



Рачунарство и информатика

Јелица Васиљевић

Одељење II см

Школска 2015/2016



Једноструко повезане листе

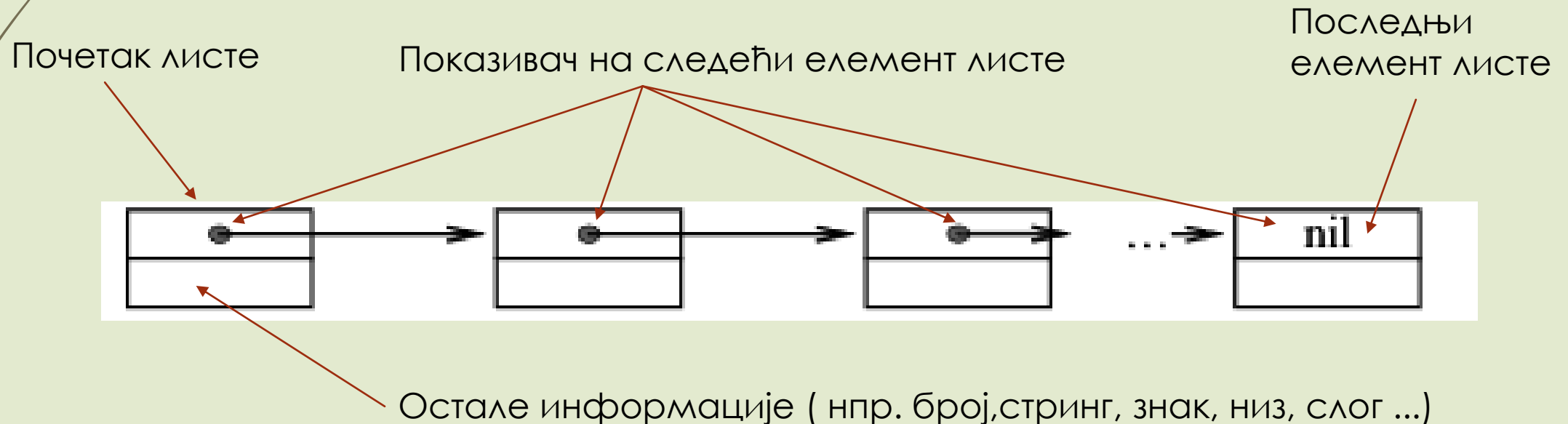
- Низови :
 - За цео низ се резервише меморија
 - Елементи низа су подједјанко удаљени једни од других
- Проблем:
 - Величина низа је позната унапред
 - Брисање/уметање новог члана низа захтева померање одређеног броја постојећих елемената
 - Шта се дешава ако нема довољно меморије на једном месту за цео низ?

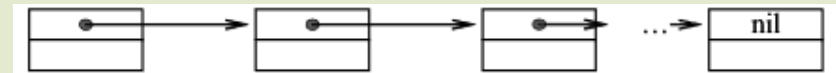
Једнострукко повезане листе

- Решење **ПОВЕЗАНИЕ ЛИСТЕ**
 - Елементи не морају да буду на истом међусобном растојању
 - Није потребно знати укупан број елемената пре почетка рада
 - Убацавање/избацивање елемента из листе не утиче на промену положаја осталих елемената
- Елементи листе се смештају **БИЛО ГДЕ У МЕМОРИЈИ** а затим се **ПОВЕЗУЈУ** и то тако да сваки елемент буде повезан са својим претходником (изузев првог елемента) и/или следбеником (осим последњег елемента)
- Како се повезују ?
 - Преко показивача
 - Сваки елемент, поред осталих информација, чува и информацију о томе где се налази његов следбеник (и/или претходник)
 - Где се налази = адреса следбеника (и/или претходника)

Једноструко повезане листе

- Повезана листа је
 - Скуп елемената који су произвољно распоређени по меморији и који су међусобно повезани
 - Ако је елемент **А ПОВЕЗАН** са елементом **Б** то значи да **А** зна где се **Б** налази у меморији тј. има адресу објекта **Б**





Једноструко повезане листе

- Дефинисање једноструко повезане листе

```
type
  pokazivac = ^slog;
  slog = record
    broj : integer;
    sledeci : pokazivac;
  end;
var
  element_liste : pokazivac;
```

