Рачунарство и информатика

Јелица Васиљевић

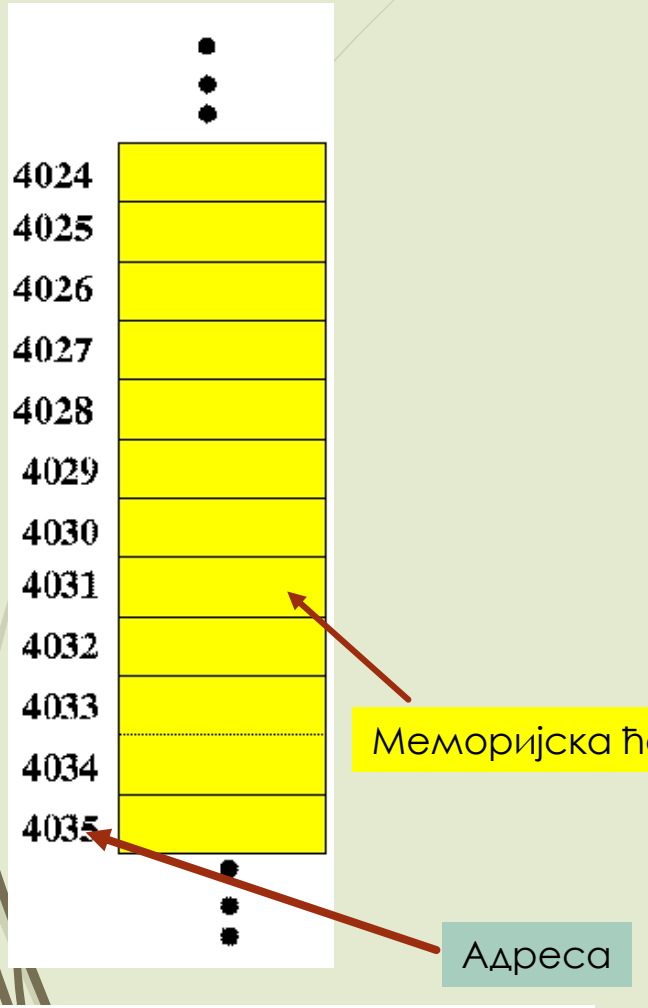
Одељење II см

Школска 2015/2016

Динамичке структуре података

- Декларисање променљивих
  - Заузима се меморијски простор одређене величине
  - Унпред познато колико променљивих је потребно
  - Широка класа проблема је решива
- Проблем
  - Број објеката није унапред познат
  - Број објеката може знатно да варира
    - Да ли резервисати простор за најгори случај (уколико је познат)
  - Шта се дешава ако нема довољно **резервисаног** простора ?

Динамичке структуре података

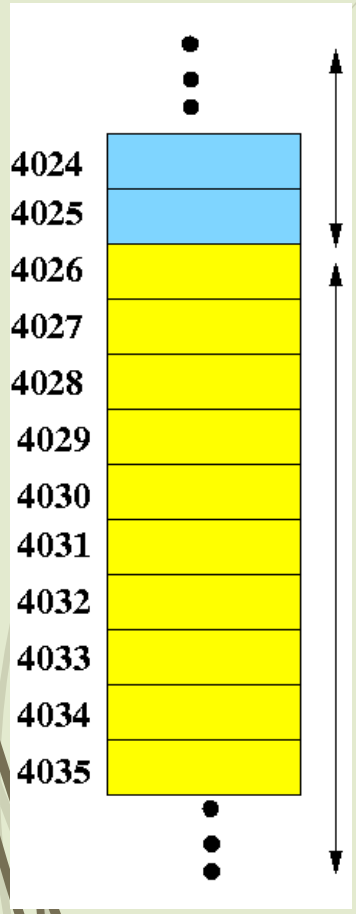


Меморија рачунара

- Решење – **динамичке променљиве**
  - **Креирају** се и **уништавају** у току извршавања програма
  - Приступа им се САМО помоћу **ПОКАЗИВАЧКИХ ПРОМЕНЉИВИХ (ПОКАЗИВАЧА)**

