Informatika 1

Spolupráca objektov

KST Katedra softvérových technológií
Fakulta riadenia a informatiky, Žilinská univerzita v Žiline

Pojmy zavedené v 4. prednáške(1)

- modularizácia a abstrakcia
- kompozícia – skladanie objektov
  - spoločné životné cykly
  - UML

Pojmy zavedené v 4. prednáške(2)

- objektové typy
- referencie
- diagram objektov

Pojmy zavedené v 4. prednáške(3)

- príkaz na poslanie správy
  - príkaz na poslanie správy "new" triede
  - príkaz na poslanie správy bez návratovej hodnotu
  - príkaz na poslanie správy s návratovou hodnotou
- objektový výraz
  - výstupný parameter správy objektového typu
  - špeciálne: reťazcový výraz

Pojmy zavedené v 4. prednáške(4)

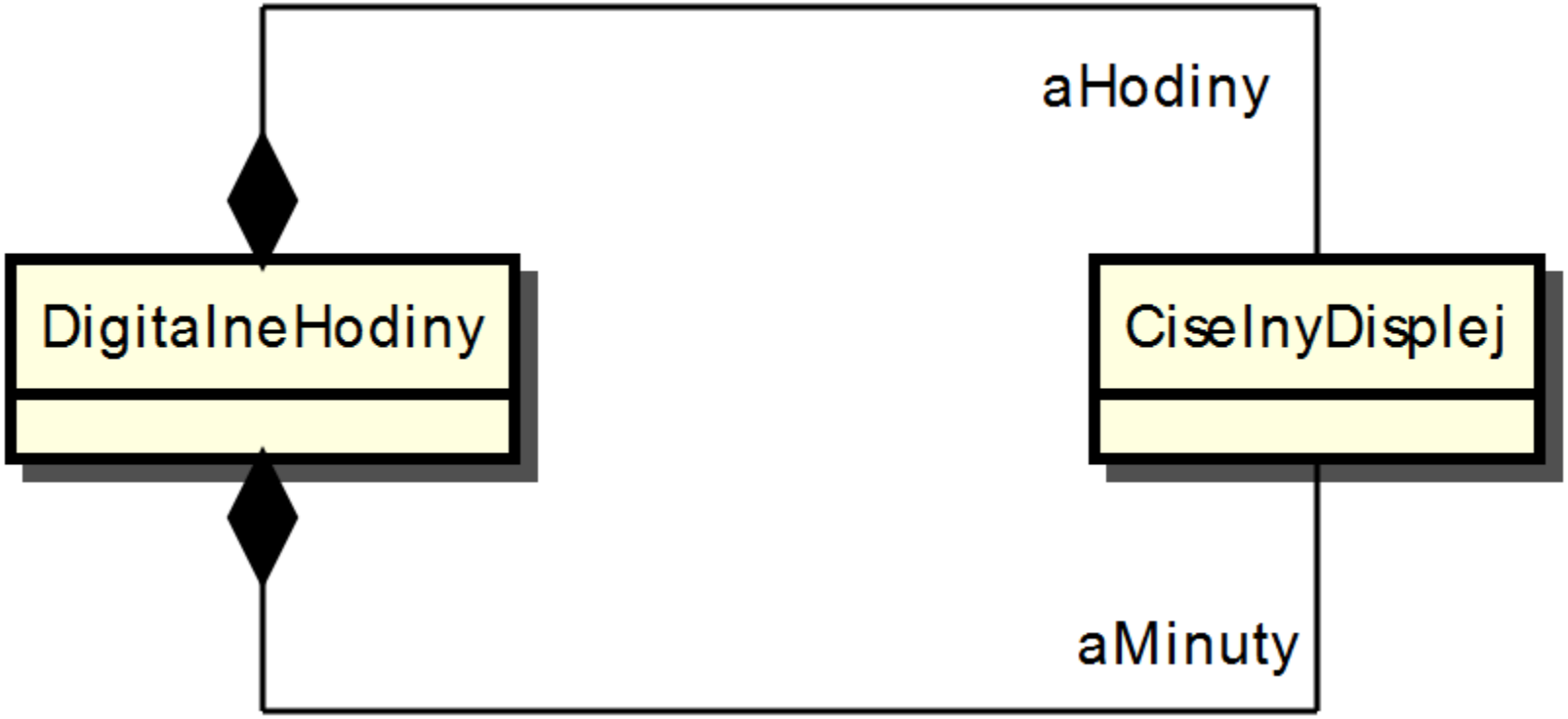
- reťazce
  - reťazcové literály
  - spájanie reťazcov
  - reťazcové výrazy
  - trieda String

