|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение высшего образования.  «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»  (МГТУ им. Н.Э. Баумана) | | ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»  КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА»  Домашнее задание  по дисциплине  «Парадигмы и конструкции языков программирования»  НА ТЕМУ:  Первый язык программирования.  Студенты ИБМ3-34Б Ю.В.Срывалина  (Подпись, дата)      Преподаватель Ю.Е. Гапанюк  (Подпись, дата)  2024 г. | | |

**Первый язык программирования.**

Языки программирования — это средство коммуникации между человеком и машиной. Рассказываем, как развивались языки программирования, чем отличаются друг от друга, что на них влияет и что ждет в будущем

Первым высокоуровневым языком программирования считается **FORTRAN**. Он был создан в 1954 году группой программистов под руководством Джона Бэкуса в корпорации IBM. Этот язык отличается от своих предшественников, языков низкого уровня, тем, что не был привязан к конкретному типу ЭВМ. Это позволило использовать язык на разных типах машин и сделало его очень популярным. FORTRAN был разработан специально для научных и инженерных вычислений. Его эффективность в обработке математических задач и умение работать с массивами данных сделали его неотъемлемым инструментом в таких областях, как аэрокосмическая промышленность, физика, метеорология и других научных сферах.

Вначале, название этой системы писалось исключительно большими буквами: FORTRAN. Его применяли для различных версий до появления Fortran 77. Для бесплатных вариантов синтаксиса новое написание использовалось, начиная с Фортран 90. Особенность версии Fortran 77 заключается в том, что в состав языка не входили строчные символы, но в качестве дополнения многие компиляторы их могут поддерживать.

* Первая программа, которая компилировала текст, написанный на языке Фортран, в набор машинных кодов разрабатывалась с нуля в период с 54 по 57 года прошлого века. Тогда еще не были созданы высокоуровневые (= HLL) символьные системы. В основном все ОС отличались простотой, а память компьютеров не превышала 16 Kb. Тот компилятор был ориентирован на IBM 704. Система HLL оказалась более производительной, чем ассемблер, что и способствовало ее быстрому распространению.
* Разработка Fortran завершилась в 1958 году. Тогда же появился и третий Фортран, но эта версия еще не была широко представлена.
* Фортран IV появился уже в 1961 году. Он включал ряд улучшенных разработок, в том числе, были реализованы операторы COMMON и EQUIVALENCE.
* Еще через год агентством стандартизации США были развернуты работы по разработке стандартов для объективно ориентированной системы Фортран. Это был необходимый шаг для того, чтобы этот язык программирования мог использоваться на всех новых компьютерах. В результате Fortran получил еще большую популярность и был доступен на Apple и TRS80.
* Первый международный стандарт HLL получил реализацию в языке FORTRAN 66 в 1967 году. Благодаря его публикации, ФОРТРАН по широте своего применения намного опередил другие языки того времени. В средине 1970-х годов прошлого века эта система уже использовалась почти на всех ПК, мини или мэйнфреймах. В языке программирования FORTRAN 66 применялись утверждение if, goto-statement и spagethi-программы. Аналогичное структурированное написание программ получило особую популярность в шестидесятых и семидесятых годах.

Система Fortran работала на перфорированных картах, в том числе и с платформой FMS. Разработчикам приходилось оптимизировать размещение источников языка до того момента, когда в Fortran 90 появился «свободный» синтаксис. В этой системе началом кода массива, который не превышал 72000 символов, был 7-й столбец.

При этом, пока не появилась версия языка программирования Fortran 90 не учитывались пробелы, расположенные между 7-м и 72-м столбцами. Другими словами, цикл «DO I = 1.5» мог представляться в форме записи. В то же время, «DO I = 1,5» имеет значение аналогичное с «DOI = 1.5».

Большое количество промышленных кодов писались в Nastran, NAG и IMSL. Важно, чтобы обновленные вариации языка программирования были совместимы с ранними версиями. Этому требованию полностью соответствовал Фортран 90. Эта версия совместима с Fortran 77. В дальнейшем при появлении новых разработок данное требование уже не соблюдалось.

Более современные версии языка программирования Fortran 90 и 95 были обновлены до стандарта Фортран 2003, который действует до настоящего времени. Отметим, что новые программы для компилирования в машинный код могут без ограничений использоваться в современных версиях и могут поддерживать даже 64-хразрядные процессоры. Кроме того, с учетом существующих тенденций разработчики создали компиляторы для Linux в форме объектно-ориентированной системы Actor Fortran.

**Структура** **языка программирования Fortran**

Программа на языке программирования Fortran представляет собой систему, составленную из элементов библиотеки, включающих основной программный текст, модули, внешние подпрограммы и процедуры.