- Меморија рачунара може се представити као низ нумерисаних (адресираних) меморијских ћелија.
  - Најмања меморијска ћелија којој се може приступити је **бајт**
    - Меморијским ћелијама може се приступати појединачно или групно.
      - Тип података : integer - 2 (или 4) бајта, real – 4 (или 8) бајта, char – 1 бајт, итд.
        (наведене вредности зависе од компојлера)

Динамичке структуре података

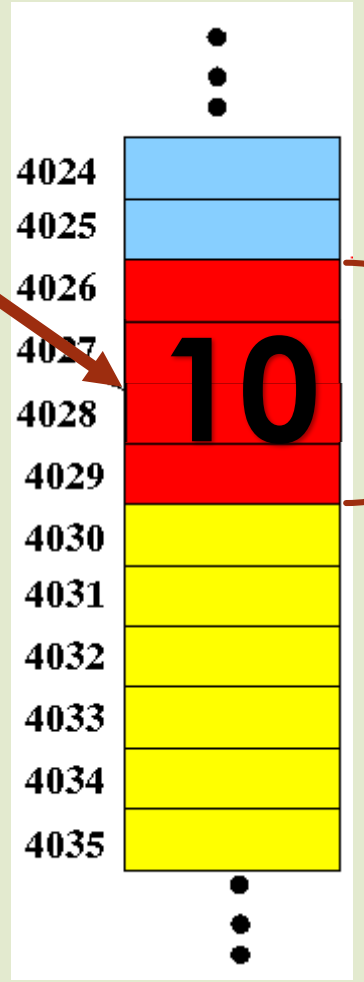


Већ се користи

Слободно

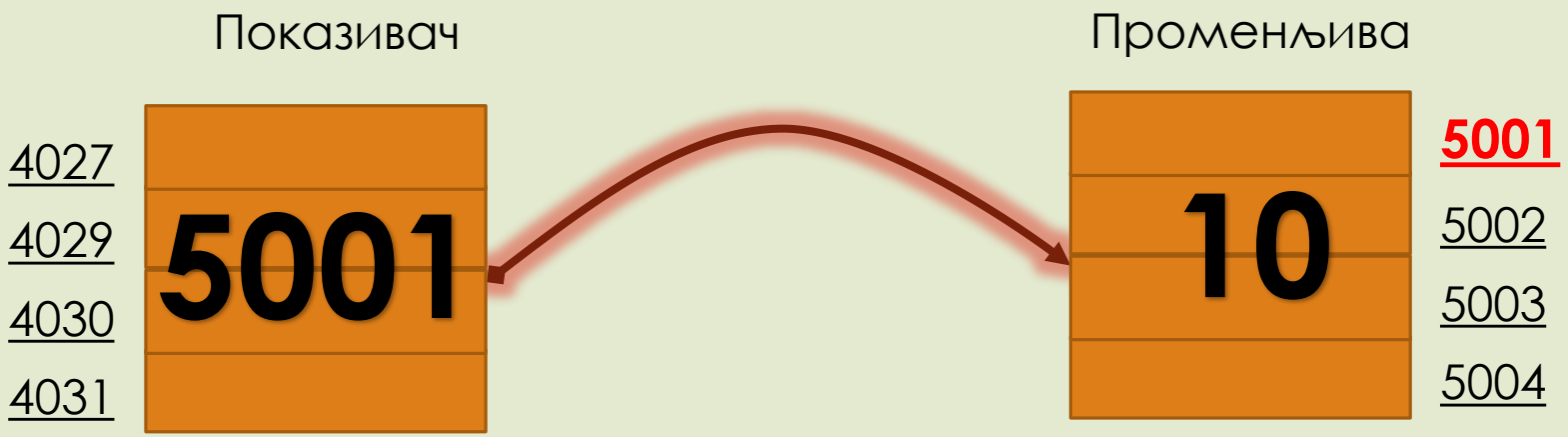
```
var
   x:integer;

   ...
   ...

   x:= 10;
```

