

# Variabile dinamice

## Tipul *referință*

---

**Material didactic pentru Informatică**  
(În corespondență cu curriculum-ul la Informatică)  
**Clasa a XI-a**

# Obiectivele lecției:

---

- O1** - să cunoască noțiunea de variabilă dinamică;
- O2** - să creeze și să distrugă variabile dinamice;
- O3** - să efectueze operații cu variabile dinamice;
- O4** - să poată să utilizeze tipul de date articol;
- O5** - să fie capabil de a elabora programe, în care să utilizeze tipul de date referință.

# Structuri dinamice de date

---

Programarea clasică structurată are la bază celebra ecuație a lui Niklaus Wirth:

**Program = Structuri de date + Algoritmi**

## MEMORIA CALCULATORULUI

**PROGRAM Test;**  
**VAR *A, B: Integer*;**

**BEGIN**

...

***A := 12;***

***B := -5;***

...

**END.**

*Declarare  
variabile*

*A*



*B*



*Atribuire  
valori*

*A*



*B*



# Variabile statice

---

- Toate variabilele pentru care în momentul creării programului se poate evalua cantitatea de memorie necesară stocării acestora se numesc **variabile statice**;
- Variabilele declarate în program și notate printr-un identificator (nume) se numesc **variabile statice**;
- Variabilele care există pe toată durata de execuție a blocului în care au fost declarate se numesc **variabile statice**.

# Variabile dinamice

---

Variabilele care sunt create și distruse pe parcursul execuției programului se numesc **variabile dinamice**.

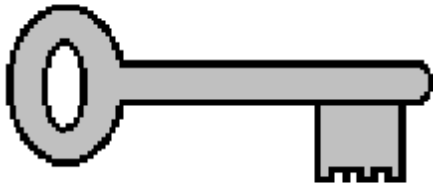
# Variabile dinamice

---

**Ține minte!!!**

1. Variabilele dinamice nu se declară într-o secțiune VAR deci nu se pot identifica prin nume.
2. Nu există pe toată durata de execuție a programului sau activării blocului în care s-au creat.
3. Variabilele dinamice se alocă dinamic într-o zonă specială de memorie, numită memorie HEAP și se distrug la cererea programatorului.
4. Accesul la variabilele dinamice se poate face prin intermediul altor variabile speciale, numite variabile REPER, REFERINȚĂ, INDICATOR, ADRESĂ, POINTER.

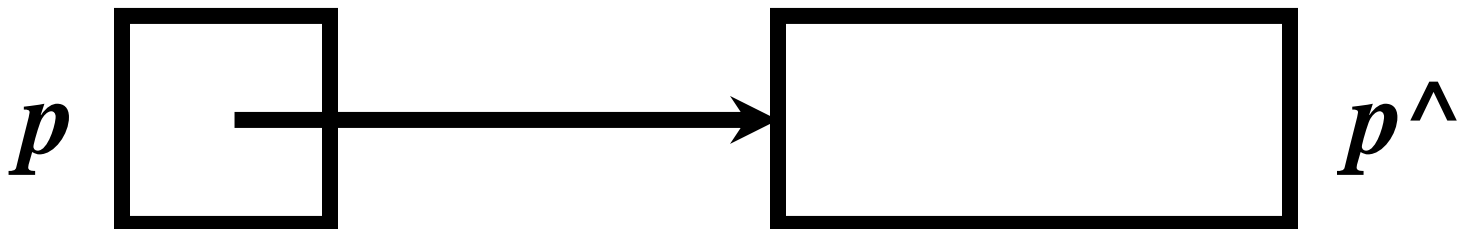
# Variabile dinamice



Variabila REFERINȚĂ



Variabila DINAMICĂ





# Variabila referință

Accesul la variabilele dinamice se face prin intermediul variabilelor de tip referință, care se definește printr-o declarație de forma:

