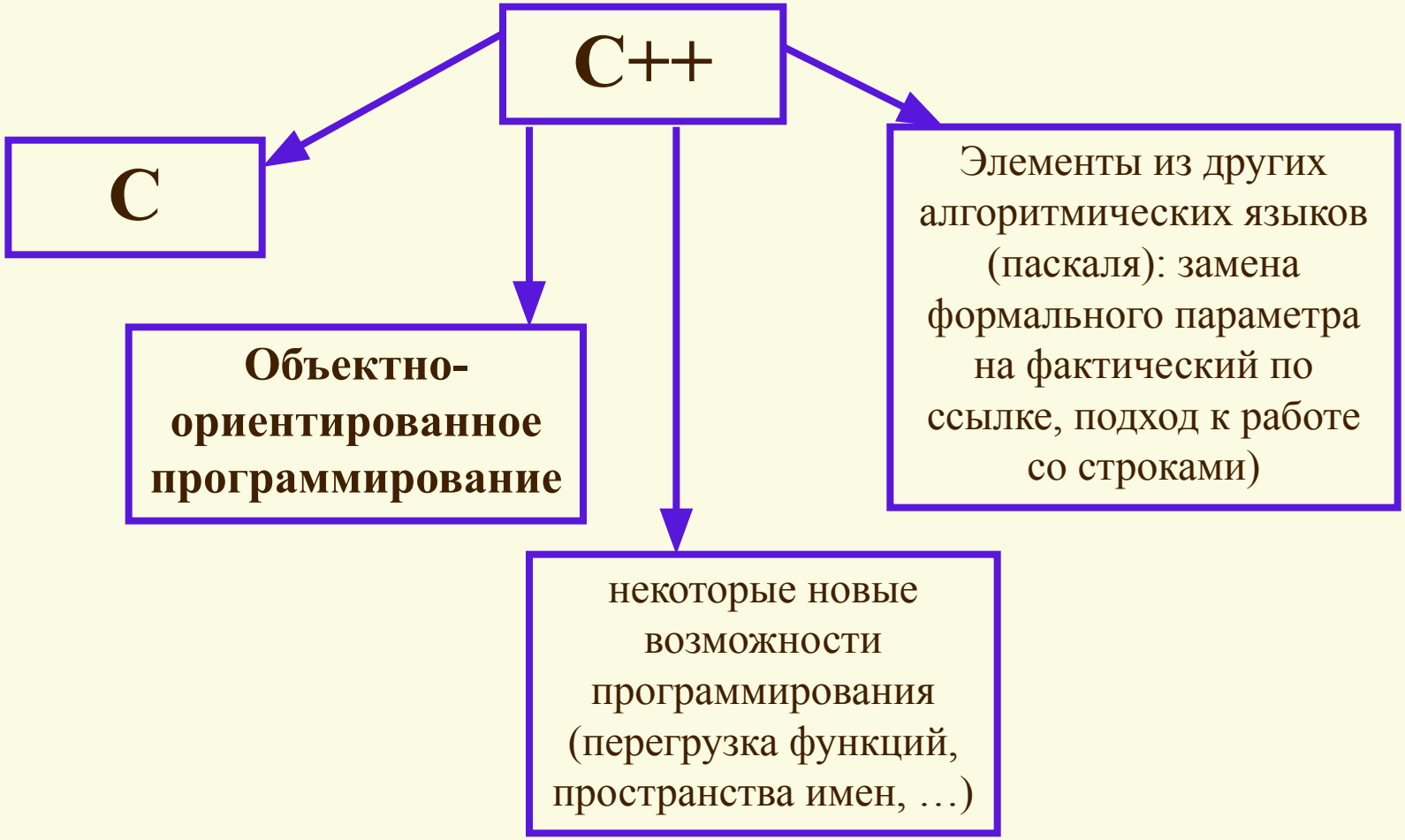
Элементы языка СИ

Средства для написания простейших программ

C и C++


C++
C
Объектно-ориентированное программирование
некоторые новые возможности программирования (перегрузка функций, пространства имен, …)
Элементы из других алгоритмических языков (паскаля): замена формального параметра на фактический по ссылке, подход к работе со строками)

Краткая история языка C и его диалекты

- 1969-1973 годы- Деннис Ритчи создал язык C
- 1978 г. - Брайан Керниган1978 г. - Брайан Керниган и Деннис Ритчи опубликовали первую редакцию книги «Язык программирования Си» (K&R)
- 1989 – ANSI C (или C89)
- 1990 – ISO ANSI C (C90)
- C99 и C11

Язык C++

- 1983 г. – Бьёрн (Бьерне) Страуструп создал C++
- 1985 г. - вышло первое издание «Языка программирования C++»
- Последний стандарт ISO C++ - C++11 (2011 г.)