Целобројни тип заузима 4 бајта.
Променљива се смешта на слободну локацију



Динамичке структуре података - показивачи

- Показивачи су **ПРОМЕНЉИВЕ** које ***показују*** на неку другу променљиву
  - **показивање -** садржи **АДРЕСУ меморијске локације** на којој се чува нека друга променљива
  - Показивачи су променљиве => заузимају меморију
    - 32-битне платформе – 4 бајта (ћелије) (4 * 8 = 32)
    - 64-битне платформе – 8 бајтова (8 * 8 = 64)



Динамичке структуре података - показивачи

- Показивачи су **типизирани**
  - Тип се односи на тип податка на који показује
  - Битно јер различити типови података имају различите меморијске захтеве
- Показиваче је потребно декларисати у секцији ***var*** ***(јер су промењиве!)***
- Декларација

```
var
        pokazivac : ^ tip_podatka;
```

- Знак ^ означава се да је променљива показивач
- `tip_podatka` означава тип податка на који показивач може да указује
- Када је једном декларисан показивач на одређени тип, тај показивач може показивати **САМО** на ***промењиве тог типа***

Динамичке структуре података - показивачи

- За приступање променљивој на коју показује показивач користи се ^
  - Када се ^ налази испред назива типа - декларише се показивач
  - Када се ^ налази испред **показивача** – приступа се променљивој на коју показује показивач

```
var
pokazivac : ^ integer;
```

pokazivac^: "шта пише на меморијској локацији на коју показивач показује"

Динамичке структуре података - показивачи

- Операције над променљивима **типа показивач**
  - Операција доделе :=
    - Додела адресе променљиве на коју показује
  - Операција поређења =
    - Да ли два показивача показују на **исту** меморијску локацију (да ли су исте адресе)
- *Није могуће учитавати нити исписивати вредности показивачке променљиве*

Динамичке структуре података – динамичке промењљиве

- Динамичке промењљиве
  - Креирају се у току извршавања програма
  - Уништавају се на крају програма (или када више нису потребне)
- Не декларишу се у секцији ***var***
  - **Немају име**
  - Како им приступити ?
    - Преко **показивача** који показују на њих
- Показивач садржи **АДРЕСУ**
  - У показивач сместити адресу динамичке промењљиве
  - Даљи рад са динамичком промењљивом је рад са показивачем на њу.

Динамичке структуре података – динамичке променљиве

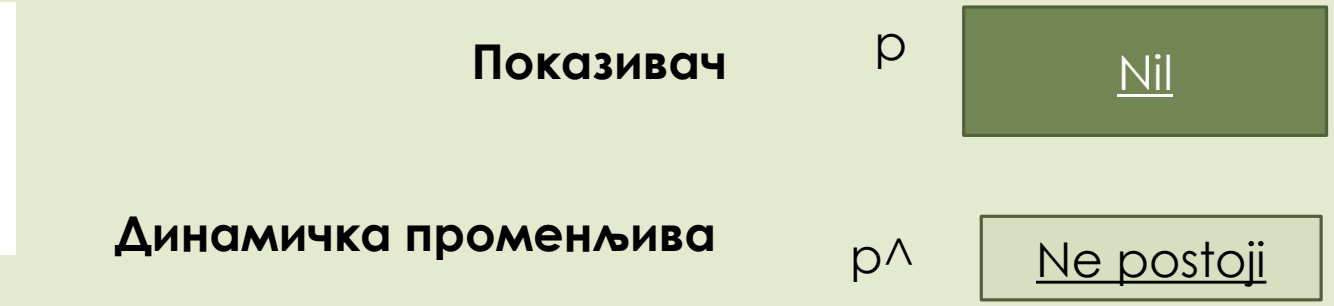
- Креирање динамичке променњиве
  - За креирање **динамичке променљиве** користи се процедура **NEW**
    - Прихвата ЈЕДАН аргумент и то показивач у који ће сместити адресу креиране динамичке променљиве
    - Обезбеђује (алоцира) меморијски простор чија величина дговара **ТИПУ** податка на који показује показивачка променљива.
    - Показивачкој променљивој додељује адресу креиране динамичке променљиве тј. почетну адресу алоцираног меморијског простора

```
var
p : ^ integer;
```