Cieľ prednášky

- logické výrazy
- relačné výrazy s objektmi
- spolupráca objektov – asociácia

- príklad: Digitálne hodiny

Digitálne hodiny – verzia 1

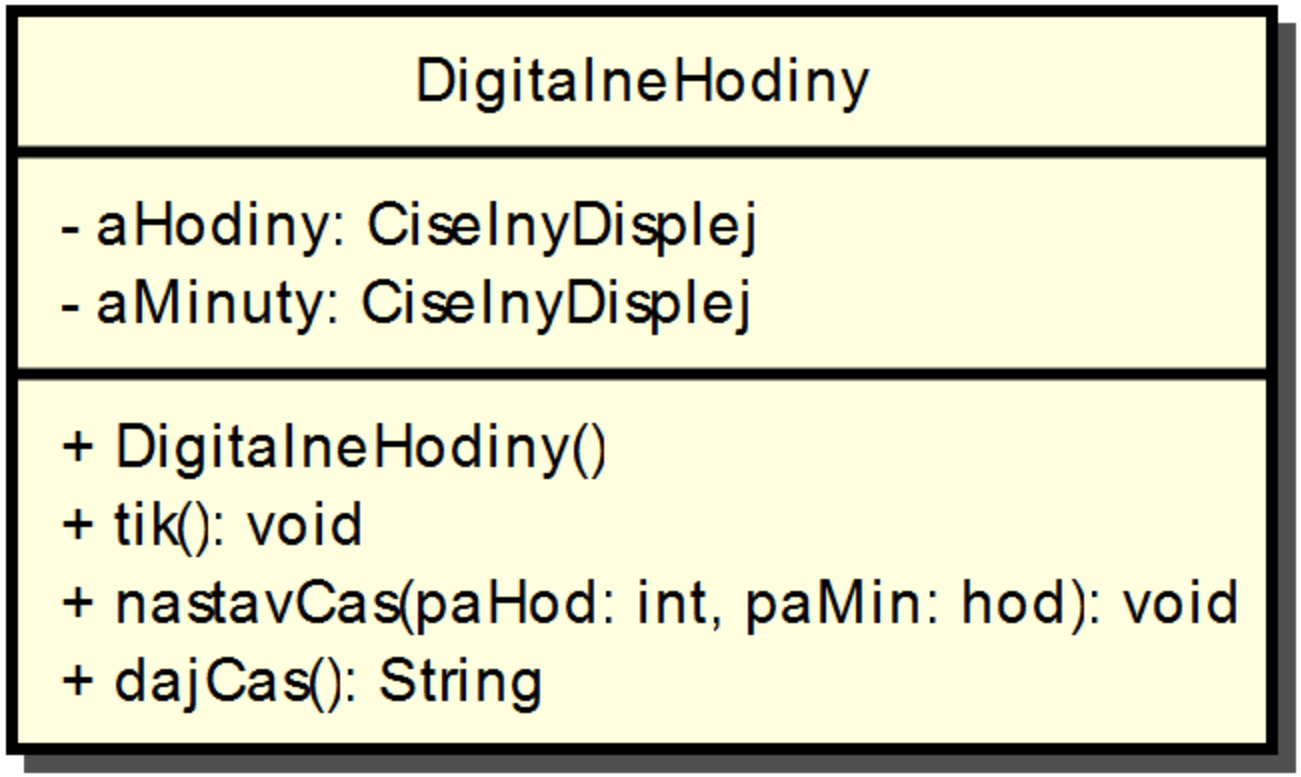


Projekt digitálne hodiny

