Программирование

Лекция 1

Петров Александр Владимирович
Фёдоров Станислав Алексеевич

(по материалам Веренинова Игоря Андреевича с изменениями и дополнениями на Fortran 08 и UML)

Сентябрь 2013

Состав курса лекций

- Основы алгоритмизации
- Процедурное программирование на языке Fortran 08
- Объектно-ориентированное программирование на языке Fortran 08*

Задачи курса (знание)

- Принципов алгоритмизации многомодульных программных систем
- Механизмов передач параметров при работе с процедурами
- Разнообразных типов данных в строго типизированном языке высокого уровня
- Разнообразных динамических структур данных и их применения
- Принципов объектно-ориентированного программирования и механизмов работы с объектами в языке Fortran 08

Задачи курса (умение)

- Проводить декомпозицию задач
- Кодировать разработанные многомодульные алгоритмы с использованием принципов процедурного и объектного программирования
- Тестировать и отлаживать многомодульные программные системы, используя процедурный и объектный подход к разработке

Виды практических занятий

- Упражнения, на которых рассматриваются схемы алгоритмов и программы с применением различных типов и структур данных
- Лабораторные занятия, на которых рассматриваются и дополнительно тестируются преподавателем выполненные задания

Что надо сделать в первом семестре

- Сдать все домашние задания по алгоритмам, задаваемым на упражнениях
- Написать две контрольные работы :
  - на многомодульные алгоритмы
  - на динамические переменные
- Сделать и сдать 11 программ по индивидуальным заданиям

В результате прохождения курса Вы должны уметь

Разрабатывать и отлаживать хорошо структурированные многомодульные программы с применением различных типов и структур данных в процедурном и объектном исполнении на языке Fortran 08

Список литературы — структуры данных и алгоритмы

1. Алгоритмизация и структурное программирование . Учеб. пособие. СПб.: изд-во СПбГПУ, 2000, 56 с.
2. Веренинов И.А.. Программирование на языке высокого уровня. Объектно-ориентированное программирование на языке Turbo Pascal 7.0. Учеб. пособие. СПб.: изд-во СПбГПУ, 2004, 42 с.
3. Мартин Фаулер. UML. Основы : краткое руководство по стандартному языку объектного моделирования.— 3-е изд. — СПб. : Символ-Плюс, 2008.
4. Структуры данных и алгоритмы : Пер. с англ. / А.В. Ахо, Д.Э. Хопкрофт, Д.Д. Ульман .— Москва : Вильямс, 2003.
5. *Никлаус Вирт. Алгоритмы и структуры данных, 2011.
6. *Хьюз Дж. и Мичтом Дж.. Структурный подход к программированию, М.: Мир, 1980.
7. *Программирование на языке высокого уровня. СПб.: изд-во СПбГПУ, 2006, 212 с.. (только в отделе научной литературы)
8. *Лингер Р., Миллс Х., Уитт Б.. Теория и практика структурного программирования, М.:Мир,1992.

Список литературы — программирование на Fortran 08

1. Бартеньев О. В. Современный Фортран / О. В. Бартеньев .— Изд. 4-е, доп. и перераб .— М. : Диалог-МИФИ, 2005 .
2. Сборник задач: Основы программирования : Метод. указания/ Ленинградский политехнический институт им.М.И.Калинина ; Сост.И.А. Веренинов, В.А. Зимницкий, Л.К. Кириллова .— Ленинград, 1986.
3. M. Metcalf, J. Reid, M. Cohen - Modern Fortran Explained, 2011
4. N. Clerman, W. Spector - Modern Fortran. Style and Usage, 2012

Краткая история

FORmula TRANslation разработан командой под руководством Джона Бэкуса в 1954-1958 годах в IBM.

FORTRAN 66 (Стандарт ISO 1972)

FORTRAN 77 (1980)

Fortran 90 (1991)

Fortran 95 (1996)

Fortran 2003 (2004)

Fortran 2008 (2011)

Предварительные замечания

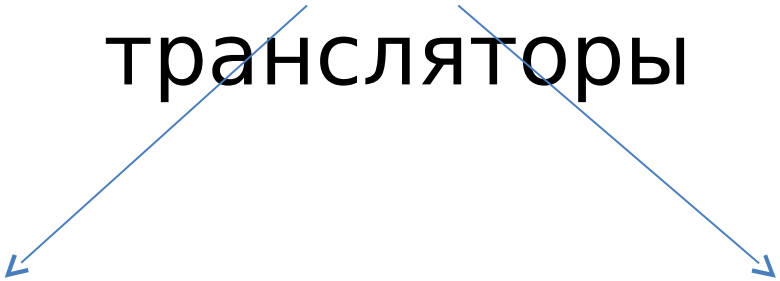
- Программирование — чрезвычайно широкая область инженерной деятельности, не менее широкая, чем строительство, машиностроение и пр., поэтому надо уточнить, про какие программы будет идти речь в курсе:
  - программы, создаваемые одним человеком, а не коллективом
  - программы обработки символьной информации, а также реализующие какие-либо вычислительные алгоритмы или несложные графические программы
- Уровень автоматизации разработки – минимальный:
  - Компилятор GNU Fortran (последняя стабильная версия)
  - Текстовый редактор Vim
  - Редактор диаграмм Dia
- Задание на разработку дает преподаватель, оговаривая требования к выполнению программы
- Разработчик должен сам тщательно тестировать свои программы

