Системное програмирование

Алгоритм

Алгоритм - это четко опредененный план действий для исполнений

Конечность

Понятность

Массовость

Дискретность

Результативность

Формальность

Завершаемость

Программа - это

- алгоритм, записанный на каком либо языке программирования

- набор команд для компа

Команда - описание действий которые должен выполнить комп

- откуда взять исходные данные

С клавиатуры, из интеренета

- что нужно с ними делать

Обработка, редактировать

- куда поместить результат

Вывести на экран, сохранить

Языки программирования

-машинно ориентированные - каждая прог соответствует одной команде процессора(ассемблера)

- языки высокого уровня - приближены к естейственному (англ) языку, легче воспринимается человеком, не зависят от конкретного компа

---для обучения

---проффесиональные С с++ java

# - директива для проц

.h -

Std - cандарт

Input - ввод

Output - вывод

# include “<stdio.h>” ввод вывод библиотека стандарт

Main() <----- аргументы функции основная функция

{

Return 0

Тело функции

}

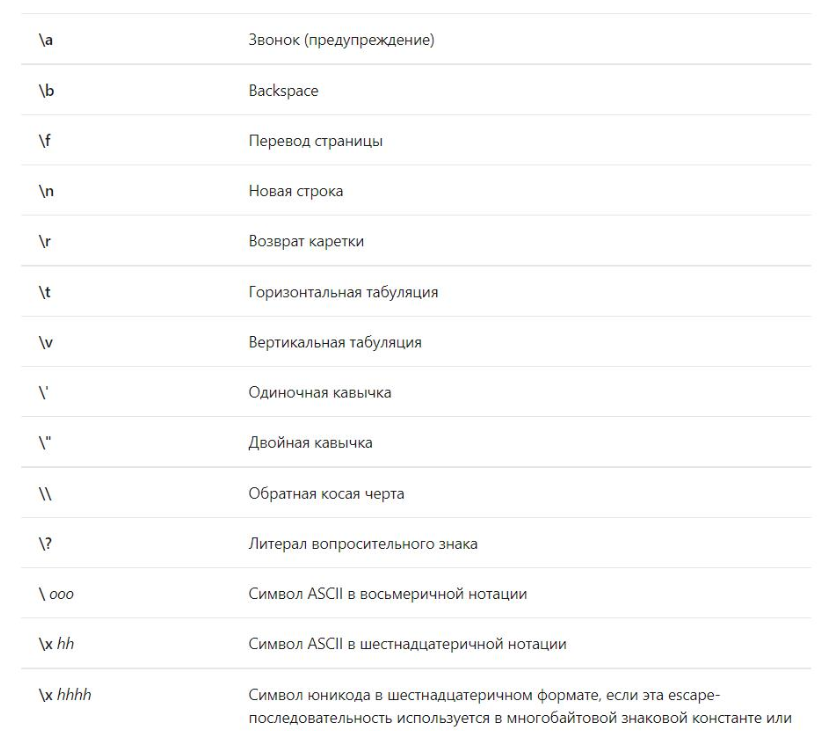
Printf(“привет”);

; пустой оператор

Getch(); ждать нажатия клавиши

\n

Esc последовательности



Переменная - ячейка памяти в озу меет имя и хранит значение какое-то

Int 4 байт 32 бит

Float 4 байт = 32 бит

Char - 1 байт = 8 бит

Bool

Имя переменной не может начинаться с цифры!!!!!

Обьявить переменную = определить ее тип и имя, значение не обязательно int a

Инициализировать переменную = в одну операцию задается и тип и имя а = 16

B = 17 присвоение

‘ ’ литералы для одной

“” ковычки

Если значение не задано в переменной хранится мусор

Имя переменной = выражение

В выражениях

Константы ()

Имена переменных

Знак препинания

Вызов функций

И () для изменения порядка действий

Int a = 7

Float x

X = a / 4 =1

X = 4/ a = 0

X = float(a)/4 = 1.75

X = 1.\*a/4 1. куча нулей

Постфиксная а++

Префикс ++ф

Инкремент

Декремент

++ф срабатывает быстрее чем ф++

% это остсаток от деления

/ целая часть

%I integer

%f float

%d decimal

%c char

Ввод чисел

#Include <stdio.h>

#includen <conio.h> вывод ввод консольный

Int Main()

Int a, b;

Scanf (“%d%d”, &a, &b ); ввод с клавиатуры

& амперсант обращение в ячейку памяти ячейка памяти

Printf (“%d”, c);

Getch();

Минимальное число позиций равно 6

Ветвление алгоритмов

Разветвляющиеся алгоритмы - алгоритмы в котррых последовательность шагов зависит от выполнения некоторых условий.