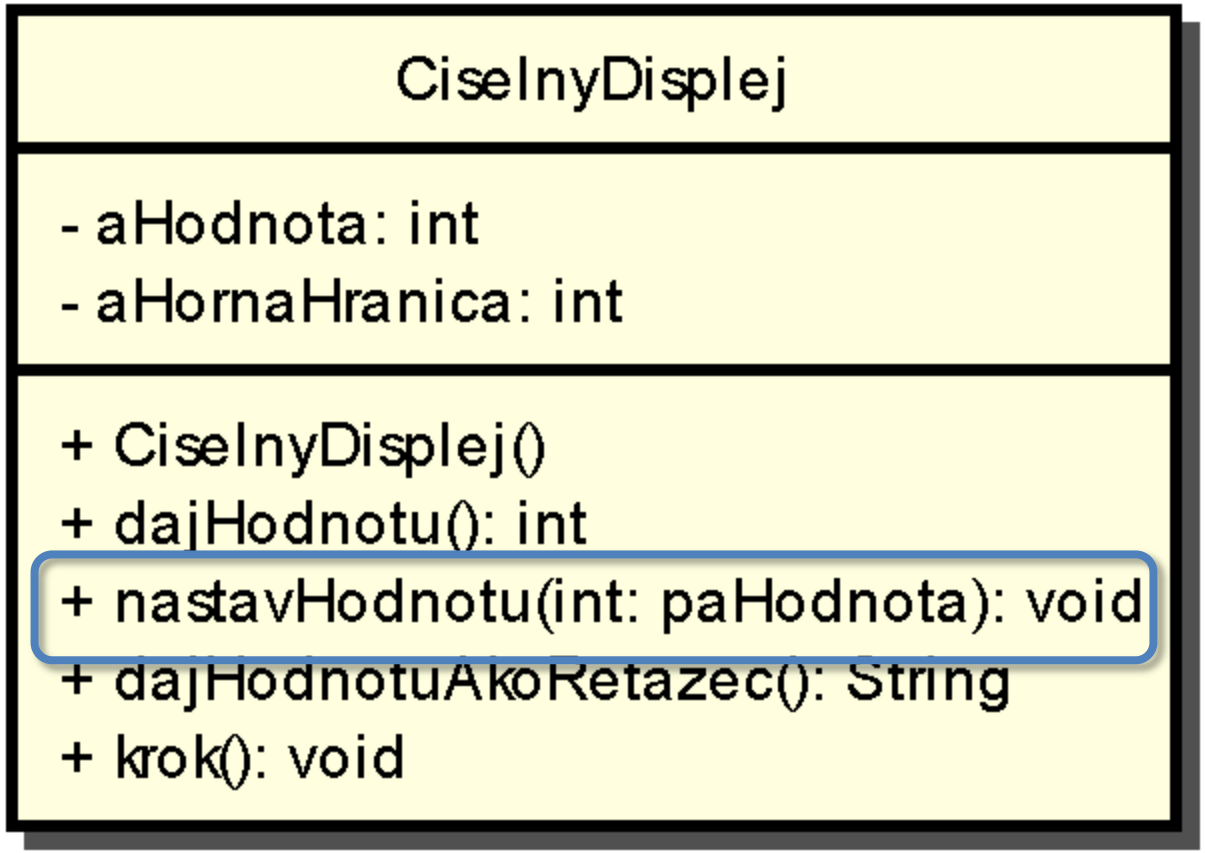
- požadované služby:
  - zobrazujú aktuálny čas 24-hod. formát
    - 00:00 – polnoc
    - 23:59 – minúta pred polnocou
- dajú sa nastaviť na požadovaný čas
- „tikajú“ – plynutie času – časový krok
  - krok 1 minúta

