



Лабораторная работа № 1

Программы с простейшей структурой

Терентьев Александр Николаевич

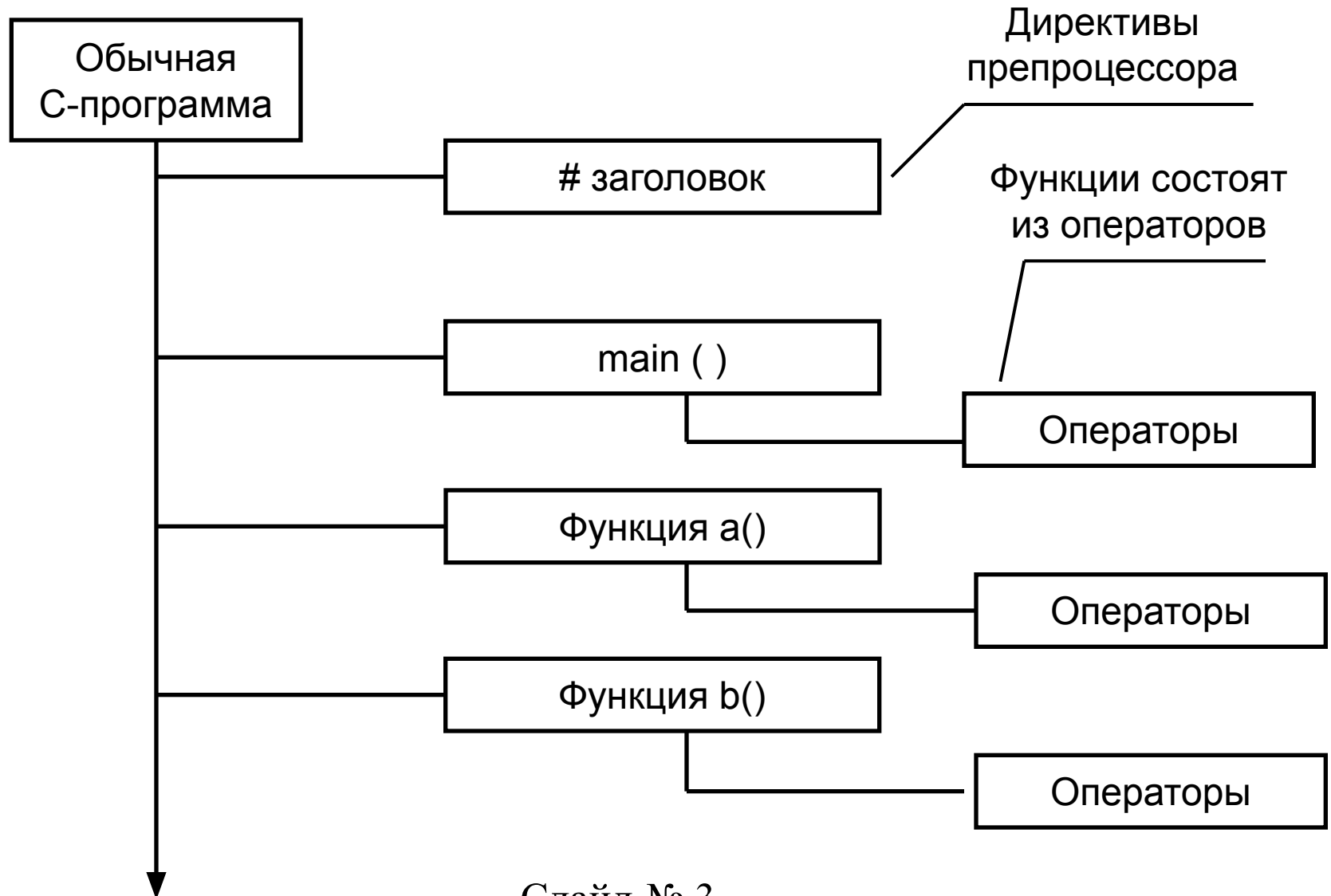
Национальный Технический Университет Украины
“КПИ”, Институт Прикладного Системного Анализа

Киев 2007

Цель работы:

знакомство со средой программирования
Turbo C / C++ и основными принципами
программирования.

