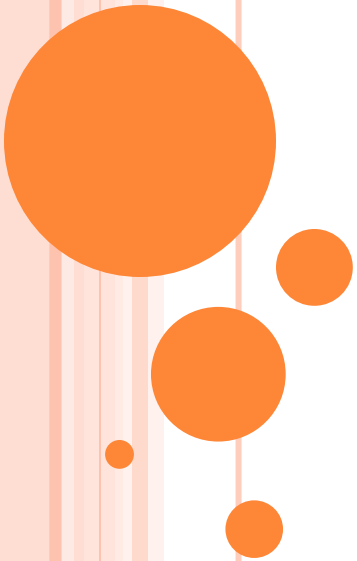


Типы данных, константы и переменные, выражения

Лекция №2

План лекции

- Представление данных в памяти компьютера – понятие типа данных
- Константы
- Переменные
- Ввод-вывод
- Операции и выражения
- Работа в режиме калькулятора

Исходные предпосылки

- Все операции в программе выполняются над данными
- Данные хранятся в памяти в определенном формате и занимают конечный объем
- Содержимое ячейки памяти интерпретируется по-разному в зависимости от формата (типа данных)
- Одни и те же операции над данными в разных форматах выполняются по-разному

Тип данных

- Соглашение о хранении данных в памяти компьютера
  - Формат
  - Длина в байтах
  - Диапазон значений
  - Правила выполнения операций
  - Служит для создания программных объектов
    - Констант
    - Переменных

Основные типы данных

- Стандартные (идея заложена в компилятор)
  - Целочисленный
    - Обычный (int)
    - Беззнаковый (unsigned)
  - Вещественный
    - Обычный (float)
    - Двойной точности (double)
  - Символьный
    - char
- Пользовательские (составлены программистом)
  - Структуры (struct)
  - Объединения (union)
  - Классы (class)
  - Перечислимый тип (enum)

Тип int

- 4 байта -32 бита
  - Бит 31 – знак (0 – плюс, 1 – минус)
  - Биты 0-30 – число
- Диапазон
  - -2^{31} - $2^{31}-1$
- Точное представление данных
- Рекомендован, если диапазон заранее известен
- Высокая скорость вычислительных операций
- Возможна потеря дробной части (10/3 = 3 – грубая погрешность)
- Возможно переполнение (2^{30} * 100 = ошибка!!!)

Тип unsigned

- 4 байта -32 бита
  - Биты 0-31 – число
- Диапазон
  - $0 - 2^{32}-1$
- Те же свойства, что и у int
- Расширение в 2 раза диапазона положительных чисел за счет отказа от отрицательных чисел
- Рекомендован для представления чисел, которые по смыслу задачи не могут быть отрицательными

Тип float

- 4 байта -32 бита
  - Двоичная мантисса (основание) (a)
  - Двоичная экспонента (порядок) (b)
  - Форма $a * 2^b$
- Диапазон
  - $-10^{38} – -10^{-38}$ И $10^{-38} – 10^{38}$
- Рекомендован, если диапазон заранее неизвестен
- Сохранение дробной части при делении
- Высокая трудоемкость вычислительных операций
- Неопределенность представления 0 – возможны ошибки при проверке элемента данных на равенство или неравенство
- Возможно накопление вычислительных погрешностей
- Предпочтителен для представления исходных данных и результатов

Тип double

- 8 байт -64 бита
  - Двоичная мантисса (основание) (a)
  - Двоичная экспонента (порядок) (b)
  - Форма $a * 2^b$
- Диапазон
  - $-10^{308} – -10^{-308}$ И $10^{-308} – 10^{308}$
- Рекомендован, если диапазон заранее неизвестен
- Предпочтителен для представления промежуточных данных
- Те же свойства, что и у float

Тип char

- 1 байт -8 бит
  - Бит 7 – знак (0 – плюс, 1 – минус)
  - Биты 0-6 – число
- Диапазон
  - -2^7 - 2^7-1
- Те же свойства что и у int
- Служит как для представления кодов символов, так и обычных целых чисел
- Устанавливает соответствие между кодами символов и их графическими образами

Для чего нужны типы данных?