Этапы разработки программ

- Постановка задачи, техническое задание, ЕСПД
- Выбор методов решения задачи и инструментальных средств
- Алгоритмизация
- Под алгоритмом будем понимать точное предписание, определяющее процесс преобразования исходных данных в искомый результат

Этапы разработки программ

- Программирование (кодирование) алгоритма
- Трансляция текста программы



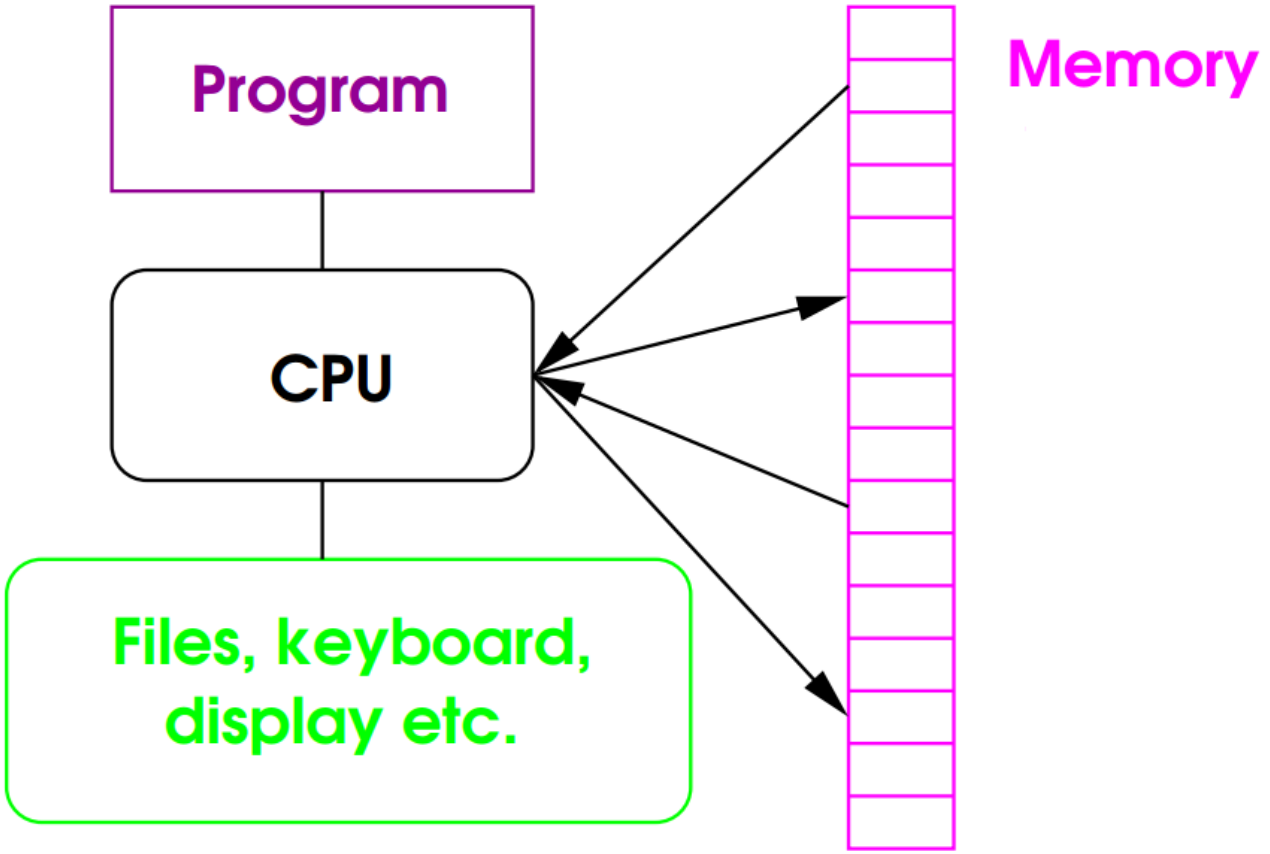
Этапы разработки программ

- Отладка - исполнение программы на компьютере с целью обнаружения, локализации и устранения смысловых ошибок.
  - Тест - совокупность входных и заранее известных выходных данных.
- Изготовление программного изделия (program product) . Это вся документация на программную систему и сама система , представленная на носителе.