Динамичке структуре података – динамичке промењљиве

- Показивач на почетку програма има **недефинисану** вредност
  - Може садржати било шта
  - Пошто су вредности показивачке промењљиве адреса, показивач показује на неку локацију у меморији која може бити било где
- Показивач који нема додељену вредност треба да показује **ни на шта**
  - Показивач показује ни на шта ако има вредност **NIL**
    - **Nil** је константа и резервисана реч

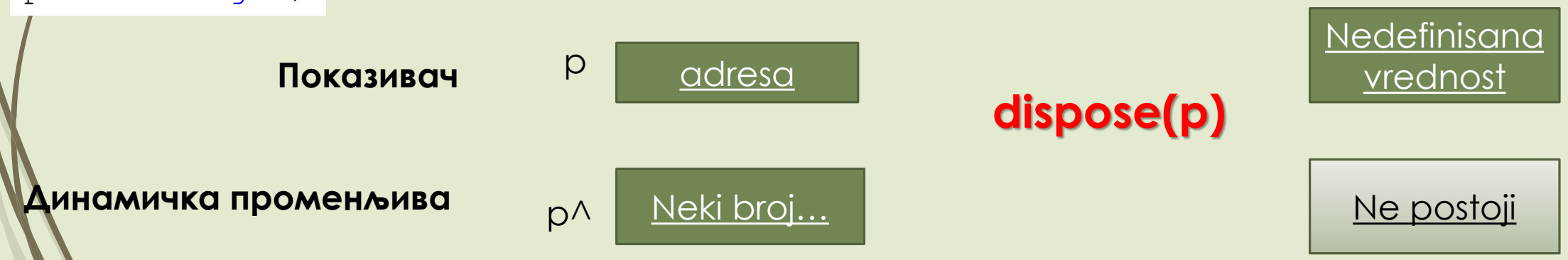
```
var
p : ^ integer;
…
p:=nil;
```



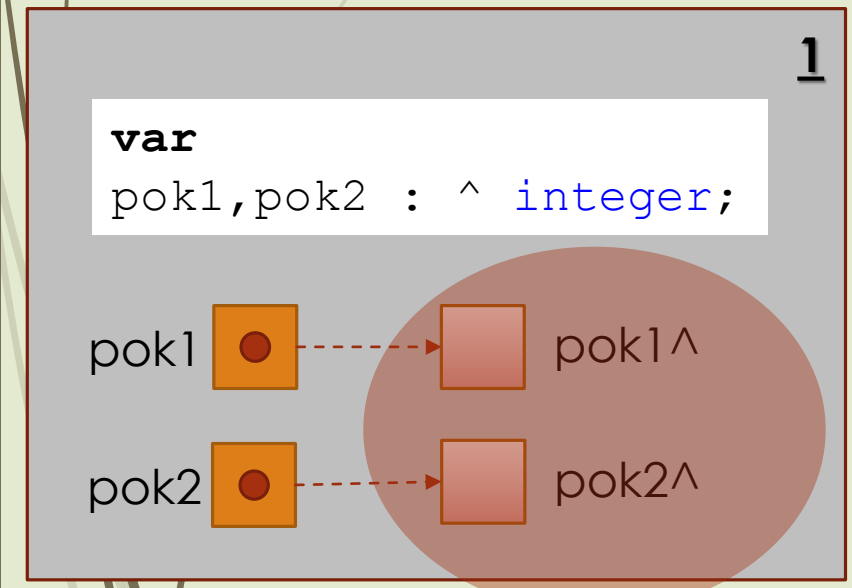
Динамичке структуре података – динамичке променљиве

