

Средняя общеобразовательная школа № 654 имени А.Д. Фридмана

Общие представления о динамической памяти

Программирование

11 класс

Профильная подготовка

Учитель: *Ермаков Максим Геннадьевич*

Для чего программе память?

Для хранения:

- самой программы;
- определяемых пользователем констант и структур данных;
- точек возврата из процедур и функций;
- временного хранения промежуточных результатов вычислений, а также исходных данных и результатов работы при выполнении операций ввода и вывода.

Какая бывает память программы?

- **Статическая**

- для хранения значений переменных, объявленных в блоке описания программы.
- Размер ограничен 64 КБ

- **Динамическая**

- оперативная память, выделяемая программе за вычетом сегмента данных (64 Кб), стека (16 Кб) и собственно тела программы.
- Размером можно управлять.

Какие бывают переменные?

- *Статические*
 - объявлены в блоке описания программы.
 - Память для хранения значений выделяется ДО запуска программы и не изменяется в процессе её работы.
 - Компилятор может их обрабатывать без запуска программы.
- *Динамические*
 - переменные, память для хранения значений которых выделяется и освобождается в процессе работы программы.

Какие бывают переменные?

- Особый вид переменных – *указатель (ссылка)*.
- **Указатель** – переменная, в качестве значения хранящая АДРЕС первого байта памяти, где размещены данные.
- Адрес занимает 4 байта и хранится в виде 2-х слов (каждое слово = 2 байта или 16 бит), первое из которых определяет **сегмент**, второе – **смещение**.
При этом сами данные могут занимать в памяти КИЛОбайты.

Виды указателей

- Типизированные
*содержат адрес, по которому записана переменная
заранее известного типа*
- Нетипизированные
указатели, не связанные с каким-либо типом данных

Объявление в программе

- Типизированные указатели
 - type `Pint` = `^integer`;
 - var `A` : `Pint`;
 - var `A` : `^integer`;
- Нетипизированные указатели;
 - type `Pptr` = `pointer`;
 - var `A` : `Pptr`;
 - var `A` : `pointer`;

Операции с указателями

Разрешены:

- присваивание
- проверка на равенство или неравенство
 - 2 указателя равны только тогда, когда ссылаются на одну и ту же переменную

Запрещены:

- арифметические операции
- сравнение больше-меньше

Правила работы с указателями

- Связать указатель с памятью, хранящей значение существующей переменной можно при помощи функции «@»

```
var P:^integer;  
    A: integer;  
begin  
    A:=5;  
    P:=@A;
```
- Изменить значение переменной через указатель можно приписывая ему окончание «^»

```
P^:=P^+2;
```
- Любые действия над указателем в программе располагаются между процедурами **New** (или функцией @) и **Dispose**.

Порядок использования указателя

- Выделение памяти под динамическую переменную (процедура **NEW**).
- Инициализация указателя.
- Освобождение памяти после использования динамической переменной (процедура **DISPOSE**).

Присваивание значений указателю

- Процедура **New** отводит блок памяти в области динамических переменных и сохраняет адрес этой области в указателе.
- Специальная операция **@** ориентирует переменную-указатель на область памяти, содержащую значение существующей переменной программы.
- Существует единственная константа ссылочного типа **NIL**, которая обозначает «пустой» адрес. Её можно присваивать любому указателю.
- Переменной-указателю можно присвоить значение другого указателя того же типа. Используя тип *pointer* как промежуточный, можно присвоить значение одного указателя другому **при несовпадении их типов**.

Пример `Var p1, p2: ^integer; p3: ^real; pp: pointer;`

`{! неверно} p1:= p3;`

`{! корректно} pp:= p3; p1:= pp; {а также} p1:= p2;`

Пример №1

Условие задачи

*Целое число, заданное с клавиатуры сохранить в **переменной**, адрес которой связать с указателем.*

Увеличить значение переменной в три раза, пользуясь указателем, и вывести на экран.

Особенность решения

Указатель будет связан с ячейкой **статической** памяти,

поэтому **НЕ ТРЕБУЕТСЯ** выделять для него динамическую память.

Программа решения

```
Program DemoPointer1;
Var p:^integer;
    a: integer;
BEGIN
    write('Задайте число ->');
    readln (A);
    p := @A;
    p^:= 3*p^;
    writeln('Число A=', A);
    readln
END.
```

Пример №2

Условие задачи

Целое число, заданное с клавиатуры, сохранить в **динамической** памяти, связанной с указателем. Увеличить значение в пять раз, используя указатель, и передать его в целую переменную. Вывести на экран ее значение, уменьшенное на 10.

Особенность решения

указатель будет связан с ячейкой **динамической** памяти, поэтому её **ТРЕБУЕТСЯ выделить** в начале и **освободить** в конце работы программы

Программа решения

```
Program DemoPointer2;
Var p:^integer;
    a: integer;
BEGIN
  New(p);
  write('Задайте число ->');
  readln (p^);
  p^:= 5*p^;
  A: = p^ ; A := A -10;
  Dispose(p);
  writeln('Число A=', A); readln
END.
```