- trieda vytvorí inštanciu hodín
  - začiatočný čas: 00:00

DigitalneHodiny – vnútorný pohľad



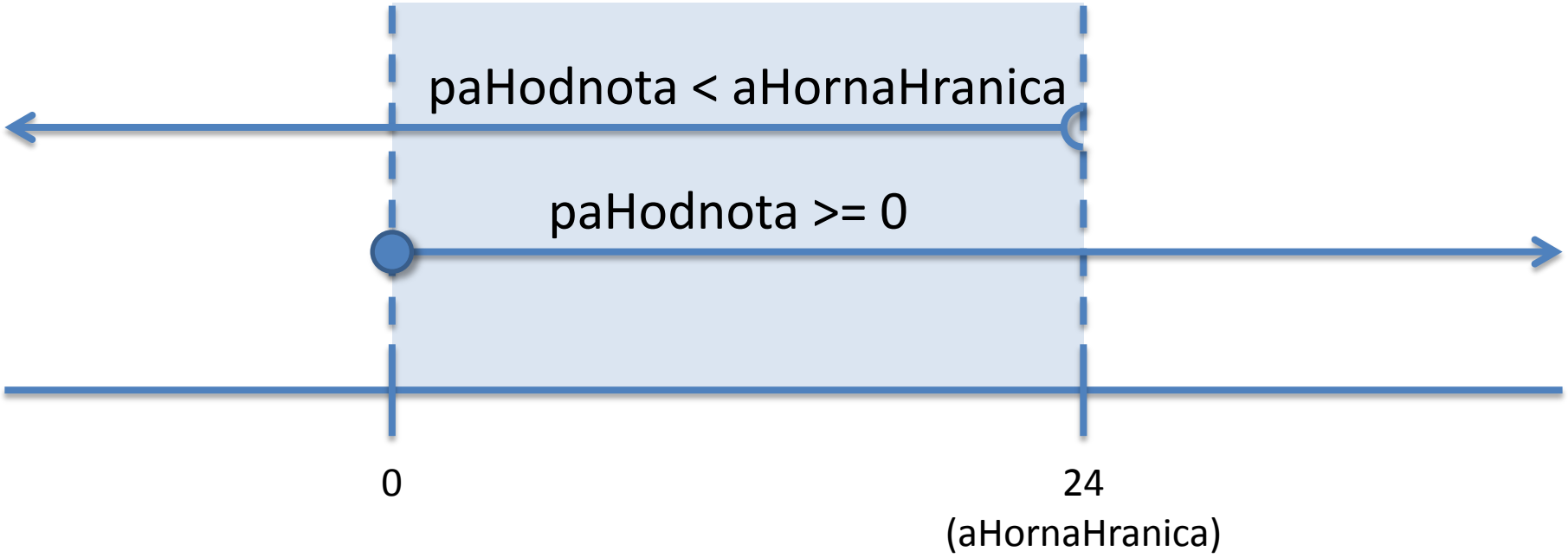
CiselnyDisplej – vnútorný pohľad



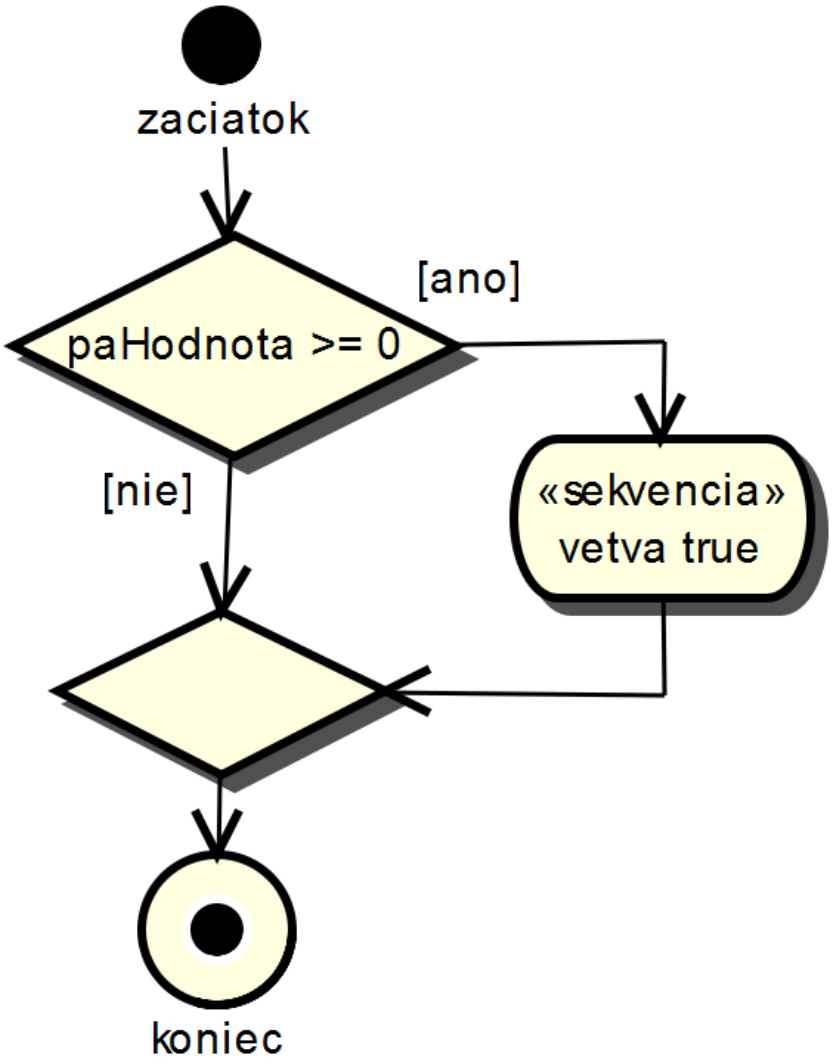
Metóda nastavHodnotu

- zmení hodnotu na displeji
- nesmie nastaviť hodnotu mimo rozsah
  - min 0
  - max aHornaHranica