If (a > b ) {

Max = a;

}

Else {

}

#include <stdio.h>

#include <conio.h>

#include <math.h>

int main()

{

float a,b,c,x,y,z,s;

printf("enter a b c x\n");

scanf("%f%f%f%f", &a, &b, &c, &x);

y = tan(a\*b\*c)-pow(10, -2)\*log(x);

z = (a\*b\*log(c)+0.4\*x)/tan(a\*b);

s = y + z;

printf("%f\n%f\n%f", y, z, s);

getch();

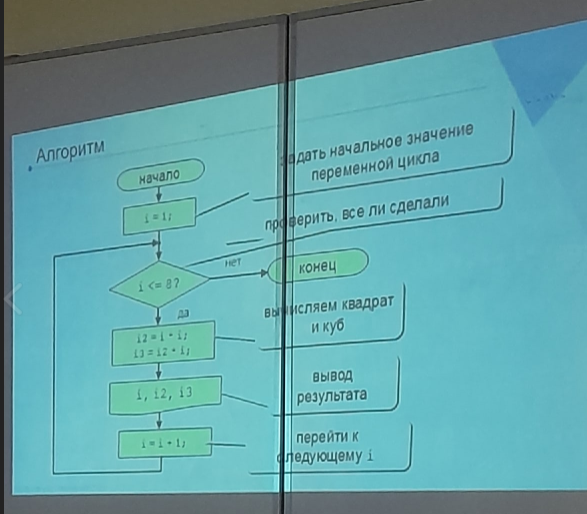
return 0;

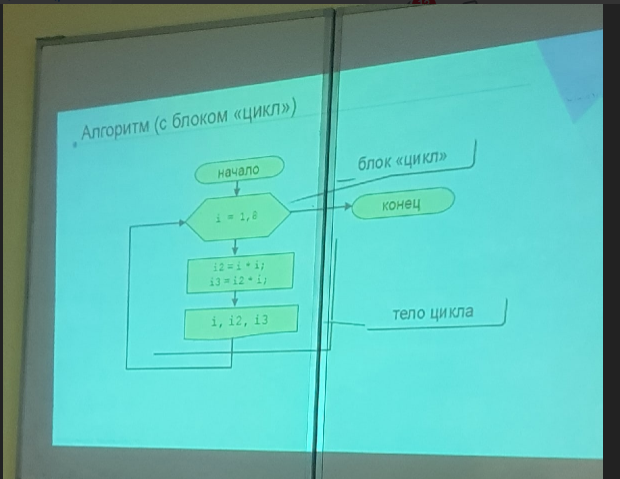
}

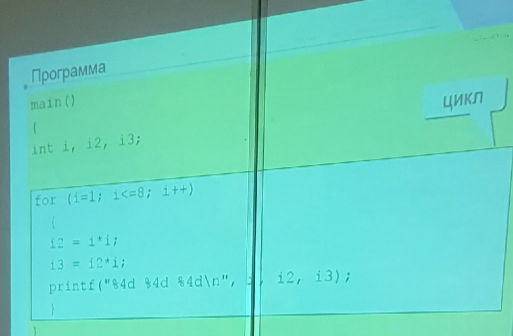
Циклы - многократное повторение последовательности действий

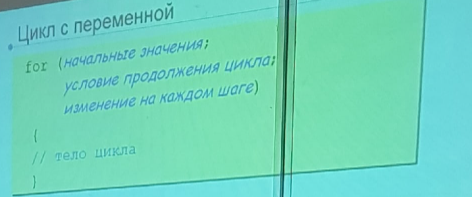
Цикл с известным количеством шагов

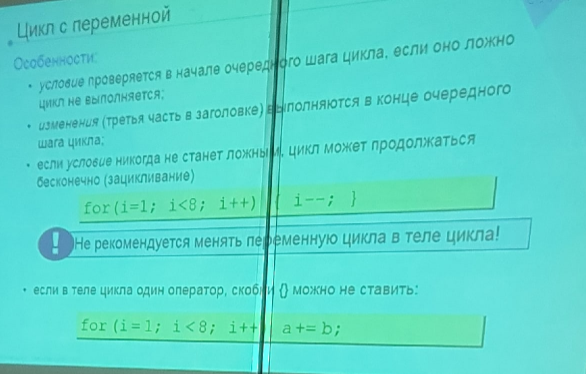
Цикл с неизвестным количеством шагов (с условием)