Структура программы написанная на С



На языке С программа представляет собой совокупность одной или нескольких функций, команд препроцессора и внешних описаний.

Одна из функций имеет имя **main** и является главной.

Функция main выполняется всегда первой.

Функции описывают совокупность действий, которые нужно выполнить.

Команды препроцессора указывают на преобразования, которые нужно выполнить над программой.

Например строка кода в программе

```
#define PI 3.1415
```

задает именованную константу PI.

Именованная константа — это просто имя, которому вы присваиваете постоянное значение (константу). Такая константа в отличие от значения переменной не может изменяться по мере выполнения программы.

Также команды препроцессора подключают информацию про стандартные библиотеки.

Например:

```
#include "math.h"
```

Имя подключаемого файла в директиве препроцессора `#include` может быть указано: в двойных кавычках - в этом случае препроцессор сначала выполняет поиск данного файла в том же каталоге, в котором расположен файл, содержащий директиву препроцессора `#include`, а затем в каталогах, определяемых опцией компиляции и переменной среды `INCLUDE`.

Например:

```
#include <iostream>
```

в угловых скобках - при этом исключается поиск имени файла в том же каталоге, в котором расположен файл, содержащий директиву препроцессора #include.

Объявление переменных размещается в начале файла программы. Все переменные должны быть описаны до их использования.

Вывод и ввод информации можно выполнять при помощи универсальных функций ввода/вывода `printf()` и `scanf()`

Функция

printf(stream, format)

записывает аргументы в поток *stream* в соответствии со спецификацией *format*.

scanf(format, pointer, ...)

вводит данные из стандартного потока ввода в соответствии с форматом *format*, записывая их в переменную *pointer*.

Синтаксис условного оператора

If (*выражение*)

 оператор 1

Else

 оператор 2

Если *выражение* принимает истинное значение, то выполняется *оператор 1* в противном случае *оператор 2*.

Нет никаких ограничений на вложенность блоков if-else.

Алфавит языка

- Буквы:

A, B, ..., Y, Z, a, b, ..., y, z и символ _

- Цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

- Символы:

+, -, *, /, >, =, <, !, &, |, .., \, ,, ', ", ?, :, ~, {, },
%, ^, (,), [,], запятая

Зарезервированные слова

asm	default	float	new	signed	unsigned
auto	delete	for	operator	sizeof	virtual
break	do	friend	private	static	void
case	double	goto	protected	struct	volatile
char	else	if	public	switch	while
class	enum	inline	register	this	
const	explicit	int	return	typedef	
continue	extern	long	short	union	

Идентификатор – последовательность
буквенно-цифровых символов,
начинающаяся с буквы, длиной до 1024
СИМВОЛОВ.

- Допустимые:

- count

- Test23

- Ptr_1

Недопустимые:

1count

sizeof

Ptr...1

Базовые типы данных

- Целые длиной один байт – char
- '0'
- Целые длиной несколько байт – int
- 48 060 0x30
- С плавающей точкой – float
- 48.f .48e2F
- Двойной точности - double
- 48.+480.0e-1
- Без значения - void

Модификация базовых типов

- Со знаком – signed
- Короткое – short
- Без знака – unsigned
- Длинное – long

Арифметические операции

- - Вычитание
- + Сложение
- * Умножение
- / Деление
- % Остаток от деления
- -- Декремент
- ++ Инкремент

Пример

```
float y,x = 5/3; //=1.0
```

```
y = 5./3; // =1.666...
```

```
int k=1;
```

```
++k;    // =2
```

```
int m=--k;    // k--, m=k;
```

```
int n=k++;    // n=k, k++;
```

```
// m=1, n=1
```