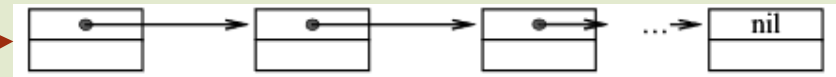


Једнострукко повезане листе

- Да би се приступило листи, позиција првог елемента мора бити позната
 - Свим осталим елементима је могуће приступити полазећи од првог (зашто ?)
 - Пошто се елементи листе динамички креирају, динамички је креиран и први елемент листе
 - Првом елементу листе приступа се преко **ПОКАЗИВАЧА** који садржи његову адресу
 - Тај показивач се најчешће назива **ГЛАВА ЛИСТЕ**.
 - Коришћење показивача за памћење првог елемента листа је згодно и због тога што листа може да буде празна
 - Без показивача – како да знамо да је листа празна ?
 - Са показивачима – вредност показивача је **NIL**

Глава листе





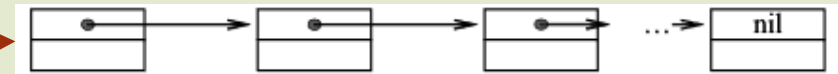
Једноструко повезане листе

- ▶ Глава листе је показивач на први елемент листе
- ▶ Листа је на почетку празна
 - ▶ Глава листе је **NIL**

```
program liste;  
type  
  pokazivac = ^slog;  
  slog = record  
    broj : integer;  
    sledeci : pokazivac;  
  end;  
var  
  element_liste : pokazivac;  
  glava_liste : pokazivac;  
begin  
  glava_liste := nil;
```

glava





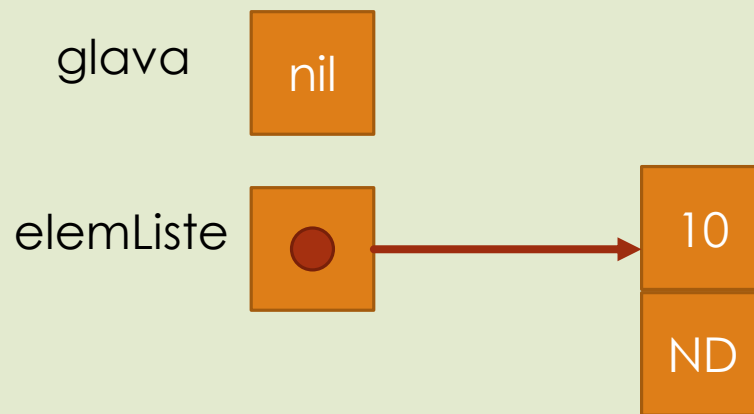
Једнострукко повезане листе

- Додавање елемента на почетак листе
 - Елементи се додају динамички
 - Најпре мора да се резервише меморија за нови елемен – `new(p)`
 - Елемент се повезује са главом листе
 - Глава листе показује на нови елемент

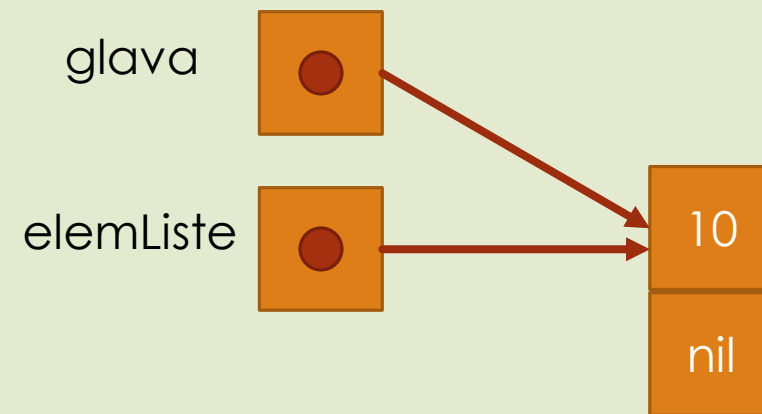
```
glava_liste:=nil;
```

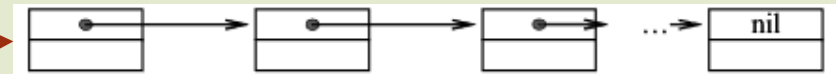


```
new(element_liste);  
element_liste^.broj:=10;
```



```
element_liste^.sledeci=glava_liste;  
glava_liste:=element_liste;
```