Литература

1. Б. Керниган, Д. Ритчи. Язык программирования Си (The C programming language). – М.: Вильямс. 2007. – 304 с.
2. Б. Страуструп. Язык программирования C++ (The C++ programming language). – М.: Бином-Пресс. 2007. – 1104 с.

ВАЖНО НЕ НА КАКОМ ЯЗЫКЕ ПРОГРАММИРОВАТЬ, А КАКИЕ ЗАДАЧИ РЕШАТЬ!

Сведения, необходимые для написания простейшей программы:

- Структура простейшей программы.
- .Типы и структуры данных, имеющиеся в алгоритмическом языке, их описание, допустимые операции.
- .Операторы преобразования данных: присваивания (преобразования внутренних данных), операторы ввода и вывода.
- Правила записи алгоритма - программирование основных алгоритмических структур.

Структура простейшей программы на любом алгоритмическом языке

Заголовок
ограничитель

описания - неисполняемые инструкции языка

операторы - исполняемые инструкции языка

ограничитель

Структура простейшей СИ-программы

Директивы препроцессора
(
в простейшем случае
#include <stdio.h>/*
ввод/вывод*/
#include <math.h>/*стандартные
математические функции*/)
void main()
{ описания

операторы
}

```
#include <stdio.h>
void main()
{int a,b,c;/*описание трех целых
переменных*/
 printf("введите a и b\n");
/*приглашение к вводу a и b*/
scanf("%d%d", &a, &b);/*ввод
a,b*/
c=a+b;/*вычисление c - суммы */
printf("c=%d\n", c);/*вывод c*/
}
```

ТИПЫ ДАННЫХ В СИ

БАЗОВЫЕ ТИПЫ:

int - *целый*

float - *вещественный одинарной точности*

double - *вещественный двойной точности*

char - *символьный*

КВАЛИФИКАТОРЫ:

short - *короткий*

long - *длинный*

signed - *со знаком*

unsigned - *без знака*

ТИПЫ ДАННЫХ В СИ

Стандартные целые типы данных Си (MS DOS)

Тип данных	Размер, байты	Диапазон значений
unsigned char	1	0...255
char, signed char	1	-128...127
unsigned int, unsigned, unsigned short int	2	0...65535
int, signed int, short int, short	2	-32768...32767
unsigned long	4	0...4294967295
long, long int	4	-2147483648...2147483647

ТИПЫ ДАННЫХ В СИ

Стандартные целые типы данных Си (Win 32)

Тип данных	Размер, байты	Диапазон значений
unsigned char	1	0…255
char, signed char	1	-128…127
short int, short	2	-32768…32767
unsigned short int	2	0…65535
unsigned int, unsigned, unsigned long	4	0… 4294967295
int, signed int, long, long int	4	-2147483648…2147483647

ТИПЫ ДАННЫХ В СИ

Стандартные вещественные типы данных Си (MS DOS, Win32)

Тип данных	Размер, байты	Диапазон порядка	Число цифр мантиссы
float	4	-38...+38	7
double	8	-308...+308	15
long double	10	-4932...+4932	19

Описания в Си

тип список_имен_переменных;

тип имя1, имя2,…,имяi,…, имяN;

*имя=значение/*инициализация*/*

Имя - идентификатор.

Идентификатор - последовательность букв, цифр и знаков подчеркивания, начинающаяся с буквы или знака подчеркивания.

Пример описания:

float a, b=1.5, _b=0.5, b1; int n=10,i=0, j, ik=1;

КОНСТАНТЫ В СИ

Обозначенные (именованные)
const *тип имя_конст=значение конст*;

защита от записи

Явные

Целые

Десятичные
9076 -561
+1111

Восьмеричные
03457 -0651 +023

Шестнадцатеричные
0x2AA -0x111 0XF4
+0X4D -0XF

Вещественные
0.0013 2.7E-9

Символьные
‘a’ ‘+’ ‘\n’ ‘\t’ ‘\0’
‘\040’ \0x20’

Текстовые
“строка” “a”

ВЫРАЖЕНИЯ В СИ

Выражения *- это* ***операнды****, соединенные знаками* ***операций****.*

Операнды*: переменные,*
константы,
результаты обращения к функциям;
выражения, заключенные в круглые скобки.

ОПЕРАЦИИ СИ

Некоторые операции Си.

