

2. Premenné ako ukazovatele na objekt



- Pre vlastné dátové typy môžeme definovať premenné
- Takáto premenná je v Jave **ukazovateľ** na objekt tohto typu

Príklad

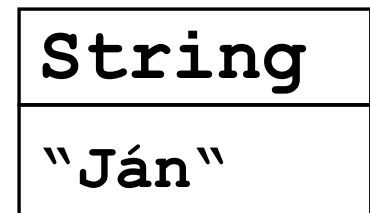
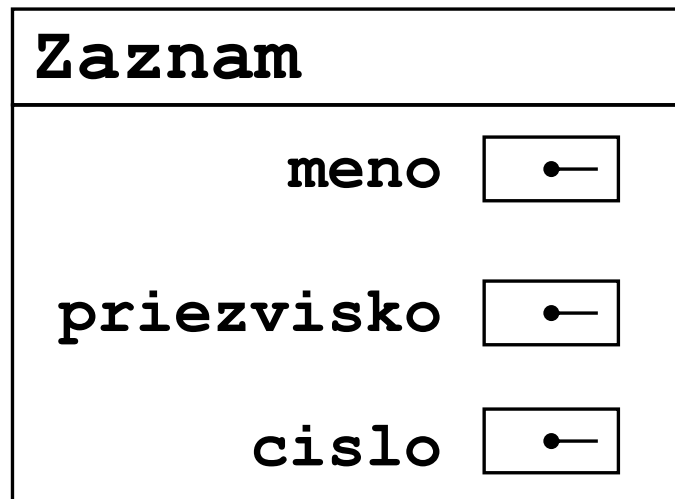
Príklad:

```
Zaznam z;
```

```
z = new Zaznam();
```

```
z.meno = "Ján";
```

z 



Premenné ako ukazovatele na objekt

Objektové premenné sa správajú analogicky ku premenným typu pole:

- Parameter vo vlastnom dátovom type je referencia (adresa v pamäti, kde sa nachádza objekt, na ktorý premenná ukazuje)
- Všetky atribúty objektu vytvoreného použitím kľúčového slova `new` a konštruktora obsahujú preddefinovanú hodnotu dátového typu atribútu (čísla: `0` resp.. `0.0`, znaky: `\u0000`, boolean: `false`, inak `null`)

`null`:

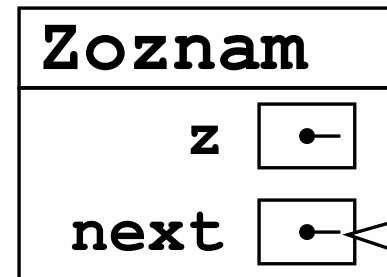
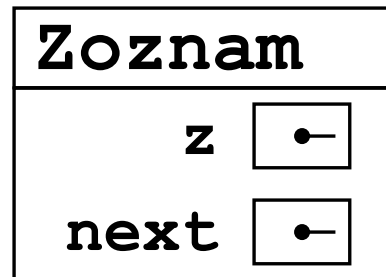
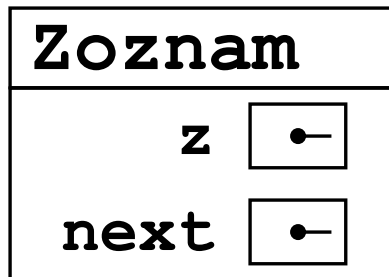
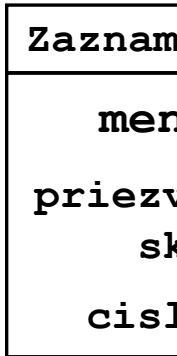
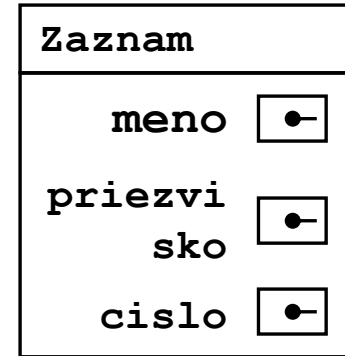
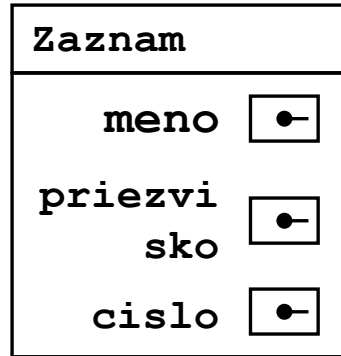
Nedefinovaná adresa v pamäti