Алгоритм - это четко определенный план действий для исполнителя

Свойства алгоритма

* Дискретност: состоит из отдельных шагов (команд)
* Понятность - должен включатьтолько команды, известные исполнителю
* Опреденность - при одинаковых исходных данных всегдавыдеает один и тот же результат
* Конечность - заканчивается за конечное число шагов
* Массовость - может примениватьтся многкратно при различных исходных данных
* Корректность -дает верное решение при любых допустимых данных

Программа

Программа - это ...

* Алгоритм,записанный на каком-либо языке програмирования
* Набор команд для компьютера

Команда - это описание действий,которые должен выполнить компьютер

* Откуда взять исходные данные?
* Что нужно с ними сдлелать? - Обрабоать
* Куда поместить результат? - Вывести на экран, сохранить в файле

Языки програмирования

* Машинно-ориентированные (низкого уровня) - каждая команда соответствует одной команде процессора (ассамблер)
* Языки высокого уровня - приблежены к естественному (английскому) языку,легче воспринимаются человеком,не зависят от конкретного компьютера.
* Для обучения: Бейсик, ЛОГО,Паскаль
* Профессиональные: Си, Фортан, Паскаль
* Для задач искусственного интелекта: Пролог,ЛИСП
* Для интерснета: JavaScript, Java, Pert, PHP, ASP

Язык Си

1972-1974- Б.Керниган, Д. Ритчи

Плюсы:

* Высокая скорость работы программ
* Стал основной многих современных языков (С++, С#, JavaScript,Java,ActionScript, PHP)

Минусы:

Порядок програмирования

Main ( )

{

Return 0:

}

Обозначение

# - include < sydioh>

“ ”

H - загаловочный файл

Std - стандарт

i - input

O - outout

( ) - аргументы - Фактические

- Формальные

Ф - функции

Main - функция

{ - begin/ }.- end.

; - пустой оператор. Конец строки

\n - переход на новую строку

\t - горизонтальная табуляция

Вывод текста на экран

#include <stdio.h>

Main ()

{

Print f (“Привет!”);

}

Ждем нажатие любой клавиши

#include <stdio.h>

#include<conio.h>

Main()

{

Printf (“Привет!”); // вывод на экран

Getch (); \*/ ждать нажатия клавиши \*/

}

Переход на новую строку

#include <stdio.h>

#include <conio.h>

Main ()

{

Printf (“Привет, \n Вася!”);

Getch();

}

Задание

1. Вывести на экран текст “Лесенкой”

Вася

\t пошел гулять

\t \t - гулять

1. Вывести на экран рисунок из букв

Ж

ЖЖЖ

Жжжжж

Нн нh

Zzzzzzz

#include <stdio.h>

int main()

{

printf(" ж \n жжж \n жжжжж \nжжжжжжж \n нн нн \n zzzzz");

return 0;

}

Что такое переменная?

Переменная - это ячейка в памяти компьютера, которая имеет имя и хранит некоторое значение.

Значение переменной может меняться во время выполнения программы.

При записи в ячейку нового значения старое стирается

Типы переменных:

Int- целое число (4 байт)

Float -вещественное число (4 байт)

Chat - символ, character (1 байт)

3 Пара

ВВод и вывод

Сложение двух чисел

Задача. Ввести два числа целых и вывести на экран их сумму

Простейшее решение:

#include <stidio.h>

#include <conio.h>

Main ()

{

Int a, b, c;

Printf (“ Введите два числа”); print - вывести

Scanf “%a%d”, &a, &b); ( scan- это скан клавиатуры. F - формат, &взяли адрес. )

(int - %I/%d)

(float - %f)

(char - %c)

(string - %s)

(bool - %b) - логический тип данных

C = a + d; (- в языке Си = это присвоить)