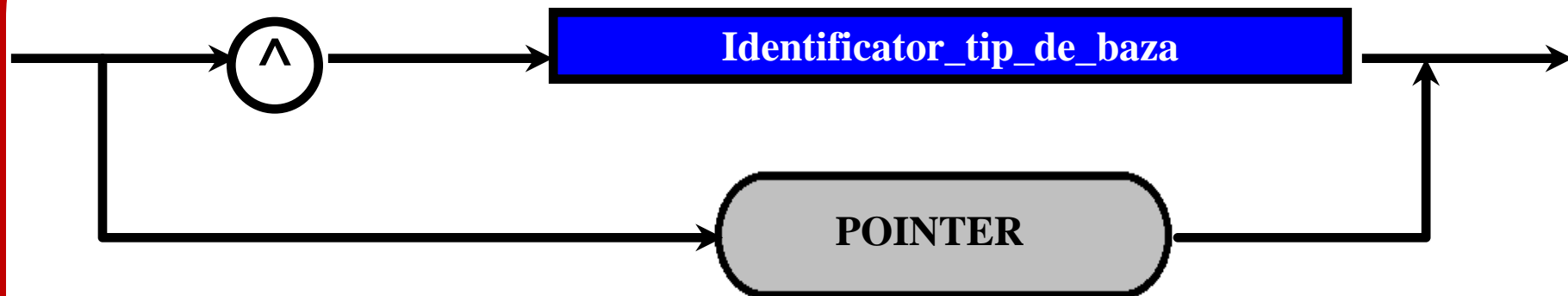
Type  $T_r = \hat{T}_\beta$

numele  
tipului  
referință

adresă

tipul de bază

# Definiri, declarări



```
TYPE Identificator_tip_Referinta = ^ Identificator_tip_de_baza;  
VAR Identificator_Variabila : Identificator_tip_Referinta;
```

```
TYPE Identificator_tip_Referinta = ^ Identificator_tip_Utilizator;  
    Identificator_tip_Utilizator = ... ;  
VAR Identificator_Variabila : Identificator_tip_Referinta;
```

sau

```
VAR Identificator_Variabila : ^ Identificator_tip_Referinta;
```

# Example

---

```
Type AdresaInteger=^integer;  
      AdresaChar= ^char;  
var i: AdresaInteger;  
    r: ^real;  
    c: AdresaChar;
```

Variabilele **i**, **r**, **c** sunt *variabile referință*, sau *variabile reper*. Pentru ele se alocă la compilare un spațiu de memorie de **4 bytes** (conține *adresa de memorie a variabilei dinamice referite*).

# Example

## TYPE

```
SubDomeniu=1..25;  
RefSub=^SubDomeniu;  
Enumerare=(Rosu, Galben, Verde);  
RefEnum=^Enumerare;  
Vector=Array[1..10] of Real;  
RefVec=^Vector;  
  
RefData=^DataNast;  
DataNast=RECORD  
    Zi : 1..31;  
    Luna : 1..12;  
    An : 1900..2050;  
end;  
  
RefXXX=POINTER;
```

## VAR

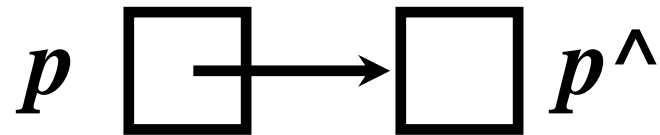
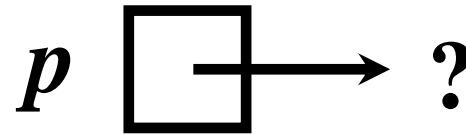
```
p : RefSub;  
p1 : RefEnum;  
p2 : RefVec;  
p3 : RefData;  
a : ^Integer;  
r : ^Real;  
ch : ^Char;
```

Variabilele **p**, **p1**, **p2**, **p3**, **a**, **r**,  
**ch** sunt *variabile referință*,  
sau *variabile reper*.

# Valorile variabilelor referință

Variabilele referință (reper) pot să primească trei tipuri de valori (să se afle în trei stări):

- să fie neinițializate, nedefinite, nedeterminate (n-au nici o valoare);
- să conțină referința (adresa) la o variabilă dinamică;
- să păstreze valoarea NIL (o constantă predefinită Pascal);



# Operații posibile cu variabile reper

## 1. ATRIBUIREA

```
TYPE IdTip = ...;  
VAR p, q : ^ IdTip;  
BEGIN
```

...

*p* := *q*;

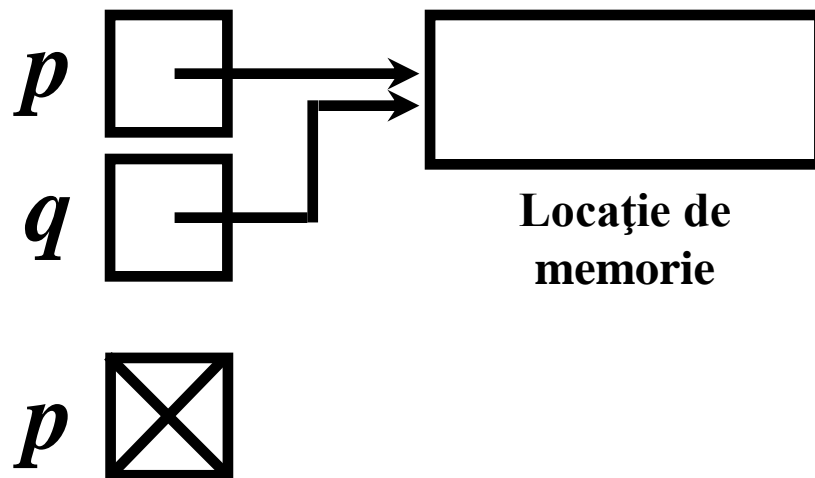
...