- Preddefinovaná hodota neprimitívnych dátových typov je `null`
- `null` hodnotu môžeme priradiť každej premennej neprimitívneho dátového typu (hodnotu „`null`“ má každý neprimitívny dátový typ, presnejšie neskôr)
- Keď platí `z == null` ; potom zavolanie `z.meno` vedie k chybe!!

```
Exception in thread "main" java.lang.NullPointerException
    at Test.main(Compiled Code)
```

3. Rekurzívne dátové typy

```
class Zoznam  
{ Zaznam z;  
  Zoznam next;  
}
```



Končí s
null

Princíp:



- Attribút objektu môže byť rovnakého dátového typu ako objekt samotný!
(V dátovom type `zoznam` je atribút `next` opäť typu `zoznam`)
- Toto je možné, keďže hodnota atribútu je **ukazovateľ** na objekt (a nie objekt samotný)
- Hodnota `null` zodpovedá ukončeniu zoznamu

Príklad:

Telefónny zoznam



- Doteraz iba pole záznamov
- Zretazeny_zoznam.java
- Záznamy v abecednom poradí

Príklad:

Telefónny zoznam



```
import java.io.*;

class Obsah
{ String meno = "";
  String priezvisko = "";
  String cislo = "";
}

class Prvok_zoznamu
{ Obsah obsah;
  Prvok_zoznamu dalsi;
}
```


Príklad:

Telefónny zoznam



```
public class Zretazeny_zoznam
{
    public static Obsah readObsah()
    { Obsah obsah = new Obsah();

        obsah.meno = Zklavesnice.readString("Zadaj meno");
        obsah.priezvisko = Zklavesnice.readString("Zadaj priezvisko");
        obsah.cislo = Zklavesnice.readString("Zadaj telefonne cislo");

        return obsah;
    }
}
```

Príklad:

Telefónny zoznam

```
public static void printObsah(Obsah obsah)
{ System.out.println(obsah.meno) ;
  System.out.println(obsah.priezvisko) ;
  System.out.println(obsah.cislo) ;
  System.out.println() ;
}
public static void printPrvok_zoznamu(Prvok_zoznamu
  prvok)
{ if (prvok != null)
  { printObsah(prvok.obsah) ;
    printPrvok_zoznamu(prvok.dalsi) ;
  }
}
```

Príklad:

Telefónny zoznam

```
public static void printNameAdr(Liste liste,
                                String name)
{ if (liste != null)
  { if (name.equals(liste.adr.name))
    { printAdr(liste.adr); }

    if (name.compareTo(liste.adr.name) >= 0)
    { printNameAdr(liste.next, name); }
  }
}
```

Príklad:

Telefónny zoznam

```
public static void printPriezvisko (Prvok_zoznamu prvok,
                                     String priezvisko)
{ if (prvok != null)
  { if (priezvisko.equals(prvok.obsah.priezvisko))
    { printObsah(prvok.obsah); }

    if (priezvisko.compareTo(prvok.obsah.priezvisko) >= 0)
    { printPriezvisko(prvok.dalsi, priezvisko); }
  }
}
```

Príklad:

Telefónny zoznam

```
public static Prvok_zoznamu insertObsah(Prvok_zoznamu prvok, Obsah obsah)
{ if (prvok == null)
  { Prvok_zoznamu vkladany = new Prvok_zoznamu();
    vkladany.obsah = obsah;
    return vkladany;
  }
  else if (prvok.obsah.priezvisko.compareTo(obsah.priezvisko) <= 0)
  { prvok.dalsi = insertObsah(prvok.dalsi, obsah);
    return prvok;
  }
  else
{ Prvok_zoznamu vkladany = new Prvok_zoznamu();
  vkladany.obsah = obsah;
  vkladany.dalsi = prvok;

  return vkladany;
}
}
```

Príklad:

Telefónny zoznam

```
public static void main(String[] pole_parametrov)
{ Prvok_zoznamu prvy = null;
  int pocetZaznamov = 0;
  char c;
  do
  { c = Zklavesnice.readChar("Zvol 1-na zadanie zaznamu, 2-na vyhladanie
zaznamu alebo 3-na vypis vsetkych zaznamov");
    switch (c)
    { case '1' : prvy = insertObsah(prvy,readObsah());
      pocetZaznamov++;
      break;
      case '2' : String priezvisko = Zklavesnice.readString("Priezvisko:");
printPriezvisko(prvy, priezvisko);
      break;
      case '3' : System.out.println("Vypis telefonneho zoznamu");
      printPrvok_zoznamu(prvy);
      break;
    }
  }
  while ( c == '1' || c == '2' || c == '3');
}
```

