиева // Труды XII Международного семинара «Супервычисления и математическое моделирование» Саров: РФЯЦ-ВНИИЭФ, 2011.
6. Волгин А. В., Тарасов В. И., Красов А. В., Кузнецов М. Ю. Библиотека ЕФР для универсального представления расчетных данных // Труды РФЯЦ-ВНИИЭФ. 2007. Вып. 11.
7. Terence Parr. Language Implementation Patterns. The Pragmatic Bookshelf, Raleigh, NC and Dallas, TX, 2009.
8. MPI: A Message-Passing Interface Standard Version 3.0; Message Passing Interface Forum; September 21, 2012
9. OpenMP Application Program Interface Version 4.0; OpenMP Architecture Review Board; July 2013.

## ВОЗМОЖНОСТИ УЧЕТА НАПРЯЖЕННО-ДЕФОРМИРОВАННОГО СОСТОЯНИЯ КОНСТРУКЦИЙ ПРИ РЕШЕНИИ ЗАДАЧ МОДАЛЬНОГО АНАЛИЗА В ПП ЛОГОС-ПРОЧНОСТЬ

*В. Н. Речкин, А. И. Чембаров, С. С. Косарим, О. Л. Александрова*

Российский федеральный ядерный центр –  
Всероссийский НИИ экспериментальной физики, г. Саров

Одним из направлений развития пакета программ (ПП) ЛОГОС-ПРОЧНОСТЬ является разработка программных средств для решения задач на определение собственных частот и форм колебаний конструкций, в том числе и с учетом их напряженно-деформированного состояния (НДС) [1]. Решение таких задач является важной составной частью расчетных исследований динамического отклика конструкций при действии вибрационных нагрузок.

В некоторых случаях, когда возникающее в конструкции НДС приводит к существенному изменению жесткости отдельных элементов, расчет ее основных собственных частот и форм колебаний необходимо проводить с учетом этих эффектов. Известно, что значения собственных частот изгибных колебаний предварительно растянутой балки выше по сравнению с их значениями для ненагруженной балки. И наоборот, наличие предварительных напряжений сжатия в балке приводит к уменьшению значений собственных частот ее изгибных колебаний. Такие особенности проявляются в основном в тонкостенных или рамных конструкциях, изгибная жесткость отдельных