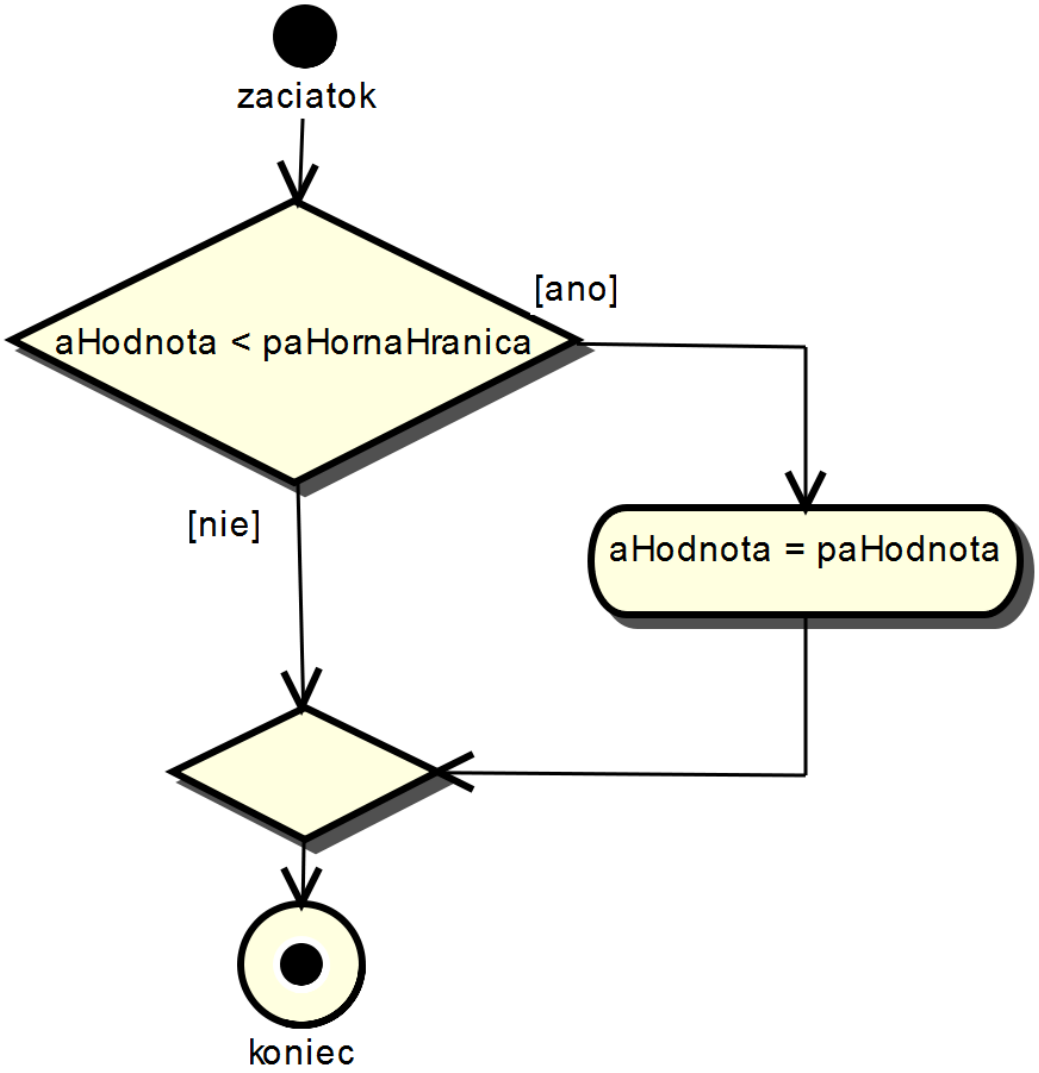
Zložená podmienka



Zložená podmienka – prvá časť



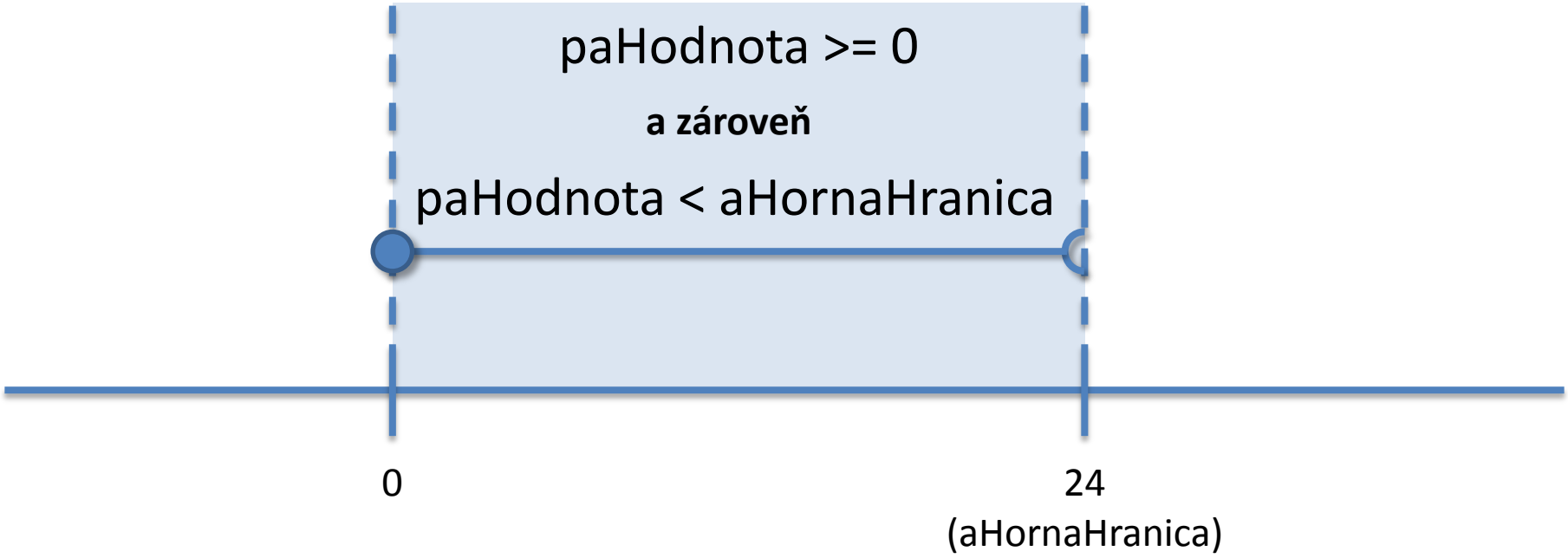
Zložená podmienka – druhá časť



Metóda nastavHodnotu

```
public void nastavHodnotu(int paHodnota)
{
    if (paHodnota >= 0) {
        if (paHodnota < aHornaHranica) {
            aHodnota = paHodnota;
        }
    }
}
```

Zložená podmienka



Logické operátory(1)

- matematické formy
  - $x \in \langle a, b \rangle$
  - $a \leq x \leq b$
  - $a \leq x \wedge x \leq b$
  - $a \leq x$ a súčastne $x \leq b$

Logické operátory(2)

- programovací jazyk Java
  - $a \leq x$ a súčastne $x \leq b$

```
a <= x && x <= b
```

Metóda nastavHodnotu

```
public void nastavHodnotu(int paHodnota)
{
    if ((paHodnota >= aDolnaHranica) && (paHodnota < aHornaHranica)) {
        aHodnota = paHodnota;
    }
}
```

Metóda nastavHodnotu

```
public void nastavHodnotu(int paHodnota)
{
    boolean vyhovuje = (paHodnota >= 0)
                        && (paHodnota < aHornaHranica);
    if (vyhovuje) {
        aHodnota = paHodnota;
    }
}
```

Logické operátory

operácia	názov	matematika	Java
negácia	not	$\bar{a}$	!a
logický súčin	and	$a \wedge b$	a && b
logický súčet	or	$a \vee b$	a || b

Priorita logických operátorov

priorita	operátory
najvyššia	unárne +, -, !
	*, /, %
	binárne +, -
	<, <=, >, >=
	==, !=
	&&
najnižšia	||