Dátové štruktúry

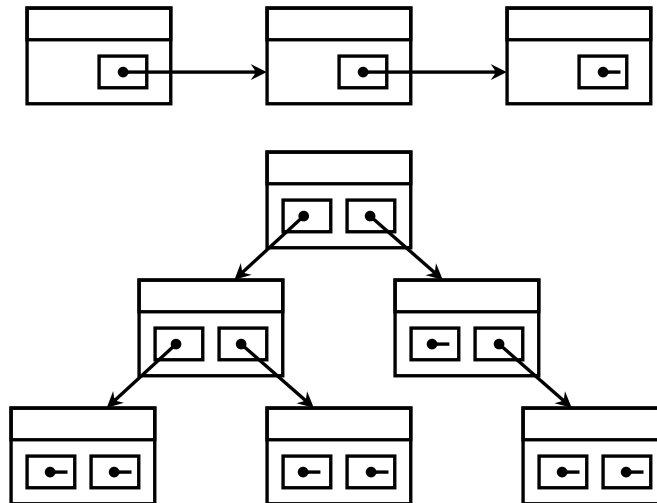
□ Rekurzívne dátové štruktúry sú základom informatiky

□ Zoznamy

□ Stromy

□ Grafy

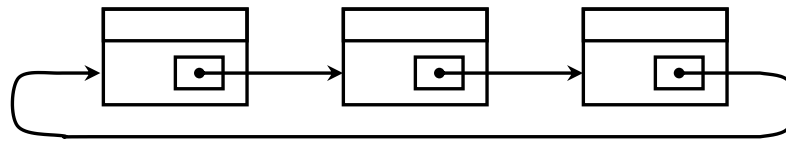
...



Datenstrukturen:

Vorlesung Informatik I

- Sú možné aj cyklické štruktúry!
(ľubovoľné grafy)
Príklad: cyklický zoznam



4. Rovnosť objektov



Dve premenné typu Objekt **a** a **b** majú rovnaký obsah vzhľadom na `==` práve vtedy, ak ukazujú na ten istý objekt, alebo obe majú hodnotu `null`.

Protipríklad:

```
Zaznam a, b;  
a = new Zaznam();  
b = new Zaznam();
```

Tu platí

$a \neq b$

```
a.priezvisko = "Juhas";  
b.priezvisko = a.priezvisko;  
a.meno = "Gabriel";  
b.meno = a.meno;  
a.cislo = „123“;  
b.cislo = a.cislo;
```

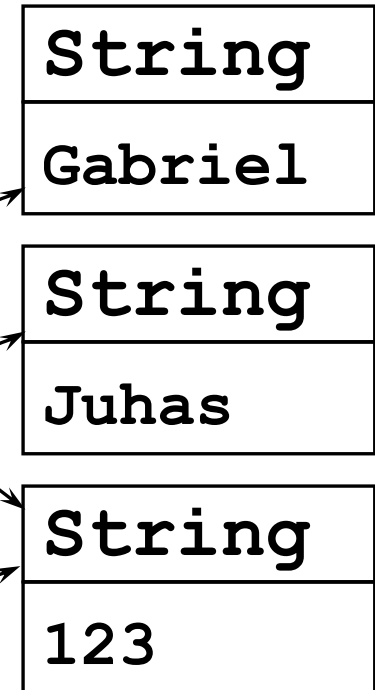
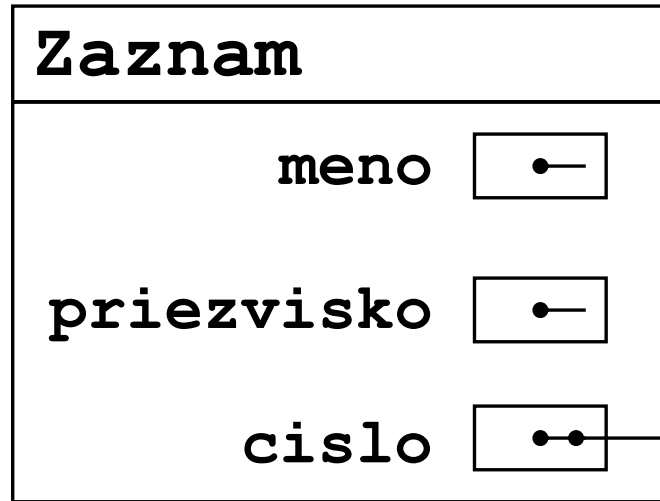
Stále platí

$a \neq b$

Objekty majú rôznu
identitu a rovnaký
obsah

Protipríklad

a 



b 