Ранг	Обозначение операции	Название операции	Ассоциативность
1	( ) [ ]	круглые и квадратные скобки	→
2	! + - ++ -- & * (тип) sizeof	логическое отрицание унарный плюс и минус инкремент (увеличение на 1) декремент (уменьшение на 1) взятие адреса взятие содержимого (см. п.1.5.3) приведение к типу определение размера в байтах	←
3	* / % %	арифметическое умножение и деление получение остатка от деления нацело	→
4	+ -	арифметические сложение и вычитание	→
6	< <= > >=	отношения (меньше, меньше или равно и т. д.)	→
7	== !=	отношения (равно, неравно)	→
11	&&	конъюнкция (логическое "и")	→
12	||	дизъюнкция (логическое "или")	→
14	= операция=	присваивание составное присваивание	←
15	,	операция "запятая"	→

Некоторые операции Си

Две формы инкремента:

++имя_переменной - префиксная

(увеличение операнда до использования)

имя_переменной++ - постфиксная

(увеличение операнда после использования)

Пример. int i=1,c;

1-й фрагмент	2-й фрагмент
c=2*i++;	c=2*++i;
/*в результате i равно 2, c равно 2*/	/* i равно 2, c равно 4*/

Некоторые операции Си

sizeof вычисляет размер в байтах для типа операнда. Две формы: **sizeof (*выражение)*** и **sizeof (*тип*)**. Использование: для построения алгоритмов, обрабатывающих выражения различных типов.

Операция **(*тип*) *выражение*** - приведение выражения к типу, указанному в скобках.
Пример:
(float)i/(float)j

Тип результата выражения

- **Смешивание в выражении операндов разного типа допустимо, но правила автоматического приведения типа сложны - лучше использовать *операцию (тип)*.**
- **Если операнды имеют одинаковый тип, то результат имеет тот же тип: 5/2→2**

Некоторые операции Си

Присваивание:

имя переменной=выражение;

- не только оператор, но и операция⇒ допустима цепочка: a=b=c=d=0

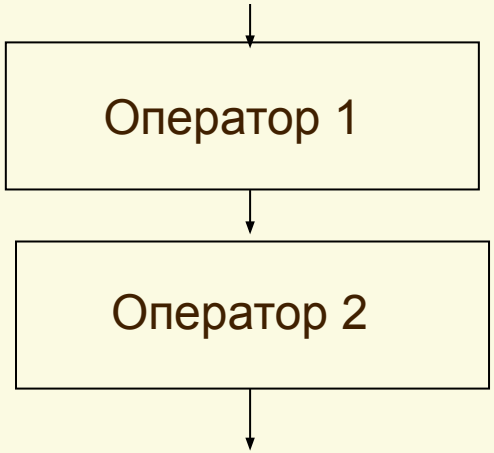
Составное присваивание: операция=

Пример: S+=a; ⇔ S=S+a;

P*=a; ⇔ P=P*a;

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ

Следование

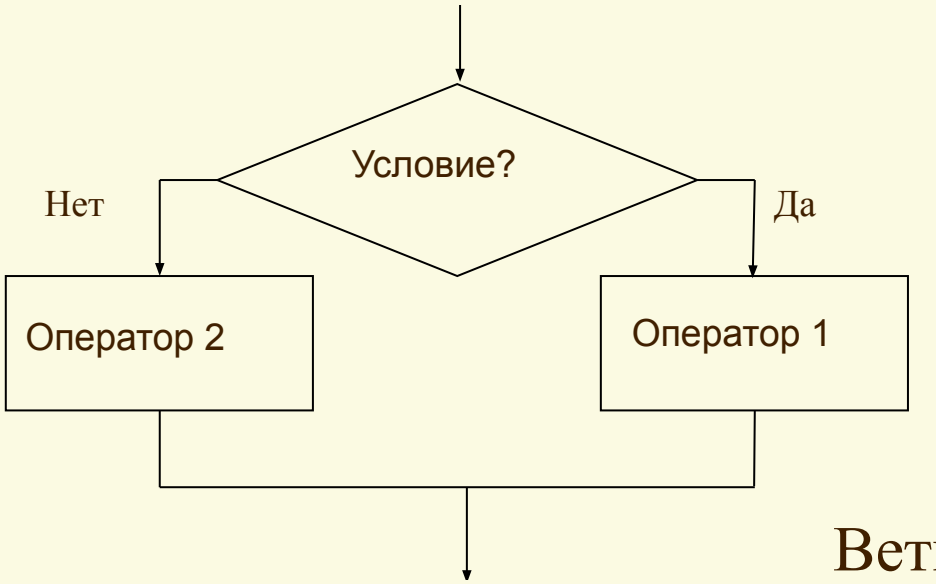


Кодирование на Си:

```
Оператор 1;
Оператор 2;
```

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ

Разветвление (развилка)



Кодирование на Си:

if (*условие)*
***оператор 1*;**
else
оператор 2;

Ветвь «Нет» пустая⇒**else** и ***оператор 2*** отсутствуют.