*p* := NIL;

...

```
END.
```

*Variabila referință poate primi valoarea unei alte variabile sau funcție de același tip referință cu ea, sau poate fi inițializată cu constanta NIL.*



# Operații posibile cu variabile reper

## 2. COMPARATIA

```
VAR R, RR : ^ Real;  
    Ri, Rc : ^ Integer;  
    logic : Boolean;
```

```
BEGIN
```

```
...
```

```
    logic := Ri <> NIL;
```

```
...
```

```
    If (R=RR) OR (logic) Then ... ;
```

```
...
```

```
END.
```

*Variabilele referință pot să apară în expresii relaționale, singurii operatori relaționali admiși fiind “=” și “<>”.*

## 3. PARAMETRI

*Variabilele referință pot fi parametri ai unor proceduri sau funcții.*

# Variabile dinamice: CREARE

Variabilele dinamice sunt variabilele de un tip oarecare simplu sau structurat pentru care se poate alocă memorie numai în faza de execuție a programului în funcție de necesitate.

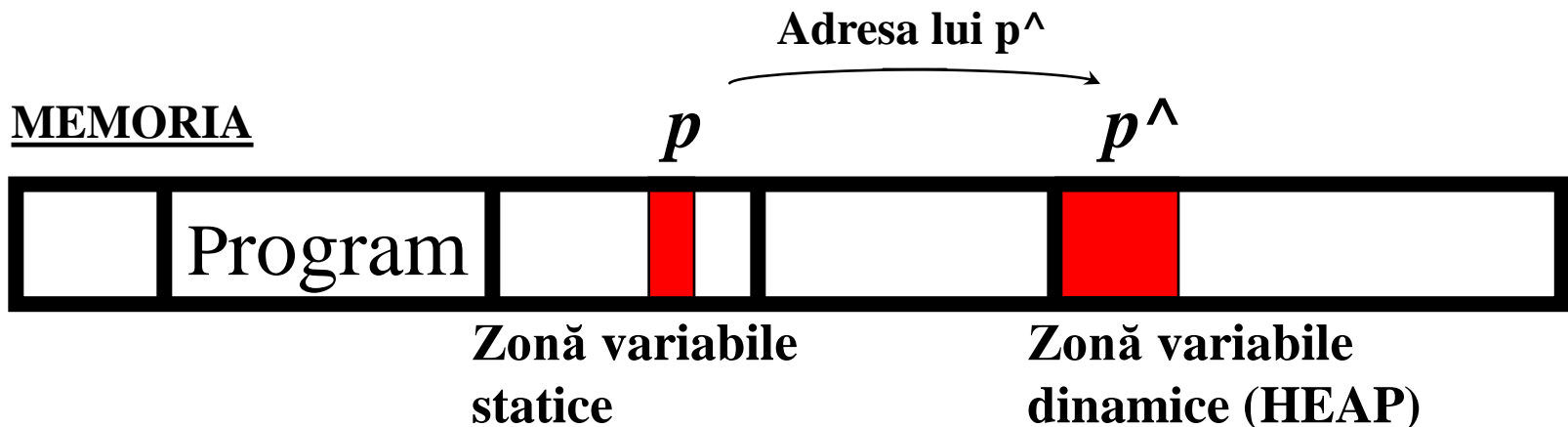
Type RefTip = ^Real;

Var p: RefTip;

Begin

**NEW(p);**  
...

End.



M a r i a G u t u



# Variabile dinamice: DISTRUGERE

Eliberarea zonelor de memorie, ocupată de variabilele dinamice, create prin execuția procedurii NEW, pentru a nu deveni “gunoi” se realizează prin execuția procedurii predefinite DISPOSE.

Type *RefTip* = ^ *Real*;

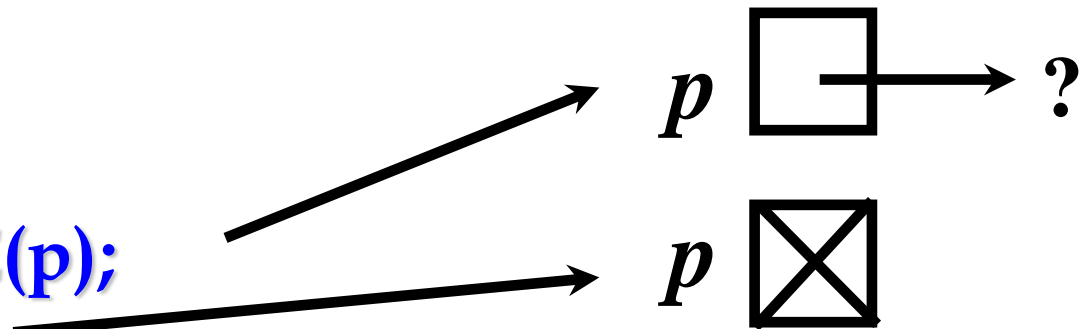
Var *p*: RefTip;

