Вычислительная техника и компьютерное моделирование в физике

Лекция 2

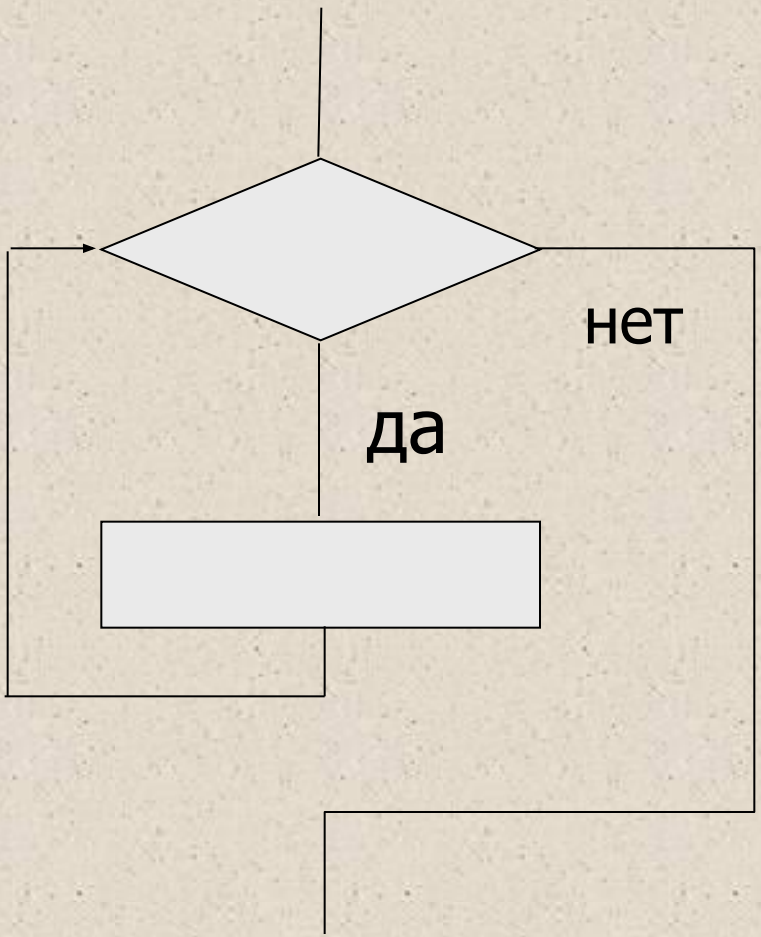
Зинчик Александр Адольфович
zinchik_alex@mail.ru

Операторы цикла

- while (expression) operator
- do operator while (expression)
- for(init_expr;cond_expr;increment_expr)

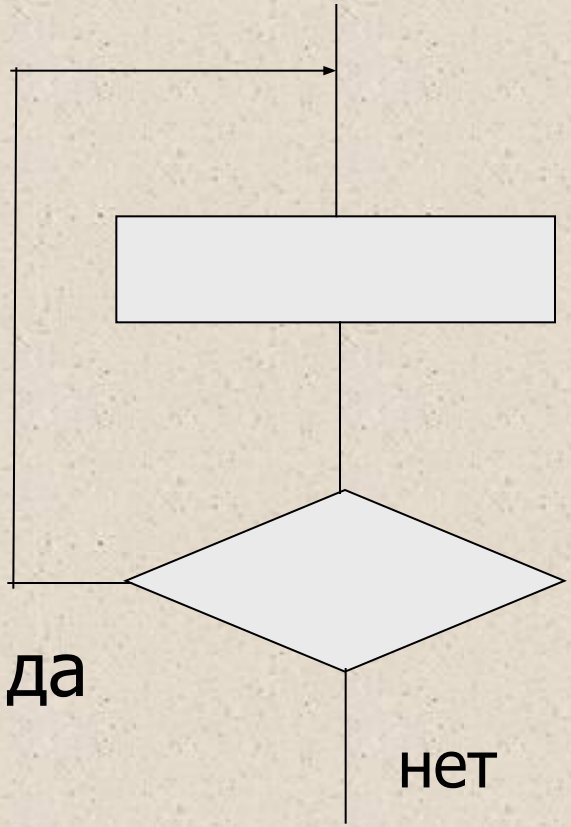
Цикл с предусловием (while)

while (выражение) оператор



Цикл с постусловием

do operator while (expression)



- Пример (программа печатает таблицу значений функции $y=x^2+1$ во введенном диапазоне):

```
#include <stdio.h>
int main()
{
   float Xn, Xk, Dx;
   printf(“Введите исходные данные\n”);
   scanf(“%f%f%f”, &Xn, &Xk, &Dx);
   printf("|   X   |   Y   |\n");
   float X = Xn;
```

```
while (X <= Xk)
{
   printf("| %5.2f | %5.2f |\n", X, X*X + 1);
   X += Dx;
}
 return 0;
}
```

Пример (программа находит все делители целого положительного числа):

```
#include <stdio.h>

int main()
{
  int num;
  printf("\nВведите число : “);
  scanf(“%d”, &num);
  int half = num / 2; // половина числа
  int div = 2;        // кандидат на детель
```

```
while (div <= half){
    if (!(num % div))
            printf("%d\n", div);
    div++;
   }
  return 0;
}
```

Цикл с постусловием (do while)

- **do оператор while выражение;**
- Сначала выполняется простой или составной оператор, составляющий тело цикла, а затем вычисляется выражение. Если оно истинно (не равно false), тело цикла выполняется еще раз. Цикл завершается, когда выражение станет равным false или в теле цикла будет выполнен какой-либо оператор передачи управления. Тип выражения должен быть арифметическим или приводимым к нему.