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ

Разветвление (развилка)

Если развилка является структурной, то:

- Оператор1 и оператор2 не имеют связей.
- Существует четко определенная точка соединения ветвей.

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ

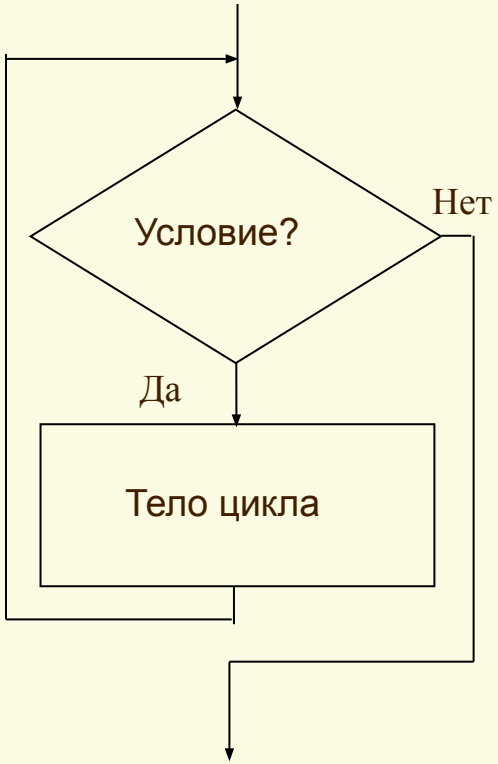
Разветвление (развилка)

```
if (условие)
 {оператор 1_1;
  оператор 1_2;
        …
  оператор 1_N;
 }
else
{оператор 2_1;
  оператор 2_2;
        …
  оператор 2_M;
 }
```

фигурные скобки позволяют объединить несколько операторов в один составной

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ

Цикл ПОКА (с предусловием)



Кодирование на Си:

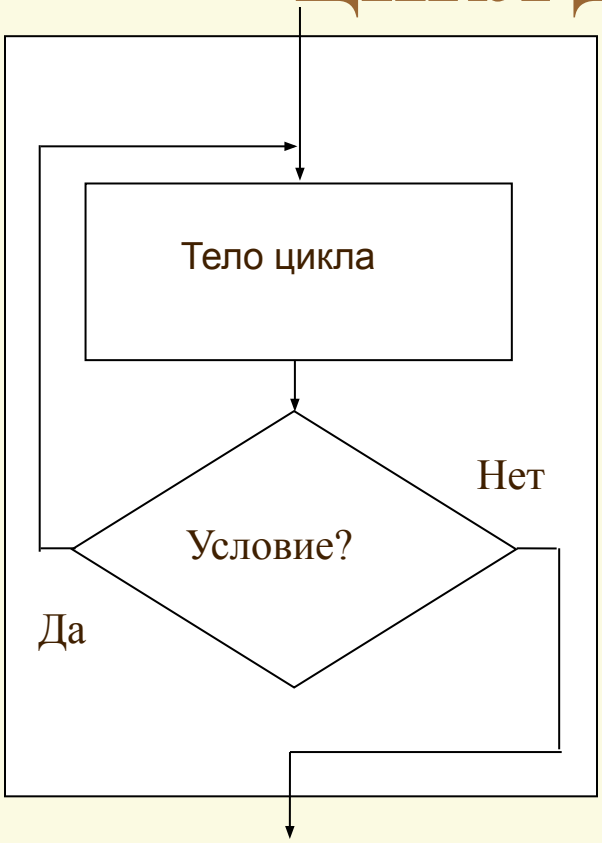
while (*условие*)
тело цикла;

Тело цикла - один оператор, простой или составной.

Тело цикла может не выполниться ни разу.

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ

ЦИКЛ ДО (с постусловием)



Кодирование на Си:

Do

***тело цикла*;**

while (*условие*);

Тело цикла выполняется хотя бы один раз.