Printf (“%d”, c);

}

4 Пара

Ввод чисел с клавиатуры

Scanf (“%d%d” &a, &b);

Формат - символьная строка,которая показывает какие числа вводятся/выводятся

%d - Целое число

%f - вещественное число

%c - 1 символ

%s - символьная

Вывод чисел на экран

Printf (“%d”, c|;

Printf (“Результат: %d”, c);

Print(“%d+%d+%d=%d”, a, b, c,);

Printf(“%d+%d+%d+%d”, a, b, a+b);

Вывод вещественных чисел

Float x = 123.4567

Printf (“%f”, x);

Минимальное число позиций, 6 цифр дробной части

Printf(“%9.3f”, x);

Полное решение

#include <stdio.h>

#include <conio.h>

Main ()

{

Int a, b, c;

Printf(“Введите два целых числа\n”);

Scanf (“%d%d”, &a, &b);

C = a + b;

Printf (“%d+%d=%d”, a, b, c);

Getch ();

}

Протокол:

Введите два числа целых числа - это выводит компьютер

25 30

25+30=55

Блок-схема линейного алгоритма

Начало -блок начало

Блок a, b - Блок «ввод»

C = a + b; - блок «процесс»

Вывод С - «Вывод»

Конец - блок «Конец»

Задание №4

Ввести три числа, найти их сумму и произведение

Пример:

Введите три числа:

4 5 7

4+5+7=16

4\*5\*7=140

#include “stdafx.h”

//#include <conio.h>

Int main ()

{

Float a, b, c, sum, pr;

Printf(“Введите три числа:\n”);

Scanf(“%f%f%f”, &a, &b, &c);

Sum = a + b + c;

Pr = a \* b \* c;

Printf (“%f + %f + %f = %f/n”, a, b, c pr);

Return 0;

}

1. n - общее количество знаков после запятой

%.2 - неважно знаков перед числом, но два знака после запятой

4 Пара

Переменные

20.11.2021

Имена переменных

Могут включать

Латинские буквы (A-Z, a-z)

Знак подчеркивания \_

Цифры 0-9

!Имя не может начинаться !

Не могут включать

Русские буквы

Пробелы

Скобки, знаки +, =, !, ? и др.

Объявление переменных

Объявить переменную - определить её имя, тип, начальное значение и выделить ей место в памяти

Main ()

{

Int a; - Целая переменная а

}

Что такое переменная?

Переменная - это ячейка в памяти компьютера, которая имеет имя и хранит некоторое значение.

Значение переменной может меняться во время выполнения программы

При запуске в ячейку нового значения старое стирается

Типы переменных:

Int - целое численное (4 байт)

Float - Вещественное число (4 байта)

Char - символ (1 байт)

Объявление переменных

Объявить переменную - определить её имя, тип, начальное значение и выделить ей место в памяти

Main ()

{

Int a;

Float b, c;

Int Tu104, ii86=23, Yak42; - целые переменные Tu104, ii86=23, Yak42 = 23

}

Int a;

Float b, c;

Int Tu104, ii86=23, Yak42;

Float x=4.56, Y, Z; - Вещественные переменные x, y, z. X=4,56

Оператор присваивания

Оператор - это команда языка программирования высокого уровня.

Оператор присваивания служит для изменения значений переменной

Пример: a=5

Оператор присваивания

Общая структура: Имя переменной = выражение.

Арифметическое выражение может включать

Константы

Имена переменных

Знаки арифметических операций:

+ - \* / %. % - остаток от деления.

Вызовы функций

Круглые скобки ()

5 ПАРА

Основные размеры Блок-Схем

Основные блоки, используемые для составления схем алгоритмов, представлены в нормальных документов, главным образом это

Гост 19.003-80 Схемы алгоритмов и программ. Обозначение условные графические

Гост 19.701-90 Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

Размеры сторон

Размер а должен выбираться из ряда 10, 15, 20 мм. Допускается увеличивать размер, а на число, кратное 5 мм. Размер b равен 1.5а.

Основным направлением потока в схемах алгоритмов принято направление сверху - вниз, слева - направо.

Записи внутри символа должны быть представлены так, чтобы их можно было читать слева направо и сверху вниз, независимо от направления потока.

Описание символа должно помещаться справа над символом

Терминатор

Начало, конец, прерывание процесса обработки данных или выполнение программы.