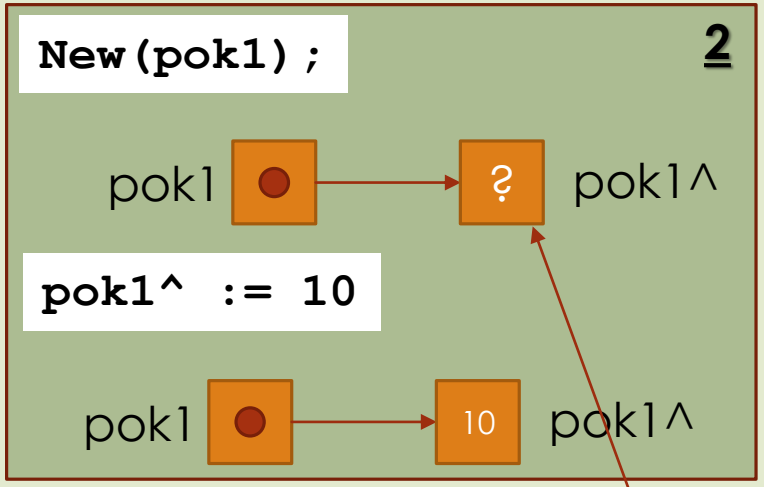
- Уклањање динамичких променљивих
  - Када креиране динамичке проименљиве више нису потребне, треба их уклонити и ослободити меморијски простор које су заузеле
  - Уклањање динамичких променљивих врши се процедуром **DISPOSE**
    - Прима један улазни параметар и то показивач који садржи адресу динамичке променљиве
    - Ослобпођа простор који је заузела та динамичка променљива
    - Показивач добија недефинисану вредност али и даље постоји !

