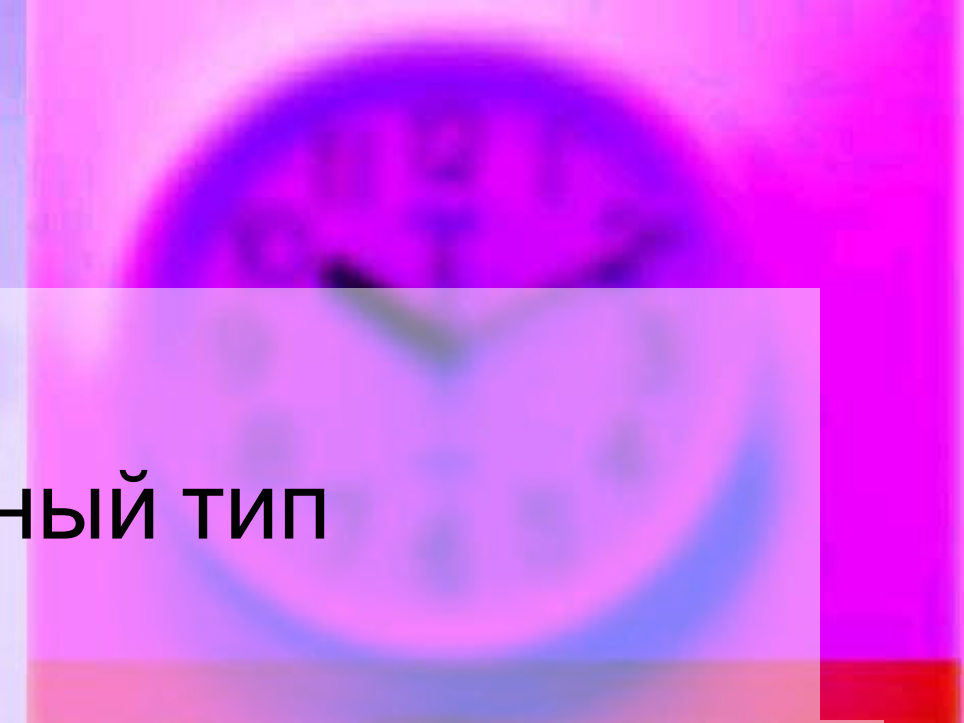
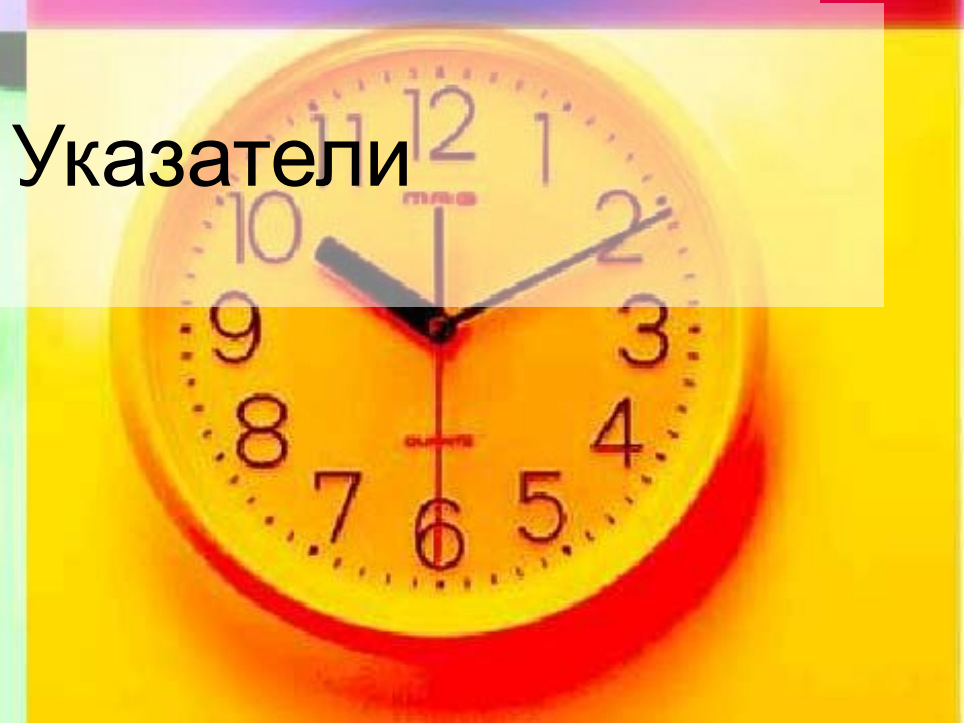
Адресный тип

