

ИАТЭ НИЯУ "МИФИ"

Программирование + Информатика

Кафедра ИКД
(Информационно компьютерных дисциплин)

Юрий Романович Кофтан

koftan@obninsk.org

icq: 380720879

т. 910-912-1220

Информатика

Смысловая ориентировка (зачем делать) –

1. Владеть основами алгоритмизации и методами решения типовых задач
2. Понимать особенности выполнения программ на ЭВМ

Функционально-целевая ориентировка (что делать) –

1. Изучить способы составления алгоритмов и их применения для решения типовых задач
2. Изучить машинное представление данных и распределение памяти выполняемых программ

Исполнительская ориентировка (как делать) –

1. Освоить составление графических схем алгоритмов с описанием действий/операций в псевдокоде
2. Изучение системы счисления, размещение данных в ПЭВМ с Intel-совместимыми процессорами и распределение памяти выполняемых программ на Pascal в среде MS DOS

Языки программирования высокого уровня (ЯП) Pascal и C

Смысловая ориентировка (зачем делать) – овладеть инструментом, позволяющим решать задачи с использованием ЭВМ

Функционально-целевая ориентировка (что делать) –

1. Программированию невозможно научить, ему можно только научиться !
2. Необходимо изучить синтаксис и семантику изучаемых языков программирования и правила использования их для реализации алгоритмов

Исполнительская ориентировка (как делать) –

1. Будем изучать ЯП Pascal и C в оболочках Borland Pascal v.7.0 и Borland C v.3.1
2. Изучение ЯП осуществляется на примере реализации алгоритмов, решающих наиболее типовые задачи программирования

Язык программирования PASCAL

Borland Pascal v.7.0

Язык программирования C / C++

Borland C++ v.3.1

Создание индивидуальных настроек на компьютерах студентов

На Вашем индивидуальном рабочем диске Z: создать папки (диск размещён на сервере):

- BP – настройки и исходные тексты Pascal,
- BC – настройки и исходные тексты C / C++,
- <Фамилия_студента _рус > – файлы для проверки преподавателем,
- Koftan – проверенные преподавателем задания.

В папку BP поместить файлы:

- BP.TP
- BP.DSK
- BP.PIF – не обязательно
- BP.BAT, -

Первые 3 файла переместить вручную из папки BP\BIN с диска C: или с диска Y: (с сервера).

Файл BC.BAT – скопировать с Y:\Temp\!Koftan\Compiler\Образцы файлов bp.bat и bc.bat.

В папку BC поместить файлы:

- TCCONFIG.TC
- TCDEF.DSK
- TCDEF.DPR
- BC.PIF – не обязательно
- BC.BAT, -

Первые 4 файла надо переместить вручную из папки BC\BIN с диска C: или с диска Y: (с сервера). Файл BC.BAT – скопировать с Y:\Temp\!Koftan\Compiler\Образцы файлов bp.bat и bc.bat.

В файлах BP.BAT (BC.BAT) надо вызвать программу русификации оболочки языка программирования: z:\RUSFNT.com (взять на диске Файл BC.BAT – скопировать с Y:\Temp\!Koftan\Compiler\Образцы файлов bp.bat и bc.bat), а затем исполняемый файл компилятора (bp.exe или bc.exe): c:\bp\bin\bp.exe или c:\bc\bin\bc.exe.

Про желаниии, для файлов BP.PIF (BC.PIF) отредактировать Свойства: в закладке Программа указать в Команде путь к файлам *.BAT (z:\BP\BP.bat, z:\BC\BC.BAT), а в закладке Экран – вид Режима: Полноэкранный или Оконный.

Язык программирования PASCAL

Borland Pascal v.7.0

Язык программирования C / C++

Borland C++ v.3.1

Создание индивидуальных настроек на компьютерах студентов

(продолжение)

Для входа в среду программирования (оболочку) надо запустить (двойным щелчком левой клавиши мыши) выполняемые файлы **bp.bat** и **bc.bat**, соответственно для входа в оболочку **Borland Pascal v.7.0** и **Borland C++ v.3.1**.

Теперь надо осуществить настройку оболочек и сохранить эти настройки на диске **Z:** в своих папках **BP** и **BC**.

Требуется настроить:

1. Папку (Directory) по умолчанию: **File -> Change dir...**;
2. Для оболочки Pascal установить размер основного окна 43/50 строк (в C/C++ этот размер установлен по умолчанию сразу): **Options->Environment->Preferences...->Screen sizes**;
3. Создать окно исходного кода программ: **File -> New**;
4. Создать окно для ввода входных значений в программу и просмотра выходных сообщений Вашей программы (Output) -
в Pascal: **Debug->Output**, а в C/C++: **Windows->Output**;
5. Прописать в оболочках пути к библиотечным модулям Pascal и C, которые установлены на Вашем компьютере (в ауд. 2-613 и 2-614 они лежат в папках BP и BC на диске C:\): в Pascal: **Options->Directories...** (настроить путь к папке BP\UNITS в строках **Unit directories** и **Resource directories**;
Directories... (настроить путь к папке **BC\INCLUDE** в строке **Include directories** и к папке **BC\LIB** в строке **Library directories**;
6. **Сохранить настройки оболочек:**
в Pascal: **Options->Save as...** выбрать путь к папке **Z:\BP** и файлу **BP.TP**,
в C/C++: **Options->Save...** выбрать путь к папке **Z:\BC** и все пункты из списка.