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ

Если цикл является структурным, то:

- **Цикл имеет один блок анализа на выход из (продолжение) цикла.**
- **Блок анализа на выход из (продолжение) цикла стоит либо в начале (цикл ПОКА), либо в конце (цикл ДО) цикла.**
- **Ветвь «обратной связи» не содержит операторов.**

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ

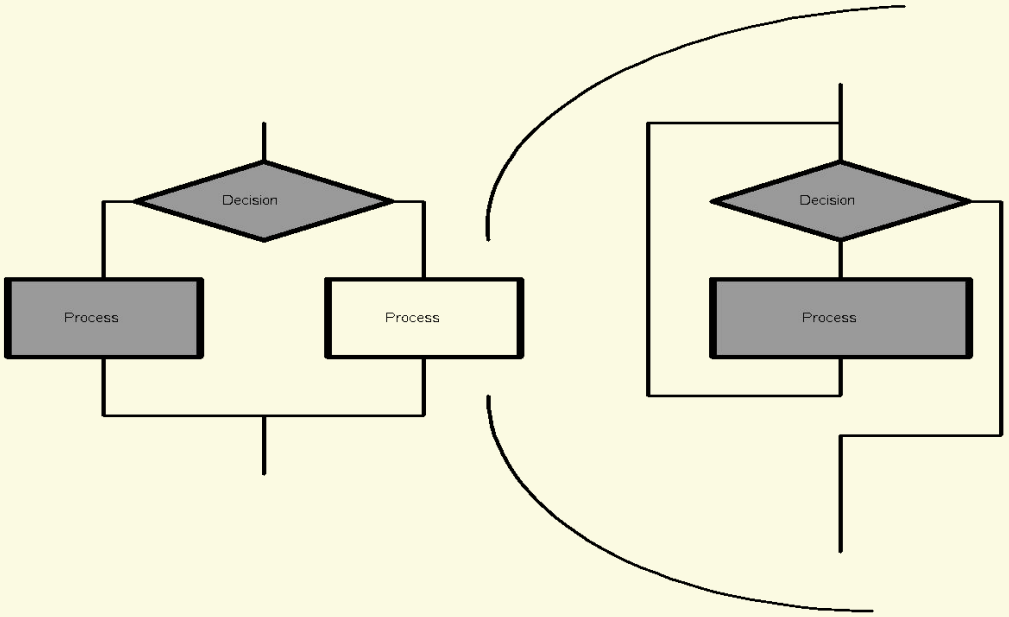
Принцип Дейкстры.

Для построения любого алгоритма достаточно иметь три базовых структуры: следование, ветвление, цикл (безразлично -ПОКА или ДО).

Метод нисходящего проектирования

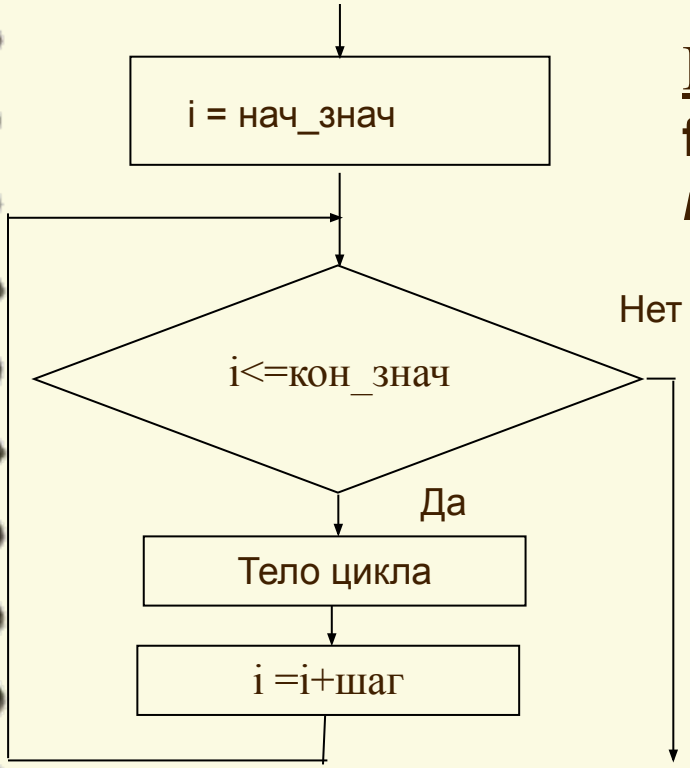
Разбиение алгоритма на части и установлении между ними связей. При установлении связей очень важно, чтобы каждая часть имела один вход и один выход, так что **нисходящее проектирование успешно сочетается с использованием базовых структур алгоритмов**. Каждая часть в свою очередь разбивается на части, и процесс повторяется. Можно сказать, что нисходящее проектирование алгоритма состоит в иерархической последовательной разработке алгоритма от сложного к простому.

Метод нисходящего проектирования



БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ

ПАРАМЕТРИЧЕСКИЙ ЦИКЛ



Кодирование на Си:

for(i=*нач_знач*;i<=*кон_знач*; i=i+*шаг*)
тело цикла;