Указатели



Если компилятор отводит сегмент данных, переменные статические.

При обозначение области данных не именем переменной, а указанием ее адреса - Переменные - динамические.

Указатель – это переменная, которая в качестве своего значения содержит адрес байта памяти
(сегмент в адресе кратен 16 + смещение)

Указатели делятся на типизированные и нетипизированные

Var <имя переменной>: <указатель>;

Для объявления типизированного указателя

```
Var        <имя переменной>: ^ <тип>;
Var
  u1, u2: ^integer;     {типизир.указатель на целое}
  r: ^real;         {типизир.указатель на вещ.}
```

Нетипизированный

```
Var
p: pointer;          {нетипизированный указатель
```

объявляет переменные, значением которых будет адрес (не может быть явно выведен на экран или печать) }

Куча Heap - хип

Память в куче под любую динамически размещаемую переменную **выделяется** процедурой

NEW (<типизированный_указатель>);

Пример:

```
Var
    U1, U2: ^integer;     {типизир.указатели}
    R: ^real;             {типизир.указатель}
    p: pointer;           {нетипизир.указатель}
BEGIN
  new(u1);    {выделяется 2 байта памяти,
                 указатель смещается на 2 байта}
  new(r);     {выделяется в памяти 6 байт,
            указатель смещается на 6 байт (тип REAL) }
```

Для нетипизированного указателя другой способ резервирования

Операции присваивания

Пример:

```
new(u2);
U1:=U2 ;      {запрещено U1:=R, R:=U2(разные типы)}
p:=U1;
U2:=p;
```

сравнения на = и < >




Чтобы по указанному адресу разместить значение, за именем указателя ставится ^ (без пробела).

u1^:=20; {в область памяти u1 помещено значение 20}

r^:=2*Pi; {в область памяти r - значение 6.28}

Пример:

r^ := sqr(r^) + u1^ - 27;

значение значение

Но нельзя r:= sqr(r^) + u1^ - 27;

т.к. указателю нельзя присвоить значение вещ - го типа

Недопустимо смешивать адреса (указатели) и значения (данные):

U1^: = U1 + U2^; r^:= sqr(r); нельзя адрес возводить

значение адрес значение в квадрат

r^:=u1; вещ. данным нельзя присваивать значение указателя

Освобождение динамической памяти

DISPOSE (<типизированный указатель>);

Пример:

```
DISPOSE(R);
DISPOSE(U1); {2 оператора вернут 8 байт}.
```

NIL (пустой)

Пример:

```
Const
i:^integer = NIL;    {объявление константы-указатель}
...
Begin   …
if i=NIL then    {проверка указателя: "свободный"}
        NEW(i);                    {резервирование памяти}
...                          {обработка данных}
if i <>NIL then       {проверка указателя: «занят"}
        DISPOSE(i);          {освобождение памяти}
i=NIL;                       {пометка свободным}
```

администратором кучи

Для работы с нетипизированными указателями используют процедуры:

GETMEM(<нетип.указатель>,SIZE); - резервирование памяти,

FREEMEM(<нетип.указатель>,SIZE); - освобождение памяти.

SIZE - размер в байтах требуемой или освобождаемой части кучи.

За одно обращение до 65521 байта

```
Var
  p: pointer;
Begin
  Getmem (p,8);
…
  freemem (p,8);
End.
```

Функция, которая преобразует сегмент и смещение в значение типа указатель

Ptr(Seg,Ofs:Word): Pointer

Функция, которая возвращает адрес заданного объекта

Addr(x): pointer