- Для создания программных объектов
  - Констант
  - Переменных
- Над константами и переменными выполняются действия с учетом их типов данных (форматов)
- Значения констант и переменных могут вводиться с клавиатуры и выводиться на экран с учетом их форматов

Константы

- Целочисленные
  - Десятичные 12, -123, 0
  - Восьмеричные 024, 015, 0321
  - Шестнадцатеричные 0x10, 0x1A, 0XFFED
- Вещественные
  - 1.25, -0.234, 3.1E-3 ($3.1*10^{-3}$)
- Символьные
  - Печатаемые ‘A’, ‘b’, ‘+’,’1’
  - Управляющие ‘\n’,’\t’, ‘\\’
- Строковые
  - “Informatika”,”Semester-2\n”,”12345”
- Нуль-символ NULL

Задание констант

- С помощью директивы препроцессора
  - #define N 1024
  - #define G 9.81
  - #define STR “Program1”
- С помощью модификатора const
  - const int D=1;
  - const float E=2.1E5;
  - const char C=“!”;

Переменная

- Поименованная область памяти, значение которой может быть изменено
  - Тип данных
  - Имя переменной (вместо адреса в памяти)
  - Начальное значение (необязательно)
- Формат описания
  - тип имя=нач_значение;
- Примеры переменных
  - int a=3,b;
  - float f=2.1;
  - double e;
  - char c='A';
- Если не указано начальное значение, то после запуска программы значение переменной случайное

Правила написания имен переменных

- Только латинские буквы, цифры и _
- Имя начинается с буквы
- Большие и маленькие буквы считаются разными
  - (B и b – разные переменные)
- Имя должно быть уникальным

Области видимости переменных

- Глобальная переменная вне функций
  - вся программа
- Локальная переменная
  - Отдельная функция, например, main

- Определение переменной должно быть выше места использования

Время жизни переменной

- автоматическая переменная – от входа в функцию до возврата из нее
  - int f=3;//по умолчанию – все переменные автоматические
- статическая переменная – полное время работы программы
  - static float a=0;//имеет модификатор static
- динамическая переменная – от команды создания до команды уничтожения
  - int* p;//определение указателя
  - p= new int;//создание переменной с сохранением адреса
  - delete p;//уничтожение переменной.

Основные действия над переменными

- Присваивание значения
- Ввод значения с клавиатуры
- Выполнение различных операций
- Вывод значения на экран

- Передача в функцию в качестве параметра
- Возврат из функции в качестве результата

Ввод-вывод

- позволяет визуализировать исходные данные и результаты
- позволяет проверить работоспособность программы
- самый простой – консольный ввод-вывод в текстовом режиме
- С помощью функций из стандартных библиотек
- stdio.h
  - printf
  - scanf
- iostream
  - cout
  - Cin
- На .NET платформе - класс Console
  - Conlole.Read
  - Console.Write

Вывод (stdio.h)

- #include<stdio.h>
- .....
- printf(“Форматная строка”,переменная)
  - printf(“a=%d\n”,a);
- Форматы
  - %d – целочисленный
  - %f – вещественный обычный
  - %e – вещественный экспоненциальный
  - %c – символьный
  - %s – строковый
- Управляющие
  - \n – перевод строки
  - \t – табуляция – отступ на 9 позиций влево

Ввод (stdio.h)

- scanf(“Формат”,&переменная);
  - & - операция взятия адреса
  - В форматной строке ничего лишнего, кроме формата

  - #include<stdio.h>
  - .....
  - scanf(“%d”,&b);

Ввод-вывод (iostream)

- #include<iostream.h>
- .....
- cout << “Vvedite a”;
- cin>>a;
- cout << “a=“<<a<<“\n”;

Присваивание значений переменным

- переменная = выражение;
  - значение выражения справа присваивается переменной слева
  - a=b*2+3;

Арифметические операции

- Сложение (+)
- Вычитание (-)
- Умножение (*)
- Деление (-)
  - при делении целых чисел теряется дробная часть
- Только для целых чисел
  - Вычисление остатка от целочисленного деления (%)
  - Увеличение на 1 (++)
  - Уменьшение на 1(--)
  - ++ и – работают быстрее, чем + и –

Примеры арифметических операций над целыми числами

- int a=10, b =3, c, d, e, f, k, g=0, h=2;
- printf(“a=%d, b=%d\n”,a,b);
- c=a+b;
- d=a-b;
- e=a*b;//возможно переполнение
- f=a/b; //потеря дробной части!
- k=a%b; //только для целых чисел
- printf(“c=%d,d=%d,e=%d,f=%d,k=%d\n”,c,d,e,f,k);
- g++; //только для целых чисел
- h--; //только для целых чисел

Примеры арифметических операций над вещественными числами

- float a1=10, b1=3, c1, d1, e1, f1;
- printf(“a1=%f, b=%f\n”,a1,b1);
- c1=a1+b1;
- d1=a1-b1;
- e1=a1*b1;
- f1=a1/b1;
- printf(“c1=%f,d1=%f,e1=%f,f1=%f\n”,c1,d1,e1,f1);

Преобразование типов

- Получение временной копии переменной в другом формате

- float z;
- int x=20,y=6;
- z=x/y;//x и y – целые, дробная часть теряется

- float z;
- int x=20,y=6;
- z=(float)x/y;// получение временной копии
- //переменной x в вещественном формате

Правила преобразования типов

- Без потерь происходят преобразования от более короткого типа к более длинному (int -> float)
- Преобразование более длинного к более короткому типу приводит к ошибкам (double->int)
- Преобразование типа не распространяется на строки ниже него.