Пример (программа осуществляет проверку ввода):

```
#include <string.h>
#include <stdio.h>
void main()
{
   char answer[15];
   do{
        printf( “\nХочу печенья! \n”);
             scanf(“%15s”, answer);
       }while (strcmp(answer , “печенье”)!=0);
  }
}
```

Цикл с параметром (for)

- Цикл с параметром имеет следующий формат:

- **for (инициализация; выражение; модификации) оператор;**

- *Инициализация* используется для объявления и присвоения начальных значений величинам, используемым в цикле. В этой части можно записать несколько операторов, разделенных запятой (операцией «последовательное выполнение»), например:

- for (int i = 0, j = 2;
- int k, m;
- for (k = 1, m = 0;

- **Областью действия** переменных, объявленных в части инициализации цикла, является цикл. Инициализация выполняется один раз в начале исполнения цикла.
- ***Выражение*** определяет условие выполнения цикла: если его результат, приведенный к типу bool, равен true, цикл выполняется. Цикл с параметром

- *Модификации* выполняются после каждой итерации цикла и служат обычно для изменения параметров цикла. В части модификаций можно записать несколько операторов через запятую. Простой или составной *оператор* представляет собой тело цикла. Любая из частей оператора for может быть опущена (но точки с запятой надо оставить на своих местах!).

- Пример (оператор, вычисляющий сумму чисел от 1 до 100):

```
for (int i = 1, s = 0; i<=100; i++)
                s += i;

s = 0;
for (int i = 1; i<=100; i++)
                s += i;
```

- Пример (программа печатает таблицу значений функции y=x2+1 во введенном диапазоне):

```
#include <stdio.h>

void main()
{
  float Xn, Xk, Dx;
    printf(“Введите исходные данные\n”);
    scanf(“%f%f%f”, &Xn, &Xk, &Dx);
  printf("|   X   |   Y   |\n");
                  // шапка таблицы
```

```
for (X = Xn; X<=Xk; X += Dx)

printf("| %5.2f | %5.2f |\n", X, X*X + 1);

// тело цикла

 return 0;
}
```

Пример (программа находит все делители целого положительного числа):

```
#include <iostream>
void main()
{
  int num, half, div;
  cout<<"\nВведите число : “<<endl;
  cin>>num;
  for (half=num/2, div=2; div<= half; div++)
   if (!(num%div)) cout>>div;
return 0;
}
```

- Любой цикл while может быть приведен к эквивалентному ему циклу for и наоборот по следующей схеме:

for (b1; b2; b3) оператор	b1; while (b2){ оператор; b3;}

Массивы

```
#include <iostream.h>
int main(){
   const int n = 10;
   int marks[n] = {3, 4, 5, 4, 4};
   int i, sum;
   for ( i = 0, sum = 0; i<n; i++)
      sum += marks[i];
   cout << "Сумма элементов: " << sum;
}
```

```
   int a[100], b[100];
   int *pa = a;   // или int *p = &a[0];
   int *pb = b;
   for(int i = 0; i<100; i++)*pb++ = *pa++;
      // или pb[i] = pa[i];
```

```
float p[10]; *u[20];
int a[5] = {1, 2, 3};
int b[] = {1, 2, 3};
char cv[4] = { 'a', 's', 'd', 'f', 0 };          //  error
```

```
p[5]      5[p]      *(p+5)
```

Пример - сортировка выбором

```
#include <iostream.h>
int main(){
    const int n = 20;          int b[n];      int i;
    for (i = 0; i<n; i++) cin >> b[i];
    for (i = 0; i<n-1; i++){
        int imin = i;
        for (int j = i + 1; j<n; j++) if (b[j] < b[imin]) imin = j;
        int a = b[i]; b[i] = b[imin];  b[imin] = a;
    }
    for (i = 0; i<n; i++)cout << b[i] << ' ';
    return 0;
}
```

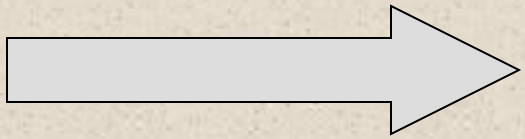
Динамические массивы

```
float *p = new float [100];
float *q = (float *) malloc(100 * sizeof(float));
```

```
delete [] p; free
        (q);
```

Многомерные массивы

int matr [6][8];



```
matr[i][j]
*(matr[i]+j)
*(*(matr+i)+j)
```

```
int mass2 [][2]={ {1, 1}, {0, 2}, {1, 0} };
int mass2 [3][2]={1, 1, 0, 2, 1, 0};
```

```
int x3d[3][5][7];
float y[4][3] = {  { 1 }, { 2 }, { 3 }, { 4 } };
// первый столбец, остальные 0
```

```
int nstr = 5;
int ** m = (int **) new int [nstr][10];
```

```
int nstr, nstb;
cout << " Введите количество строк и столбцов :";
cin >> nstr >> nstb;
int **a = new int *[nstr];          // 1
for(int i = 0; i<nstr; i++)         // 2
   a[i] = new int [nstb];           // 3
```

Заключение

- Спасибо за внимание!
- Вопросы???