Алгоритм и программа

Простейший алгоритм - это последовательность действий, которые необходимо выполнить, чтобы решить поставленную задачу за конечное число шагов.

Программа - это набор команд на языке, понятном исполнителю, реализующий некоторый алгоритм. Программы на языках низкого уровня понятных компьютеру - машинных языках - очень детальные и сложные. Программирование на них очень трудоемкое. **Поэтому разработаны ЯПВУ – языки программирования высокого уровня.**

Транслятор – это программа, которая переводит входную программу на исходном (входном) языке в эквивалентную ей выходную программу на результирующем (выходном) языке.

Компилятор – это транслятор, который осуществляет перевод исходной программы в эквивалентную ей объектную программу на языке машинных команд или на языке ассемблера. Таким образом, **компилятор отличается от транслятора** лишь тем, что его результирующая программа всегда должна быть написана на языке машинных кодов или на языке ассемблера. Результирующая программа транслятора, в общем случае, может быть написана на любом языке – возможен, например, транслятор программ с языка Pascal на язык C. Соответственно, всякий компилятор является транслятором, но не наоборот – не всякий транслятор будет компилятором.

Интерпретатор – это программа, которая воспринимает операторы входной программы на исходном языке и тут же выполняет их. В отличие от трансляторов интерпретаторы не порождают результирующую программу (и вообще какого-либо результирующего кода) – и в этом принципиальная разница между ними. Интерпретатор, так же как и транслятор, анализирует текст исходной программы. Однако он не порождает результирующей программы, а сразу же выполняет исходную в соответствии с ее смыслом, заданным семантикой входного языка.

Компиляция, отладка и тестирование

В ЯП, как и в любом другом языке, существуют синтаксис - правила записи его конструкций - и семантика - смысл его конструкций. **Компилятор проверяет только синтаксис.** Поиск же семантических ошибок занимается программист в процессе тестирования и отладки своей программы

Отладка - это поиск и исправление ошибок в программе. **Тестирование** - это составление специальных наборов входных и выходных данных (тестов), а затем исполнение программы и проверка полученных результатов в поисках возможных семантических или логических ошибок.

Простые программы

Язык программирования PASCAL

```
Program Simple;  
const  
  Text = 'Простая программа';  
begin  
  WriteLn(Text);  
end.
```

- Строки размещаются произвольно
- Пробел – разделитель конструкций языка
- Заглавные / строчные буквы игнорируются (кроме текстовых констант)

Язык программирования C

```
#include <stdio.h>  
main( )  
{  
  printf("Простая программа\n");  
}
```

- Строки размещаются произвольно
- Пробелы игнорируются, если они не разделители лексем
- Заглавные и строчные буквы различаются

Простые программы

```

Program Simple;      - объявление имени программы
const                - блок описания констант
  Text = 'Простая программа'; - имя и значение константы
begin                - начало раздела операторов
  WriteLn(Text);      - оператор вывода на экран
end.                 - завершение программы

```

```

#include <stdio.h>      - включение другого файла
main( )                - имя функции (главной)
{                      - начало тела функции
  printf("Простая программа\n"); - оператор вывода на печать
}                      - конец программы

```

stdio.h - standard input/output header - заголовок библиотеки стандартного ввода-вывода C, содержащей функцию printf

Структура простой программы

Pascal

[**Заголовок программы:**

Program XXX;]

[**Декларация** - раздел описаний]

[**Тело** - раздел операторов:

begin

Оператор1;

Оператор2;

...

end.]

В разделе описаний обязательно описываются все нестандартные идентификаторы меток (label), констант (const), типов (type), переменных (var), процедур (procedure), функций (function).

C

Функция n

[**Заголовок функции:**

- Директивы - инструкции для предпроцессора C - **#include**

- **Имя функции - ()]**

[**Тело функции - {оператор1; оператор2;}]**

Функция n+1

Программа на C – это одна или более функций, причем главная называется **main ()**. Описание функции состоит из заголовка и тела. Тело функции заключено в фигурные скобки и представляет собой набор операторов, каждый из которых оканчивается символом "точка с запятой".

Литература

- **Е.В. Крылов, В.А. Острейковский, Н.Г. Типикин. Техника разработки программ. В 2 кн. Учебник. – М. Высш. Шк., 2007. (библиотека).**
- **Т. А. Павловская. С/С++. Программирование на языке высокого уровня. – СПб.: Питер, 2005. – 461с.: ил (библиотека, файл).**
- **Н. Вирт. Алгоритмы и структуры данных. Пер. с англ. – СПб.: Невский Диалект, 2001. – 352с.:ил. (библиотека).**
- **Т. А. Павловская. Pascal. Программирование на языке высокого уровня. – СПб.: Питер, 2006. – 393с.: ил. (файл).**
- **Соболь Б.В. Информатика: Учебник. (файл, электронный учебник).**

Ссылки на некоторые ресурсы по ЯП в Интернет

Общие ссылки

<http://www.emanual.ru/cat/5/>

<http://www.citforum.ru/programming/>

Pascal

<http://it.kgsu.ru/Pascal/oglav.html> – *Начала*

С

<http://it.kgsu.ru/C++/oglav.html> – *Начала*