Begin

...  
**DISPOSE(p);**

**p:=NIL;**

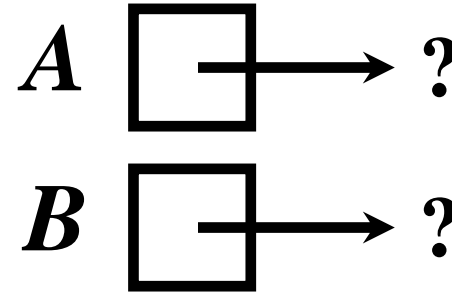
End.



# Operații cu variabile dinamice

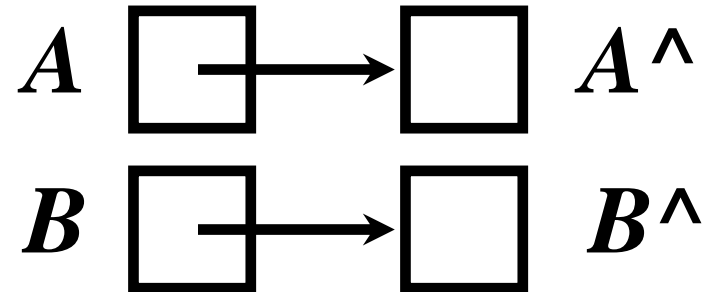
## 1. Declaraare

**VAR A, B: ^Integer;**



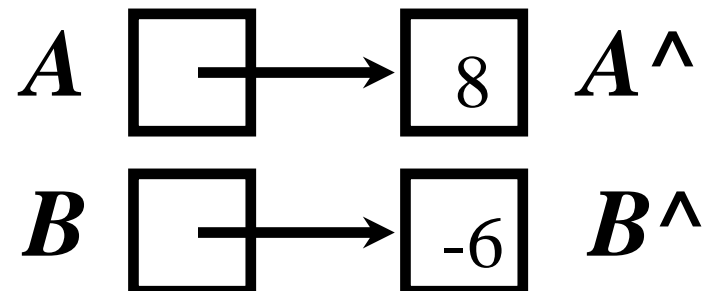
## 2. Creare

**NEW (A);**  
**NEW (B);**



## 3. Atribuire de valori

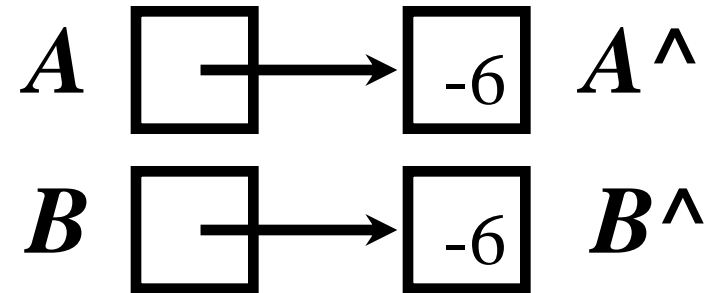
**$A^ := 8;$**   
 **$B^ := -6;$**



# Operații cu variabile dinamice

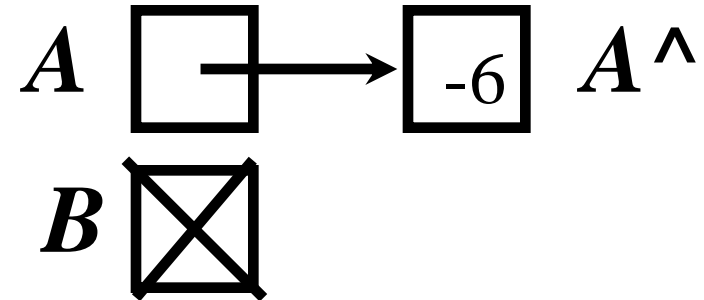
## 4. Copiere valori

$A^{\wedge} := B^{\wedge};$



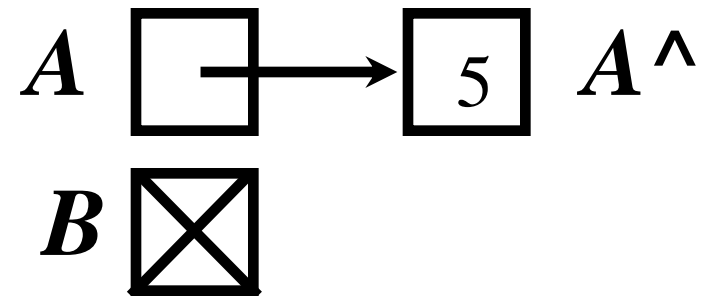
## 5. Atribuire NIL

$B := \text{NIL};$