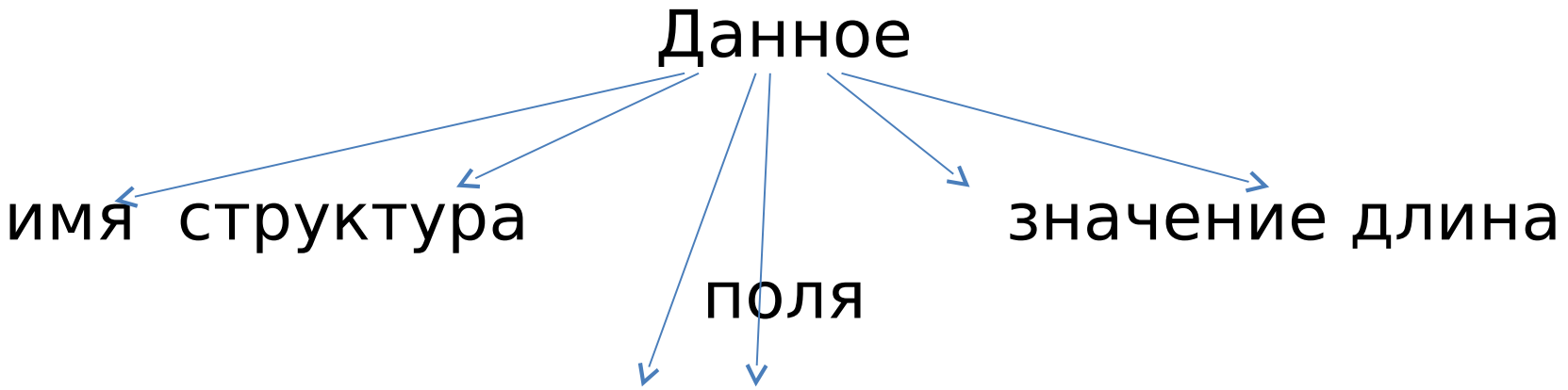
Этапы разработки программ

- Сопровождение программного изделия. Это этап эксплуатации , модернизации, обнаружения и устранения ошибок. При этом необходимо вносить все проведенные изменения в документацию.

Модель программирования на Fortran



Простейшие типы данных



- По строению данные разделяются на:

простые и составные.

- По типу значений: арифметические, логические, символьные.

Простейшие типы данных

- Адрес данного -это адрес первого байта данного в оперативной памяти.
- Длина поля данного - это его размер в байтах.
- Операции –это набор допустимых действий с этим данным.
- Элементарные данные это неделимые данные, а сложные данные - это упорядоченные структуры из неделимых данных.
- По возможности изменяться: константы и переменные.

Особенности обработки данных

- Преобразование из внутренней во внешнюю форму и наоборот . Возникающие ошибки округления вещественных данных.
- При записи значения данного в память его предыдущее значение стирается, а при выборке - сохраняется.

Особенности обработки данных

- Переменная может быть определённой или неопределённой.
- К моменту выполнения действий с данным, его значение должно быть определено (проинициализировано).
- В любой момент времени центральный процессор выполняет только одну операцию, т.е. пока будем придерживаться концепции последовательного программирования.

Описание программного продукта

В соответствии с ЕСПД документация на программный продукт должна содержать:

- Схему данных , которая отображает путь данных и определяет этапы обработки и носители данных.
- Схему работы системы, отображающую управление операциями и поток данных в системе.
- Схему взаимодействия программ - путь активации программ и взаимодействие их с данными.

Описание программного продукта

- Схему ресурсов системы, отображающую все устройства компьютера и процессоры и передачу управления между процессорами и остальными устройствами.
- Схемы алгоритмов – графическое изображение алгоритмов с использованием специальных символов, описывающих определенные действия и соединенных друг с другом.

Пример задачи

Напишите программу переводящую время, заданное в часах, минутах и секундах во время заданное лишь в секундах.

Алгоритм:

1. Умножить часы на 60
2. Добавить минуты к полученному значению
3. Умножить результат на 60
4. Добавить секунды к полученному значению

Логическая структура

1. Запуск программы
2. Выделить память под данные
3. Вывести запрос ввода данных на экран
4. Прочитать время в часах, минутах и секундах
5. Перевести время в секунды
6. Вывести значение в секундах
7. Завершить программу

Текст программы

```
program example1
! Перед комментариями ставиться восклицательный знак
      implicit none
      integer :: hours, mins, secs, temp
      print *, ’Type the hours, minutes and seconds’
      read *, hours, mins, secs
      temp = 60* ( hours*60 + mins ) + secs
      print *, ’Time in seconds =’, temp
end program example1
```

Высокоуровневая структура

1. Начало программы (или процедуры)

program example1

2. Невыполняемые операторы: объявление типов и размерности данных

3–6. Выполняемые операторы

7. Конец программы (или процедуры)

end program example1

Комментарии могут встречаться в любой части текста программы и не влияют на ее исполнение.

Невыполняемые операторы

2. Выделить память под данные

integer :: hours, mins, secs, temp

hours, mins, secs – входные данные,
temp – временная переменная.

Выходные данные - 'Time . . . =' и temp

Выполняемые операторы

3. Вывести на экран запрос ввода данных

   print *, ’Type the hours, ...’

4. Прочитать время в часах, минутах и секундах

   read *, hours, mins, secs

5. Преобразовать время в секунды

   temp = 60*(hours*60 + mins) + secs

6. Вывести число секунд

   print*, ’Time in seconds =’, temp