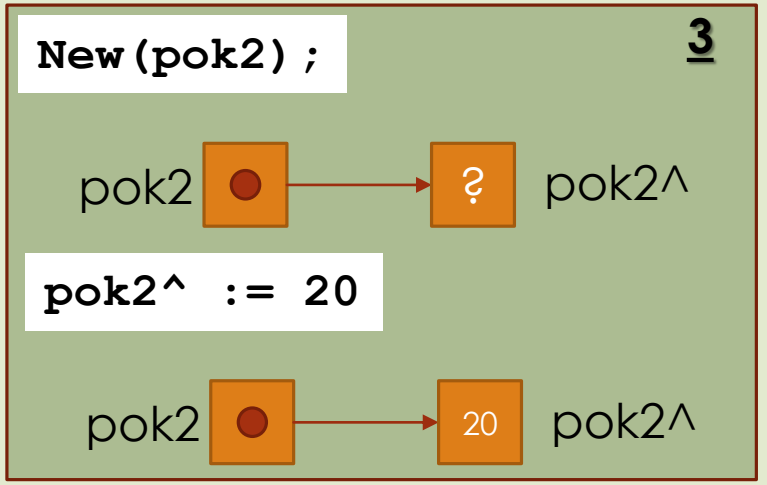
```
var
p : ^ integer;
```



Динамичке структуре података – динамичке променљиве



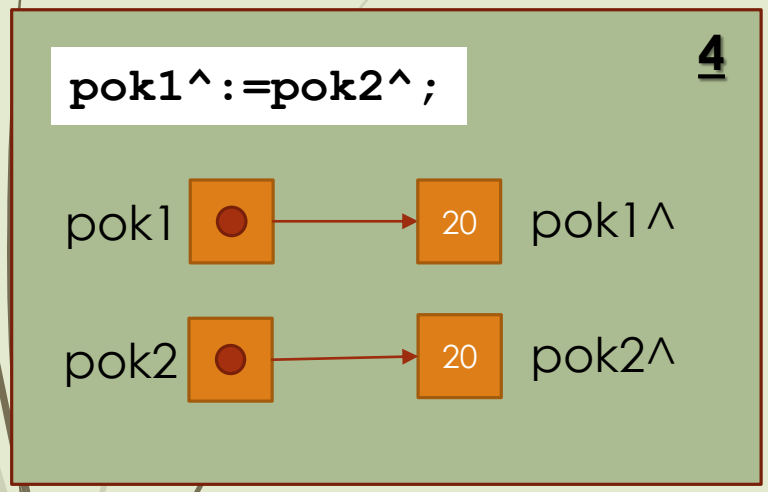


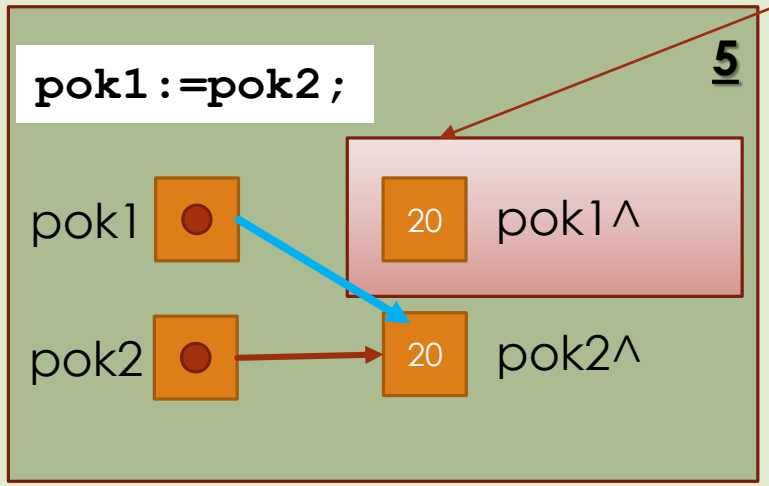


Не постоји

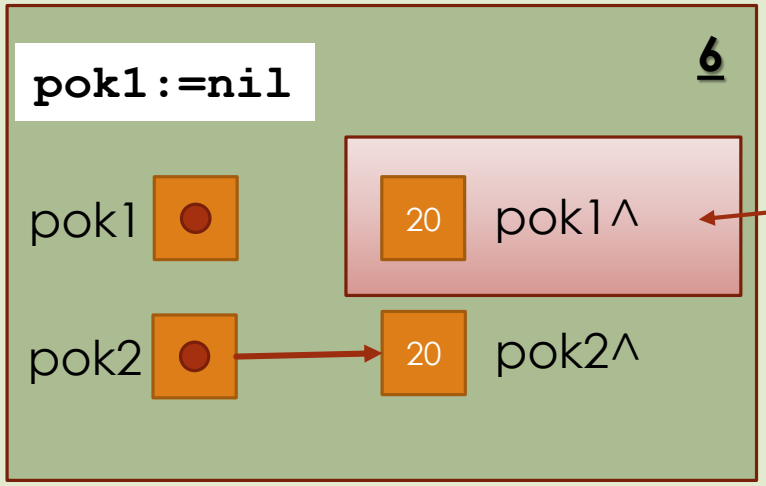
Постоји али има недефинисану вреднсот

Динамичке структуре података – динамичке променљиве





Постоји али постаје недоступно



Остаје недоступно

Динамичке структуре података – добијање адресе промењљиве

- Да би се добила адреса нека промењљиве у меморији, користи се оператор @

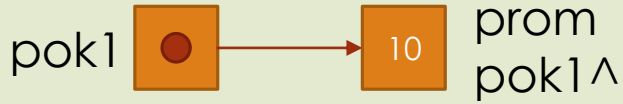
```
pok1:=@promenljiva;
```

  - Промењљива МОРА бити типа на који показивач показује

```
Var
 prom:integer;
 pok: ^integer;
```


```
prom:=10;
```


```
Pok:=@prom;
```



Динамичке структуре података

- Шта је резултат следећег програма ?

```
program prviPrimer;
type
 osoba = record
        ime:string;
        godine:integer;
        end;
pokazivac = ^osoba;

var
   p1,p2:pokazivac;
```

```
Begin

new(p1);
 p1^.ime := 'Petar';
 p1^.godine:= 23;
 writeln(p1^.ime,' ',p1^.godine);

new(p2);

 p2^.ime := 'Jovana';
 p2^.godine:= 17;
 writeln(p2^.ime,' ',p2^.godine);
```

```
p2:=p1;

 p2^.ime:='Igor';
 p2^.godine := 18;
writeln(p1^.ime,' ',p1^.godine);
writeln(p2^.ime,' ',p2^.godine);

 dispose(p1);
 dispose(p2);

 end.
```

Динамичке структуре података

- Резултат је :

```
Petar 23
Jovana 17
Igor 18
Igor 18
```