## 6. Atribuire de valori

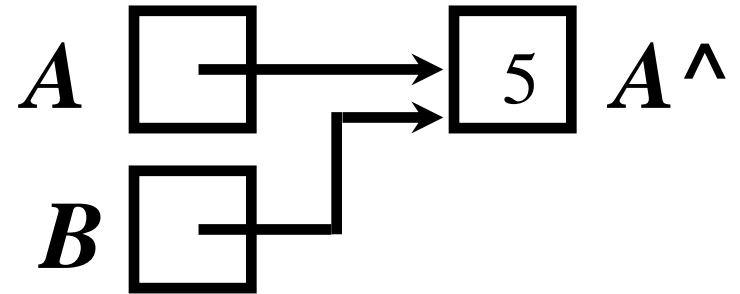
$A^{\wedge} := \text{Round}(3.14)+2;$



# Operații cu variabile dinamice

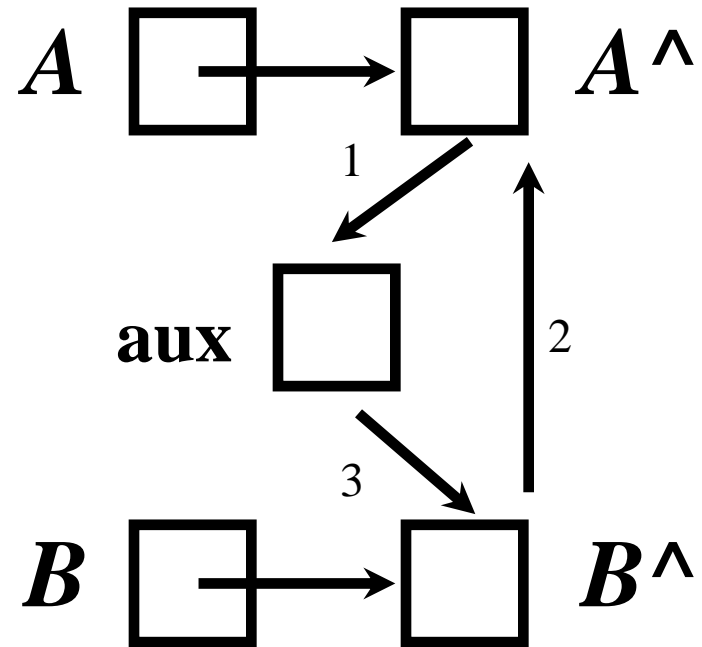
## 7. Schimb adrese

**B := A;**



## 8. Schimb valori

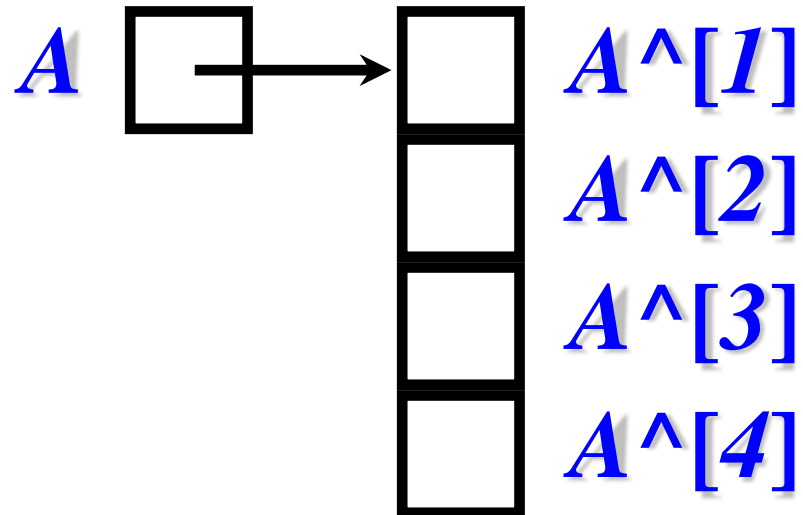
```
Var aux:Real;  
    A,B: ^Real;  
Begin  
    NEW(A); NEW(B); ...  
    {1} aux:=A^;  
    {2} A^:=B^;  
    {3} B^:=aux; ...  
End.
```



# Operații cu variabile dinamice

## 9. Creare variabile dinamice structurate

```
TYPE Vector=Array[1..4] of Integer;  
Var A:^Vector;  
Begin  
    ...  
    NEW(A);  
    ...  
End.
```