Операции сравнения и логические операции

- > Больше **&&** И
- >= Больше или равно **||** ИЛИ
- < Меньше **!** НЕ

- <= Меньше или равно
- == Равно
- != Не равно

Пример

```
int a,b,c,x=5,y=0;  
a=x&&y; // = false  
b=x||y; // = true  
c=!x;   // = false
```

Множественное присваивание

$$x = y = z = 0;$$

Задание на урок

Напишите программу для вычисления значений переменной ***y*** при произвольных значениях ***a*** и ***b*** по формуле:

$$y = \frac{a - 2 \cdot b}{3 \cdot a + b} - \frac{\sin(2)}{\cos(2b)}$$

```
#include<math.h>
#include<conio.h>
#include<stdio.h>
void main ()
```

Код программы

```
{
    float a, b, y;
    clrscr();
    printf("Введите значение a = ");
    scanf("%f", &a);
    printf("Введите значение b = ");
    scanf("%f", &b);
    if ((b==(-3*a)) | (cos(2*b)==0))
        printf("Неверные данные");
    else
    {
        y=(a-2*b)/(3*a+b) - sin(a)/cos(2*b);
        printf(" Y=%f ",y);
    }
    getch();
}
```

Контрольные вопросы при сдаче лабораторной работы № 1

1. Назовите базовые типы данных?
2. Какие типы констант можно использовать в языке C?
3. Типы операций? Приоритет выполнения операций?
4. Esc-последовательности? Их назначение и использование?
5. Приведите пример условной конструкции if-else?

Константы

Константой называется фиксированная величина, которая не изменяется в процессе выполнения алгоритма или программы. Примером является всем известное знаменитое число "ПИ", значение которого в компьютере принимается приближенно равным 3,1415926.

В языке Си определено пять типов констант:

- символьные константы,
- целые константы,
- вещественные константы,
- константы перечисляемого типа
- и нулевой указатель (null).

Символьные константы служат для изображения отдельных символов из таблицы ASCII символов. Каждая такая константа состоит из изображения самого символа и ограничивающих его с двух сторон апострофов, например, 'A' или 'b'. Внутри апострофов записываются цифры, буквы, знаки препинания и управляющие символы, называемые управляющими последовательностями, которые не имеют графического изображения на экране дисплея или на принтере. Для кодирования таких символьных констант используются комбинации из нескольких символов, имеющих графическое представление. Каждая управляющая последовательность начинается с символа обратной косой черты ("backslash" - обратный слэш).

Ниже в таблице представлены эти управляющие символы и дана расшифровка их назначения.

Esc-последовательность

- В наборе символов, используемых в компьютере, имеются символы, которые компьютером воспринимаются как команды или интерпретируются особым образом. Escape-символ или последовательность - это знак, который подается компьютеру, что последующий символ или определенное число символов должны интерпретироваться не так, как обычно.
- Например, символ **n** сам по себе интерпретируется, как соответствующая буква алфавита, но в сочетании с "\" во многих языках программирования обозначает новую строку. Так вот, "\" и является escape-символом, заставляя компьютер интерпретировать букву **n** не так, как обычно.
- в Windows обычно нажимают ALT + комбинацию цифр, например N пишется с помощью ALT + 78

Специальные символьные константы

- '\a' - сигнал-звонок
- '\b' - возврат на одну позицию (на один символ)
- '\f' - новая страница, перевод страницы
- '\n' - новая строка, перевод строки
- '\r' - возврат каретки (курсора) к началу строки
- '\t' - горизонтальная табуляция

- '\v' - вертикальная табуляция (v)
- '\\' - обратный слэш (\)
- '\"' - кавычка (")
- '\'' - апостроф (')
- '\0' - нулевой символ, нулевой байт (0)
- '\ddd' - восьмеричная константа (ddd), здесь d - восьмеричная цифра
- '\0xhh' - шестнадцатеричная константа (hh), здесь h - шестнадцатеричная цифра
- '\?' - знак вопроса (?)