Динамические структуры данных (язык Си)

Тема 1. Указатели

Статические данные

```
int x, y = 20;
float z, A[10];
char str[80];
```

- **переменная (массив) имеет имя, по которому к ней можно обращаться**
- **размер заранее известен (задается при написании программы)**
- **память выделяется при объявлении**
- **размер нельзя увеличить во время работы программы**

Динамические данные

- размер заранее неизвестен, определяется во время работы программы
- память выделяется во время работы программы
- нет имени?

Проблема:

как обращаться к данным, если нет имени?

Решение:

использовать адрес в памяти

Следующая проблема:

в каких переменных могут храниться адреса?
как работать с адресами?

Указатели

Указатель – это переменная, в которую можно записывать адрес другой переменной (или блока памяти).

Объявление:

```
char  *pC; // адрес символа
           // (или элемента массива)
int   *pI; // адрес целой переменной
float *pF; // адрес вещественной переменной
```

Как записать адрес:

```
int m = 5, *pI;
int A[2] = { 3, 4 };
pI = &m;     // адрес переменной m
pI = &A[1]; // адрес элемента массива A[1]
pI = NULL;  // нулевой адрес
```

```
scanf("%d", &m);
```

Обращение к данным

Как работать с данными через указатель?

```
int m = 4, n, *pI;
pI = &m;
printf ("m = %d", *pI); // вывод значения
n = 4*(7 - *pI);        // n = 4*(7 - 4) = 12
*pI = 4*(n - m);        // m = 4*(12 - 4) = 32
printf("&m = %p", pI); // вывод адреса
```

«вытащить» значение по адресу

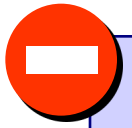
Как работать с массивами?

```
int *pI, i, A[] = {1, 2, 3, 4, 5, 999};
pI = A;       // адрес A[0] записывается как A
while ( *pI != 999 ) { // while( A[i] != 999 )
  *pI += 2; // A[i] += 2;
  pI++;     // i++ (переход к следующему)
  }
```

Оператор `pI++` увеличивает адрес на `sizeof(int)`!

Что надо знать об указателях

- указатель – это переменная, в которой можно хранить адрес другой переменной;
- при объявлении указателя надо указать тип переменных, на которых он будет указывать, а перед именем поставить знак ***;**
- знак **&** перед именем переменной обозначает ее адрес;
- знак * перед указателем в рабочей части программы (не в объявлении) обозначает значение ячейки, на которую указывает указатель;
- для обозначения недействительного указателя используется константа ***NULL*** (нулевой указатель);
- при изменении значения указателя на ***n*** он в самом деле сдвигается к *n*-ому следующему числу данного типа, то есть для указателей на целые числа на ***n*sizeof(integer)*** байт;
- указатели печатаются по формату ***%p***.



Нельзя использовать указатель, который указывает неизвестно куда (будет сбой или зависание)!