Acces la o componentă în caz general:  $A^{\wedge}[i]$

# Operații cu variabile dinamice

## 10. Creare variabile dinamice structurate

```
TYPE Complex=RECORD
```

```
    Re, Im : Real;
```

```
end;
```

```
VAR C : ^Complex;
```

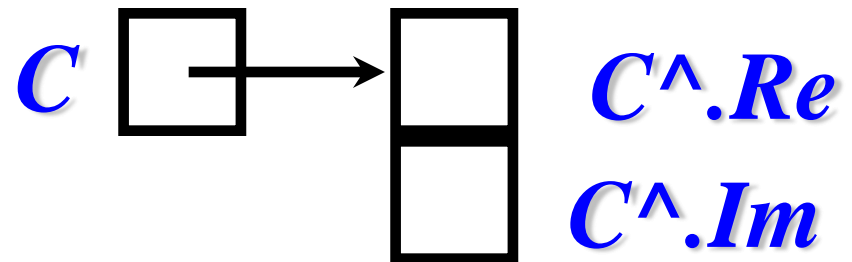
```
Begin
```

```
    ..
```

```
    NEW(C);
```

```
    ..
```

```
End.
```

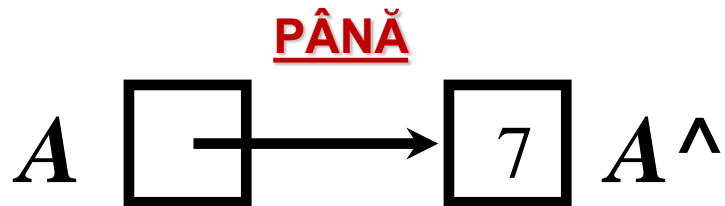


Acces la o componentă în caz general:

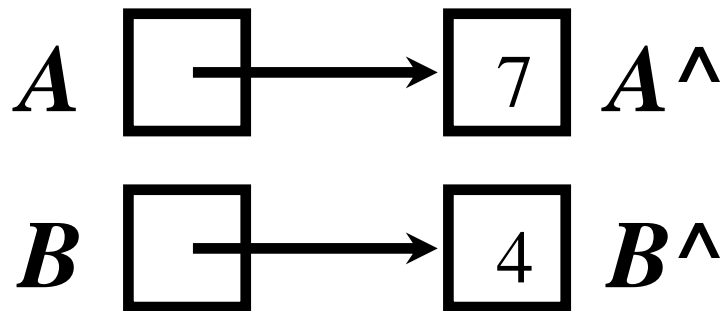
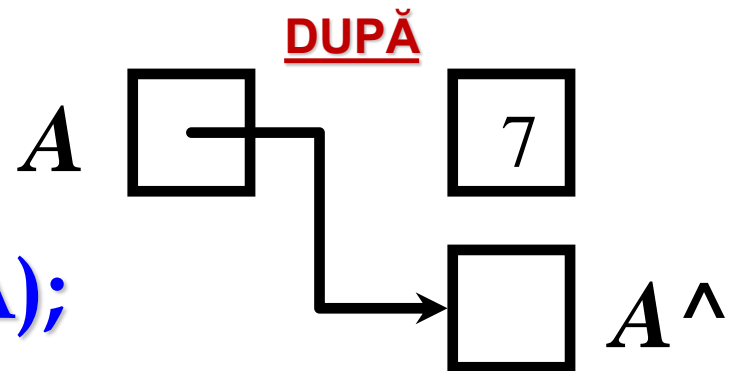
$Id\_var^.c\acute{a}mp$

# ERORI FRECVENTE

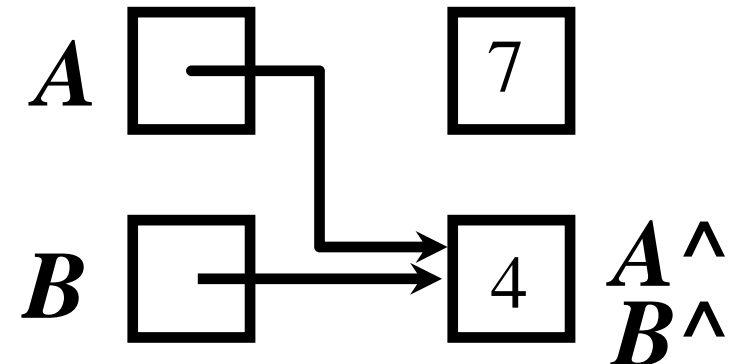
**NODURI INACCESIBILE** – Structuri dinamice cu referințe la ele pierdute (lădița încuiată cu cheia pierdută) – apar după utilizarea procedurii NEW sau instrucțiunii de atribuire.