Použitie logických operátorov

- unárny operátor !

```
operátor operand
```

- binárne operátory && a ||

```
prvyOperand operátor druhyOperand
```

- operandy – vždy logická, typ boolean
- hodnota logického výrazu – logická, typ boolean

Pravdivostné tabuľky

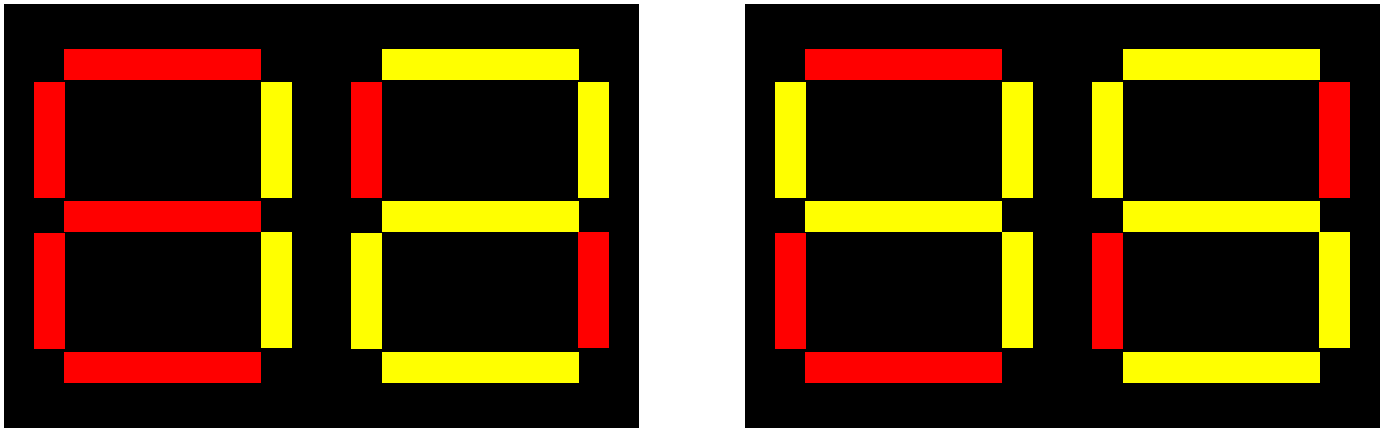
&& (and)	false	true
false	false	false
true	false	true

|| (or)	false	true
false	false	true
true	true	true

! (not)	false	true
	true	false

Úloha

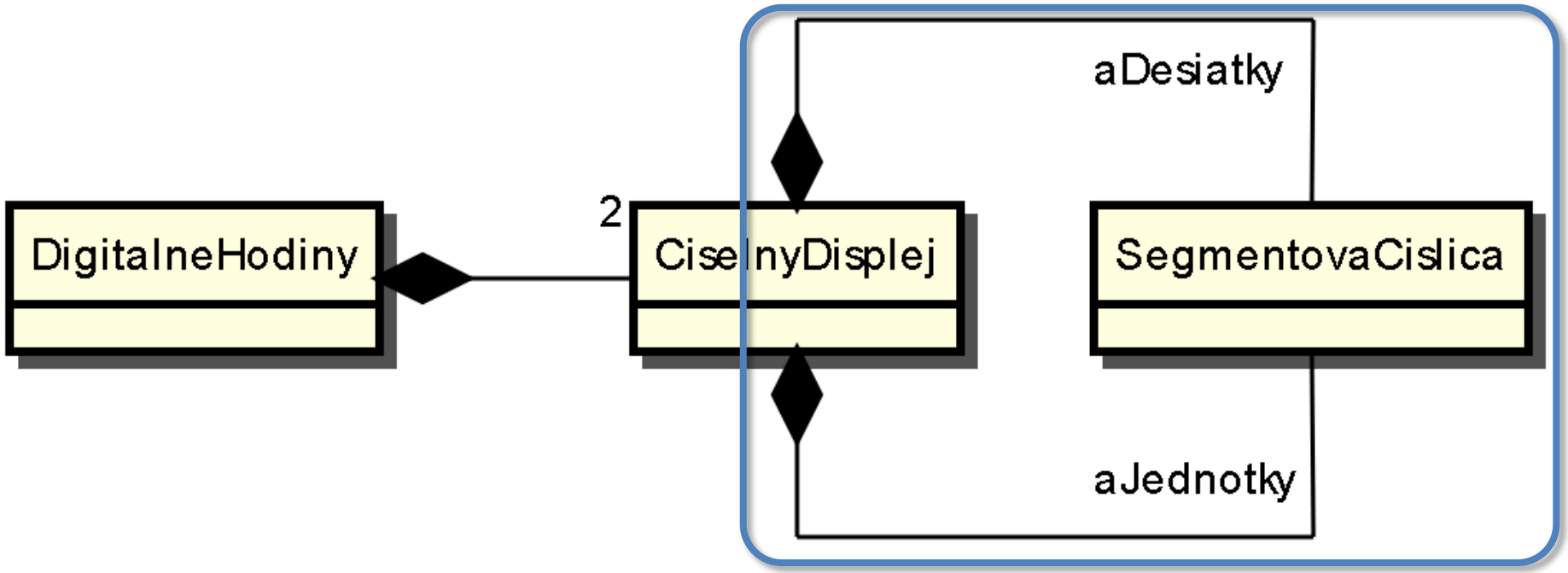
- zobraziť hodiny pomocou sedem segmentového displeja