Составное присваивание

- Изменение значения переменной
- Изменяемые переменные должны иметь известное начальное значение
- int a=1, b=2;
- a=a+5;// или a+=5;
- b=b*3;// или b*=3;

- Составное присваивание используется в циклах для накопления, подсчета или задания шага изменения переменной цикла

Стандартные математические функции

- math.h
  - Тригонометрические sin,cos,tan,asin,acos,atan
    - Аргументы в радианах
    - ctg (x) = 1/ tg (x)
  - Логарифмы log, log10
    - $\log_a b = \log(b)/\log(a)$
  - Степени pow, sqrt,exp
    - pow(a,b)
    - sqrt(a)
    - exp(a)
  - Модуль abs
    - abs(a)

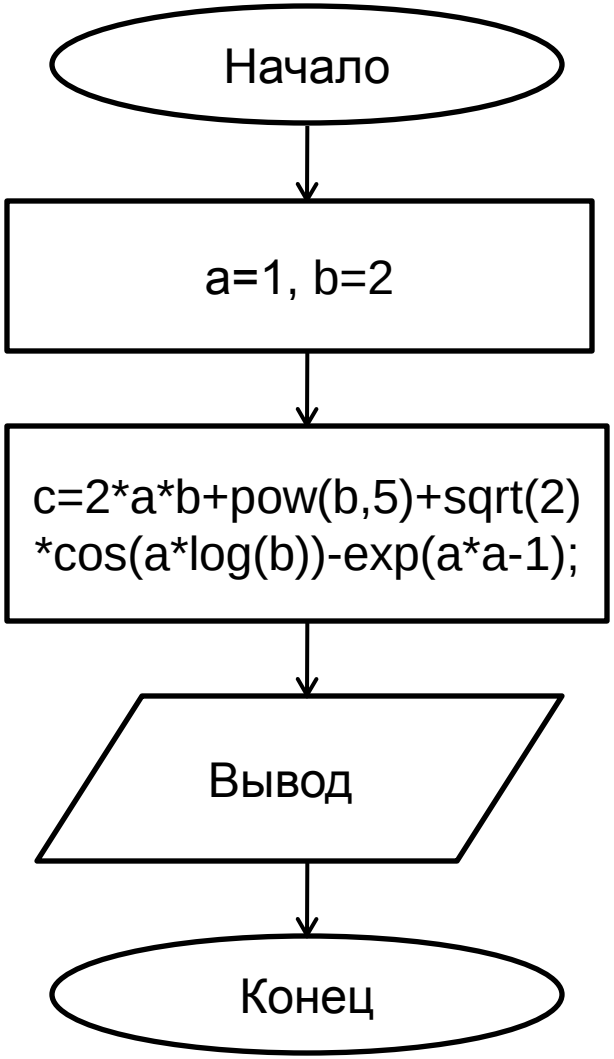
Программирование в режиме калькулятора

- Задано выражение

$$c = 2ab + b^5 + \sqrt{2}\cos\left(a\ln(b)\right) - e^{a^2-1}$$

- Написать программу для вычисления выражения для a=1, b=2;

Алгоритм – последовательность действий



Программа для вычисления арифметического выражения

```
#include <stdio.h> //ввод-вывод
#include <conio.h> //управление вводом-выводом
#include <math.h>
/*Главная функция - обязательная*/
void main(void)//заголовок
{  double a=1, b=2,c;//подходящий тип для math.h
   c=2*a*b+pow(b,5)+sqrt(2)*cos(a*log(b))-exp(a*a-1);
    printf("c=%f", c);//вывод на экран сообщения
    getch();//ожидание нажатия клавиши
}
```

Заключение

- Выбор типов данных определяется смыслом переменных
- Прежде чем использовать, переменную необходимо определить
- С именами переменных работать удобнее, чем с адресами оперативной памяти
- Целочисленное деление влечет потерю дробной части
- Целочисленные операции могут привести к переполнению
- Математические функции берутся из стандартной библиотеки