NEW(A);



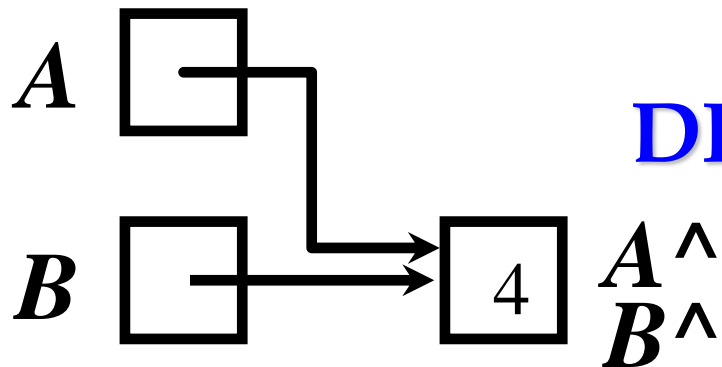
A := B;



# ERORI FRECVENTE

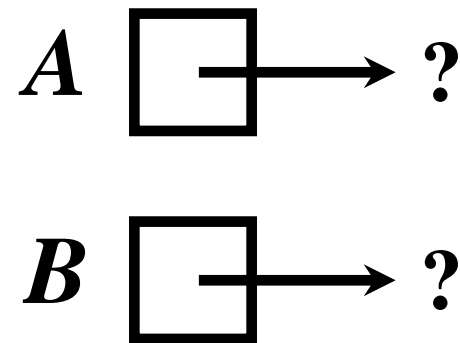
**REFERINȚE SUSPENDATE** – Indicatori ce conțin referințe la locații de memorie eliberate cu DISPOSE.