Процесс

Выполнение операции или группы операций, в результате которых имеет значение, форма представление или расположение данных

Предопределенный процесс

Использование ранее созданных или отдельно описанных алгоритмов или программ.

Ввод - вывод

Преобразование данных в форму, пригодную для обработки (ввод) или отображение результатов обработки (вывод).

Условие (решение)

Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий.

Цикл (Подготовка)

Выполнение операций, меняющих команды или группу команд, с целью воздействия на некоторую последующую функцию (установка переключателя, модификация регистра, инициализация программы)

Соединитель

При большой насыщенности схемы отдельные линии потока между удаленными символами допускается обрывать. При этом в конце (начале) обрыва должен быть помещен символ (Соединитель) Внутри блока соединителя указывается имя уникального идентификатора.

6 Пара

Развлекающиеся алгоритмы

Алгоритм, в которых последовательность шагов зависит от выполнения некоторых условий, называется развлекающимися

Вариант 1. Программа

Main ()

{

Int a, b, max;

Printf(“Введите два целых числа\n”);

Scanf(“%d%d”, &a, &b);

If (a > b) {

Max = a;

}

Else {

Max = b;

}

Printf(“Наибольшее число %d”, max);

}

7 Пара

Процедура - это алгоритм который не обязан возвращать значение

Пустой оператор - это точка с запятой. Не производит никаких особых действий

## 05.02.2022

Циклы с условием

[#include](https://vk.com/im?sel=2000000047&st=%23include) <conio.h>  
[#include](https://vk.com/im?sel=2000000047&st=%23include) <stdio.h>  
int main()  
{  
int n, count=0;  
printf("Введите число: ");  
scanf("%i", &n);  
while (n != 0)  
{  
n = n / 10;  
count = count + 1;  
  
}  
{  
printf("%i", count);  
}  
return 0;  
}

Цикл с условием

While (условие)

{

// тело цикла

}

Особенности

1. Можно использовать сложные условия;

While (a<b && b<c) {….}

1. Если в теле цикла только оператор скобки {} можно писать:

While (a<b) a++

3)Условие пересчитывается каждый раз при входе в цикл

4)Если условие на входе в цикл должно, цикл не выполняется ни разу

a=4; b=6;

While (a>b) a=a-b;

5)Если условие никогда не станет ложным, программа зацикливается

a=4; b=6;

While (a<b) d=a+b;

Предусловие

While (условие)

{.…}

Постусловие

do

{.….}

While (условие);

#include <stdio.h>

int main()

{

float a, z= -1, b=1, c=2, S=1;

a = z\*b\*c;

while (fabs (a) > 0.001)

{

S = S+a;

z = z\*(-1);

c=2\*c;

b=b+1;

a = z\*b/c;

}

printf ("%f\n", S);

return 0;

}

# Массив

# 12.03.2022

Статические (заранее знаю размер)

Динамические:

- new

- realloc

-malloc

-calloc

Int M [10]

For (int I = 0; I<10; I++)

Scanf(“&d”, &M[I]);

For (int I = 0; I<7; I++)

N[I] = rand() % 4-9;

For (int I = 0; I<9; I++)

K[I] = I ++ \* -- I\*z;

Printf (“Введите n”);

Scanf(“%d”, &n);

Int \*M = new int [n]; знак \* - указатель на массив.

Delete []M

#include <stdio.h>

int main()

{

int \*M, n, i\_max = 0, i\_min = 0;

printf ("Введите количество элементов \n");

scanf ("%d", &n);

M = new int [n];

for (int i=1; i<n;i++);

{

int max = min = M[0];

if (M[i] > max )

{

max = M[i];

i\_max = i;

}

if (M[i] < min)

{

min = M[i];

i\_min = i;

}

}

int temp = M [i\_min];

M [i\_min] = M[i\_max];

M [i\_max] = temp;

for (int i=0; i<n; i++)

printf (" &d", M[i]);

return 0;

}

# Домашнее задание

Описать и рассказать о calloc

Malloc

Rallor

Массивы

Одномерный массив: Используя функции поменять max и min

Динамический массив из n элементов

Max () -

Min () -

Swap () -

Enter () - Ввод массива

Printf () - Вывод массива

Main ()

Float \*M

Int n