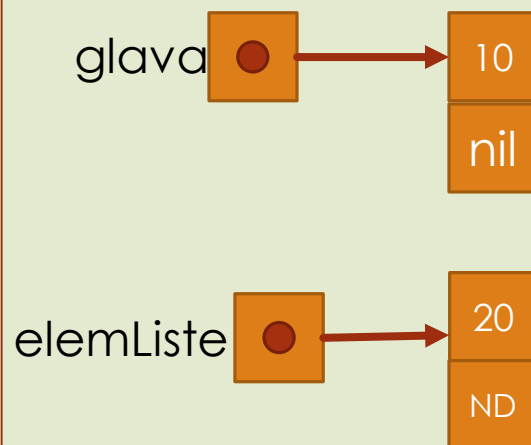
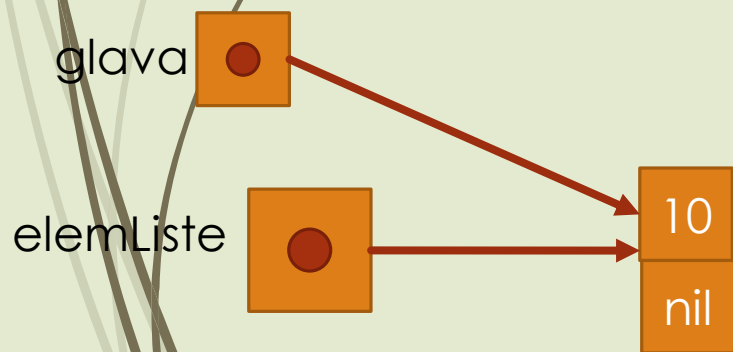


Једнострукко повезане листе

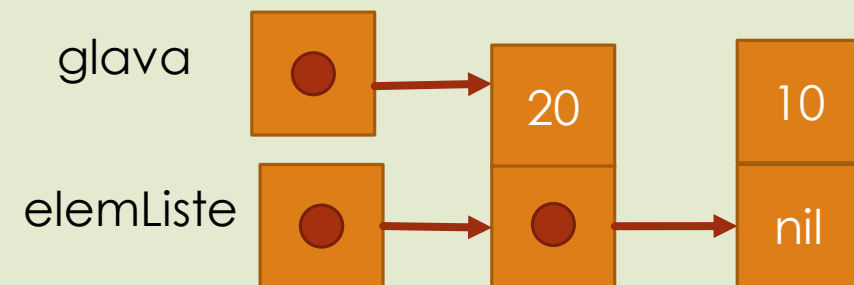
Додавање елемента на почетак листе

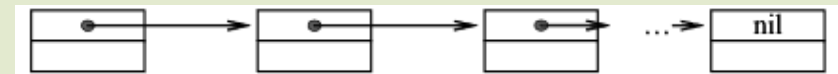
- Елементи се додају динамички
- Најпре мора да се резервише меморија за нови елемен – `new(p)`
- Елемент се повезује са главом листе
- Глава листе показује на нови елемент

```
new(element_liste);  
element_liste^.broj:=20;
```



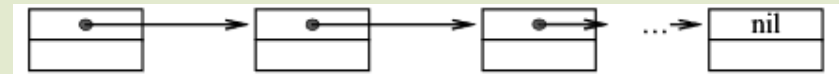
```
element_liste^.sledeci=glava_liste;  
glava_liste:=element_liste;
```





Једноструко повезане листе

1. Написати програм којим се уносе цели бројеви све док се не унесе 0. Унете бројева памтити у једноструко повезаној листи. Исписати унете бројеве у обрнутом редоследу од оног којим су унети



Једноструко повезане листе

1. Решење :

```
program liste;
type
  pokazivac = ^slog;
  slog = record
    broj:integer;
    sledeci : pokazivac;
  end;
var
  glava:pokazivac;
  elem:pokazivac;
  broj:integer;
begin
  readln(broj);
  glava:=nil;
```

```
while (broj <> 0) do
begin
  new(elem);
  elem^.broj:=broj;
  elem^.sledeci:=glava;
  glava:=elem;
  readln(broj);
end;
elem:=glava;
writeln('Uneti brojevi su( u obrnutom poretku) ');

while (elem <> nil ) do
begin
  writeln(elem^.broj);
  elem:=elem^.sledeci;
end;

end.
```