

Программирование

Лекция 1

Фёдоров Станислав Алексеевич

Состав курса лекций

- Основы алгоритмизации
- Основы структурного программирования
- Основы процедурного программирования
- Основы регулярного программирования
- Основы функционального программирования
- * Основы объектно-ориентированного программирования

Виды занятий

- Лекции, на которых рассматривается теоретический материал, а также схемы алгоритмов и программы с применением различных типов и структур данных
- Лабораторные занятия, на которых рассматриваются и дополнительно тестируются преподавателем выполненные задания

Что мы выполняем в семестре

- Домашние задания
- Две контрольные работы:
 - на работу с массивами
 - на работу с динамическими списками
- 11 индивидуальных упражнений из сборника по основам программирования
- 9 индивидуальных лабораторных работ по алгоритмам и структурам данных

Домашние работы

- 1) Создание инструментальной платформы
- 2) Профессиональная работа в менеджере окон
- 3) Эффективная работа в текстовом процессоре Vim
- 4) Создание инструментальной среды разработки

Список литературы — структуры данных и алгоритмы

1. Структуры данных и алгоритмы : Пер. с англ. / А.В. Ахо, Д.Э. Хопкрофт, Д.Д. Ульман .— Москва : Вильямс, 2003.
2. Алгоритмизация и структурное программирование . Учеб. пособие. СПб.: изд-во СПбГПУ, 2000, 56 с.
3. 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.
4. Мартин Фаулер. UML. Основы : краткое руководство по стандартному языку объектного моделирования.— 3-е изд. — СПб. : Символ-Плюс, 2008.
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 18

1. Арьен Маркус. Современный Fortran на практике. — М. : ДМК Пресс, 2015.
2. M. Metcalf, J. Reid, M. Cohen - Modern Fortran Explained, 2011
3. N. Clerman, W. Spector - Modern Fortran. Style and Usage, 2012
4. 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.
5. *Бартеньев О. В. Современный Фортран / О. В. Бартеньев .— Изд. 4-е, доп. и перераб .— М. : Диалог-МИФИ, 2005 .

Версии стандартов Fortran

FORmula **TRAN**slation 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 (2010)

Fortran 2018 (2018)

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
- 5.

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. Спецификация (невыполняемые операторы):
объявление типов и размерности данных

3–6. Исполняемая часть: **выполняемые операторы**

7. Конец программы (или процедуры)

END PROGRAM example1

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

Спецификация

2. Выделить память под данные

INTEGER :: hours, mins, secs, temp

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

Выходные данные - 'Time ... =' и temp

Исполняемая часть

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
```