**13-14 ОАП Пр2 Составление программ линейным структуры**

**Цель работы: научиться писать программы линейной структуры.**

Каждая программа на С++ начинается с одного или нескольких операторов **#include**. Эти операторы указывают компилятору включить содержимое данного заголовочного файла в текст программы. Заголовочные файлы (файлы с расширением **.** **h ― head files**) содержат определенные операторы С++ или программные определения. В таблице 1.1 перечислены некоторые заголовочные файлы стандартной библиотеки.

Табл 1.1.

|  |  |
| --- | --- |
| <iostream.h> | Стандартные входные и выходные потоки |
| <fstream.h> | Файловые входные и выходные потоки |
| <iomanip.h> | Манипуляторы |
| <math.h> | Математические функции |
| <string.h> | Функции для работы со строками |
| <stdlib.h> | Функции стандартной библиотеки |
| <stdio.h> | Стандартный ввод/вывод (в стиле С) |
| <conio.h> | Функции управления выводом |
| <graphics.h> | Графические функции |

Программа состоит из операторов. Программа с **линейной структурой** – это программа, в которой все операторы выполняются по порядку сверху вниз. Любая программа должна обеспечить выполнение поставленной задачи. Для выполнения расчетов может понадобиться заголовочный файл, содержащий описание математических или иных функций. Их подключают с помощью оператора **include**, например,

**#include <math.h>**

**#include <iostream.h>**

Некоторые стандартные функцииперечислены в таблице 1.2.

Табл 1.2.

| Функция | Назначение | Заголовочный файл |
| --- | --- | --- |
| abs(x) | Возвращает модуль числа *x*, если *x* – целое число | <stdlib.h> |
| acos(x) | Возвращает арккосинус *х* | <math.h> |
| asin(x) | Возвращает арксинус *х* | <math.h> |
| atan(x) | Возвращает арктангенс *х* | <math.h> |
| ceil(x) | Округляет число *х* вверх | <math.h> |
| cos(x) | Вычисляет косинус *х* | <math.h> |
| exp(x) | Возвращает степень числе *е* | <math.h> |
| fabs(x) | Возвращает модуль числа *х*, если *x* – вещественное число | <math.h> |
| floor(x) | Округляет число *х* вниз | <math.h> |
| log(x) | Вычисляет натуральный логарифм *х* | <math.h> |
| log10(x) | Вычисляет логарифм *х* по основанию 10 | <math.h> |
| pow(x,y) | Возводит число *х* в степень *у* | <math.h> |
| sin(x) | Вычисляет синус *х* | <math.h> |
| sqrt(x) | Вычисляет квадратный корень *х* | <math.h> |
| tan(x) | Возвращает тангенс *х* | <math.h> |
| getchar | Читает символ из стандартного ввода (с клавиатуры) | <stdio.h> |
| gets | Читает символы с клавиатуры до появления символа новой строки | <stdio.h> |
| atof(s) | Преобразует строку *s* в вещественное число | <stdlib.h>  <string.h> |
| atoi(s) | Преобразует строку *s* в целое число | <stdlib.h>  <string.h> |
| itoa(v,s,n) | Преобразует целое число *v* в строку *s* в системе счисления по основанию *n* | <stdlib.h>  <string.h> |
| random(n) | Генерирует случайные числа в диапазоне от 0 до *n* | <stdlib.h> |
| strcat(s1, s2) | Присоединяет строку *s2* к строке *s1* | <string.h> |
| strcmp(s1, s2) | Сравнивает строки *s1* и *s2*.  Результат: -1, если *s1* < *s2*  0, если *s1* = *s2*  1, если *s1* > *s2* | <string.h> |
| strchr(s,c) | Ищет первое вхождение символа *c* в строке *s*. Возвращает указатель на найденный символ или NULL | <string.h> |
| strcpy(s1, s2) | Копирует строку *s2* в строку *s1* | <string.h> |
| strlen(s) | Возвращает длину строки *s* | <string.h> |
| strncat(s1, s2,n) | Присоединяет *n* символов строки *s2* к строке *s1* | <string.h> |
| strncmp(s1, s2,n) | Сравнивает *n* символов строки *s2* и строки *s1*. Результат аналогичен strcmp | <string.h> |
| strstr(s1, s2) | Ищет подстроку *s2* в строке *s1*. Возвращает указатель на начало найденного фрагмента в *s1* или NULL | <string.h> |
| strupr(s) | Выводит строку *s* в верхнем регистре | <string.h> |
| strlwr(s) | Выводит строку *s* в нижнем регистре | <string.h> |

Все переменные, используемые в программе, должны быть объявлены. Объявить переменную – это значит указать ее тип и имя (идентитфикатор). Идентификаторы используются для обозначений имен переменных, функций, меток. Идентификатором может быть любая последовательность латинских символов (заглавных и строчных), цифр и символа подчеркивания произвольной длины.