PÂNĂ



`DISPOSE(A);`

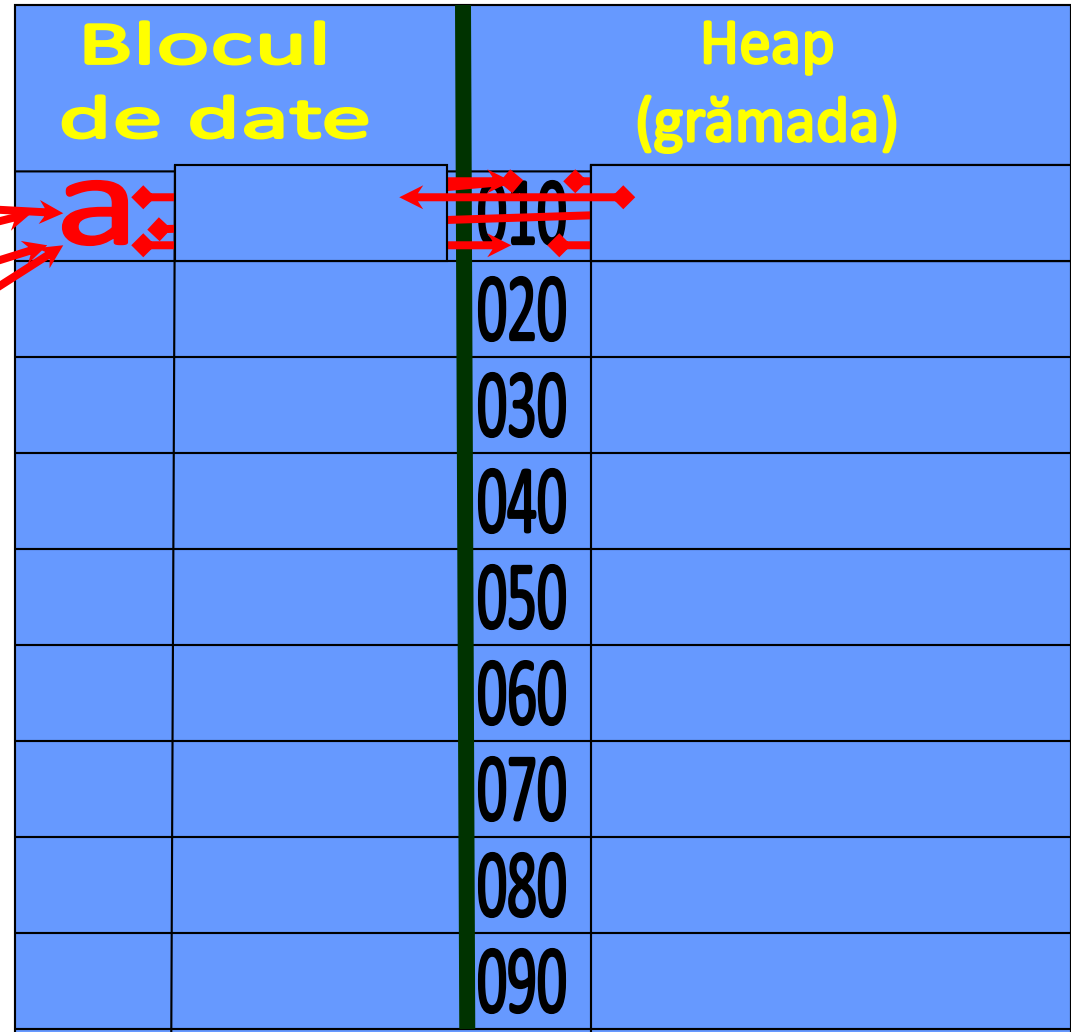
DUPĂ





# Interacțiunea program-memorie

```
Program p1;  
var a : ^integer;  
begin  
  new (a);  
  a^:=23;  
  dispose(a);  
end.
```



# Interacțiunea program-memorie

```
Program p1;  
var a,b : ^integer;  
begin  
  new (a);  
  new (b);  
  a^:=23;  
  b^:=a^+17;  
  a^:=b^+a^;  
  dispose(a);  
  dispose(b);  
end.
```

Blocul de date			Heap (grămadă)
a		010	
b		020	
		030	
		040	
		050	
		060	
		070	
		080	
		090	

# Exemplu

Program Exemplu;

{Operatii cu variabile dinamice}

Type AdresaInteger=^integer;

Var i,j,k:AdresaInteger;

    r, s, t: ^real;

Begin

{crearea variabilelor dinamice de tip integer}

new(i); new(j); new(k);

{Operatii cu variabilele create}

i ^:=1; j ^:=2; k ^:=i ^+j ^;

Writeln(k ^);

{Crearea variabilelor dinamice de tip real}

new(r); new(s); new(t);

{Operarii cu variabilele create}

r ^:=1.0; s ^:=2.0; t ^:=r ^/s ^;

Writeln(t ^);

{Distrugerea variabilelor  
dinamice}

dispose(i);

dispose(j);

dispose(k);

dispose(r);

dispose(s);

dispose(t);

readln;

end.

# Ce va afișa acest program?

```
Program test1;  
  var m,n,k: ^integer;  
      c: ^char; x,y,z: ^real; a:integer;  
begin  
  new(m); new(n); m^:=9; n^:=8; k:=m;  
  k^:=10;  
  writeln(m^,' ',n^,' ',k^);  
  k:=n; n^:=15; m^:=k^;  
  writeln(m^,' ',n^,' ',k^);  
  {new(a)}; a:=6; n^:=a;  
  writeln(m^,' ',n^,' ',k^);  
  new(c); new(x); new(z); x^:=6.3;  
  c^:='y'; {x^:=c^}; y:=z; z^:=8.9;  
  writeln(x^,' ',y^,' ',z^,' ',c^);  
  readln;  
end.
```

# Ce va afișa acest program?

```
Program test2;  
  var a,b,c : ^integer;  
      cl: ^char; x,y,z: ^real; m:integer;  
  
begin  
  new(a); new(b); a^:=3; b^:=8; c:=b; c^:=20;  
  writeln(a^,' ',b^,' ',c^);  
  c:=a; b^:=15; m:=c^;  
  writeln(a^,' ',b^,' ',c^,' ',m);  
  
  new(x); new(z); new(cl); x^:=6.3;  
  cl^:='y'; z^:=x^; y:=x; y^:=18.4;  
  writeln(x^,' ',y^,' ',z^,' ',cl^);  
  readln;  
  
end.
```

# Ce va afișa acest program?

```
Program test3;  
  var a,b,c : ^integer;  
      x,y,z: ^real;  m:longint;  
begin  
  new(a);  new(b);  a^ :=3;  b^ :=8; c:=b;  
  c^ :=20;  
  writeln(a^,' ',b^,' ',c^);  
  c := a; a^:=15;  m:= c^+5;  
  writeln(a^,' ',b^,' ',c^,' ',m);  
  new(x);  new(z);  x^:=6.5;  
  z^:=x^; y:=x;  y^ :=18 ;  
  writeln(x^:4:0,' ',y^:4:0 , ' ',z^:4:1);  
  readln;  
end.
```

# Concluzie

---

Alocarea dinamică a memoriei necesită o atenție sporită din partea programatorului care este obligat să asigure crearea, distrugerea și referirea corectă a variabilelor dinamice.

# Extindere

---

1. 2.1. de studiat și conspectat.
2. De analizat PPT din GitHub.
3. Ex. 7 - 10 (pag. 33).

<https://www.slideshare.net/mariamarinela756/variabile-dinamice-tipul-de-date-referin>