Началом каждой компьютерной программы на языке Fortran выступает запись ключевого слова program, а в ее конце указывается end program, после чего прописано название программного продукта.

Неявная команда none дает возможность компилирующей программе выполнить проверку правильности объявления типов переменных. Следует в обязательном порядке в начале программы прописывать оператор none вначале каждой программы.

Для выделения комментариев в Fortran применяется установленная перед ними метка (!). Все следующие за восклицательным знаком символы или строки компилятор будет игнорировать.

Для вывода данных на монитор следует прописать оператор print \*.

Что программа была более удобной для чтения нужно использовать отступы строк кода.

В языке программирования Fortran могут использоваться прописные и строчные символы. Данная система не имеет чувствительности к регистрам, если это не строковые литералы.

Базовый комплект символов языка программирования Fortran включает

1. Буквенные символы А … Я и А … Я.

2. Цифровые обозначения 0 … 9.

3. Метка подчеркивания (\_)

4. Спецсимволы =: + пробел — \* / () [],. $ ‘! «% &; <>?

Обозначения включают символы основной библиотеки. Модуль аутентификации (токен) может выступать ключевым словом, идентификатором, постоянной, строковым литералом или символом.

Операторы заявления программы составляются из токенов.

Идентификатор — это имя, которое применяется для определения переменной, действия или других компонентов программы, которые определены пользователем.

Правила написания идентификатора в языке программирования Fortran:

1. Максимальное количество символов в идентификаторе — 31.

2. Имя может включать буквенно-цифровые знаки, указанные выше и подчеркивание (\_).

3. Начинаться идентификатор может только с буквенного символа.

4. У идентификатор нет чувствительности к регистру.

5. Ключевыми словами называют специальные, зарезервированные в библиотеке записи, которые нельзя применять в виде идентификаторов/имен.

**Достоинства языка программирования Fortran:**

* Очень простой и доступный вариант языка для обучения программированию. Система отличается понятным синтаксисом, а ее исторический опыт применения будет полезен для тех, кто только начинает знакомиться с разработкой программ.
* Освоив азы Fortran, будет не сложно разобраться в других языках.
* Система имеет обширный набор инструментов, доступных на бесплатной основе, поэтому не придется заморачиваться с лицензиями.
* Благодаря распространенности во всем мире, Фортран имеет объемную библиотеку и большой ассортимент прикладных приложений, которые созданы за его длинную историю.
* Эта система стандартизирована для разных платформ, а ее новые версия отличаются совместимостью с более ранними вариантами.
* Язык программирования Fortran имеет набор средств (трансляторов) для преобразования в машинные системы разных компьютеров.
* Практически на всех ПК присутствует компилятор Фортран, поэтому у пользователей всегда есть возможности для сложных параллельных вычислений.
* С помощью этой системы можно формировать компактные и эффективные программные коды, чем и была обусловлена ее востребованность в то время, когда ЭВМ еще не отличались особой производительностью.
* Язык программирования Fortran полезно изучать тем, кто получает высшее образование по техническим и, особенно, физико-математическим дисциплинам.

**Основные минусы языка программирования Fortran:**

* Жесткие требования к формату кода. К примеру, версия Фортран 77 имеет ограничения по длине строки, а в ее начале нужно было делать отступ. Этакие условия делали неудобной работу даже при использовании перфокарт, не говоря уже о мониторе ПК. Позже, начиная со стандарта Fortran 90, эти ограничения смягчили.
* Небольшой набор команд для управления программной структурой. К примеру, очень сложно было писать программные приложения, при отсутствии оператора GOTO.
* Слабый набор средств для описания данных.

В заключение, можно отметить, что Fortran, несмотря на свой почтенный возраст, остается актуальным и востребованным языком программирования, особенно в областях, требующих высокой производительности вычислений. Его сильные стороны, такие как эффективная работа с массивами, оптимизация для векторных и параллельных вычислений, и обширные библиотеки для научных и инженерных задач, остаются неоспоримыми. Хотя Fortran уступает современным языкам в некоторых аспектах, таких как поддержка объектно-ориентированного программирования (хотя современные стандарты уже включают некоторые его элементы) и разработка веб-приложений, его эволюция и развитие продолжаются, стремясь к совместимости с современными требованиями. Поэтому Fortran, вероятно, будет продолжать играть значительную роль в научных исследованиях и инженерных расчетах еще в течение длительного времени, особенно в нишах, где производительность вычислений является критическим фактором. Однако, необходимо признать и ограничения языка, и понимать, что для новых проектов, требующих широкого спектра функциональности и современных подходов к разработке, выбор других языков может быть более обоснованным.