Общий вид оператора объявления:

**<тип переменной> <имя переменной>;**

Основные типы переменных перечилены в таблице 1.3. (“e” означает десять в степени…).

Табл 1.3

| Тип | Назначение | Размер в байтах |
| --- | --- | --- |
| **char** | (символ), всего 256 значений | 1 |
| **int** | целое число | 4 |
| **unsigned** | целое положительное число | 4 |
| **float** | вещественное число | 4 |
| **double** | вещественное число двойной точности | 8 |

Примеры объявления переменных:

**int age=17, i, j;** объявлены три переменные целого типа: **i, j, age**. Переменная **age**

инициализирована при объявлении – ей присвоено начальное

значение, равное 17;

**float a, b, f;** объявлены три переменные вещественного типа.

Алгоритм для решения поставленной задачи можно записать в виде геометрических фигур или блоков. Обычно при составлении блок-схемы блоки размещаются друг под другом в порядке их выполнения. Возврат назад осуществляется только в циклах. Это дает простую и наглядную структуру алгоритма, по которой легко составить программу. Основные виды блоков:

|  |  |
| --- | --- |
| Начало, завершение | Вычисление |
| Условие | Ввод, вывод |
| Начало цикла «для каждого» | Подпрограмма или функция |
| Конец цикла «для каждого» | Соединение внутри страницы, n – номер соединения |
| Выбор действий,  n – ключ выбора  n | к стр. k  Межстраничное соединение, n – номер соединения, k – номер страницы |
| Соединение между блоками (при направлении сверху - вниз и слева – направо стрелки не ставятся) | из стр. k  Межстраничное соединение, n – номер соединения, k – номер страницы |

В программе для ***вывода*** результатов на экран используют оператор следующего вида:

**сout << “ комментарий ” << идентификатор 1 << идентификатор 2<<…<< endl;**

# Здесь знак “<<” означает оператор вставки, который вставляет в выходной поток все, что следует за этим знаком. Комментарий заключают в двойные кавычки.

# Для ***ввода*** с клавиатуры используют следующий оператор:

**cin >> идентификатор 1 >> идентификатор 2>> … >> идентификатор n;**

Здесь знак “>>” означает оператор извлечения, который извлекает из входного потока (по умолчанию – это клавиатура) число или символ и помещает в указанный идентификатор. Каждый оператор заканчивают точкой с запятой.

При записи выражений следует учитывать порядок выполнения операций. В таблице 1.4 приведены некоторые операции С++ в порядке убывания приоритета. Операции с разными приоритетами разделены жирной чертой.

Табл 1.4

|  |  |  |
| --- | --- | --- |
| Операция | Краткое описание | Пример |
| **::** | доступ к области видимости | ::а |
| **.** | выбор | student**.**name |
| **->** | выбор | student**->**name |
| [ ] | индексация | a [ i ] |
| ( ) | вызов функции | sin(x), main(void) |
| ++ | инкремент (увеличение на единицу) | a++ или ++a |
| -- | уменьшение на единицу | a-- или --a |
| & | взятие адреса | &a |
| \* | разыменование | \*a |
| ~ | поразрядное отрицание | ~a |
| ! | логическое НЕ | !a |
| \* | умножение | a \* b |
| / | деление | a / b |
| % | остаток от целочисленного деления | a % b |
| + | сложение | a + b |
| - | вычитание | a - b |
| << | сдвиг влево; сдвигает биты (разряды) влево на указанное количество разрядов | a<<1 |
| >> | сдвиг разрядов вправо | a>>1 |
| < | меньше, чем | a<b |
| <= | меньше или равно | a<=b |
| > | больше, чем | a>b |
| >= | больше или равно | a>=b |
| = = | равно | a= = b |
| ! = | не равно | a !=b |
| & | побитовое И | a&b |
| | | побитовое ИЛИ | a | b |
| ^ | побитовое исключающее ИЛИ | a ^ b |
| && | логическое И | a&& b |
| | | | логическое ИЛИ | a | | b |

### Решить задачи

1. Вычислить координаты центра тяжести системы трех материальных точек с массами *m1, m2, m3* и координатами *(x1, y1), (x2,y2), (x3,y3)* по формулам:

*x = (m1x1+ m2x2 +m3x3)/ (m1+m2+m3)*

*y = (m1y1 + m2y2 +m3y3) / (m1+m 2+ m3)*

1. Вычислить медианы треугольника со сторонами *a, b, c* по формулам:



1. Вычислить значение функции



1. Вычислить корни системы уравнений:  по формулам:



1. Дана длина ребра куба. Найти объем куба, площадь его боковой поверхности и площадь грани.
2. Найти площадь кольца, внутренний радиус которого равен 20, а внешний – заданному числу *r (r>20).*