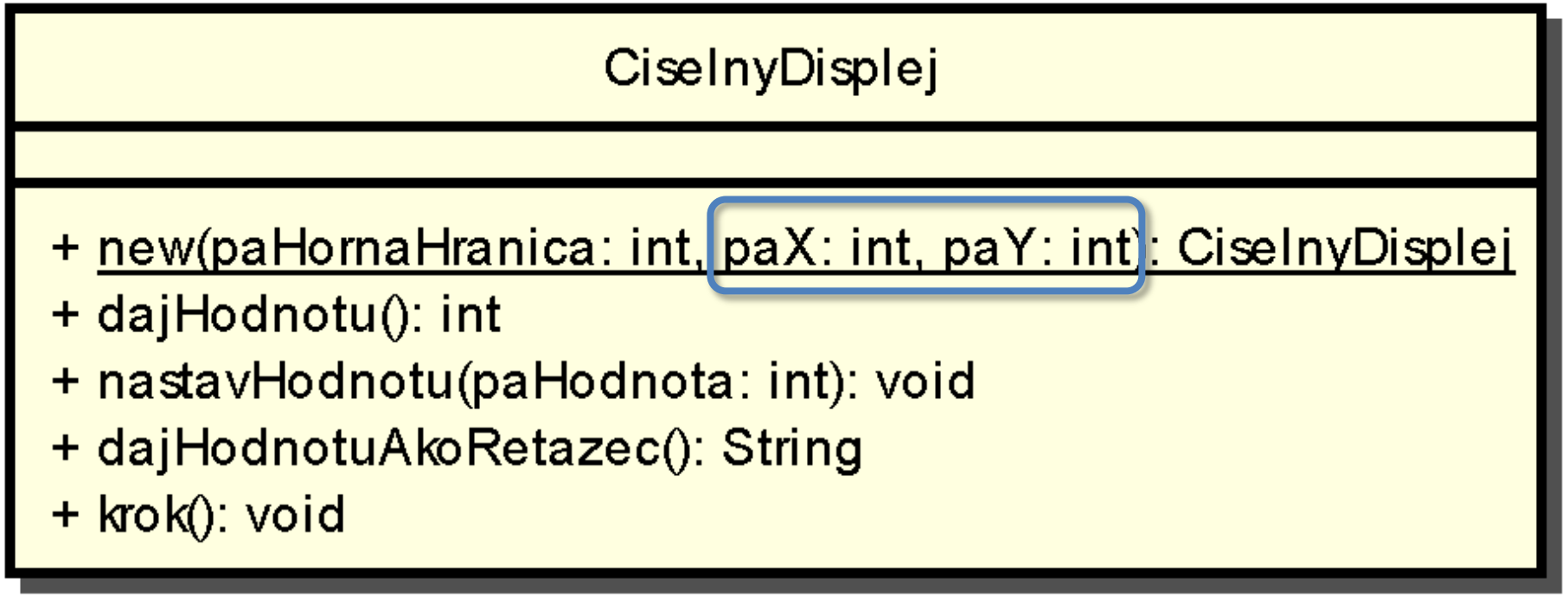
Riešenie – prvá možnosť

- kompozícia – skladanie
  - SegmentovaCislica pre desiatky
  - SegmentovaCislica pre jednotky
- nutnosť určenia pozície
  - každá číslica je na inej pozícii
  - každý číselný displej je na inej pozícii

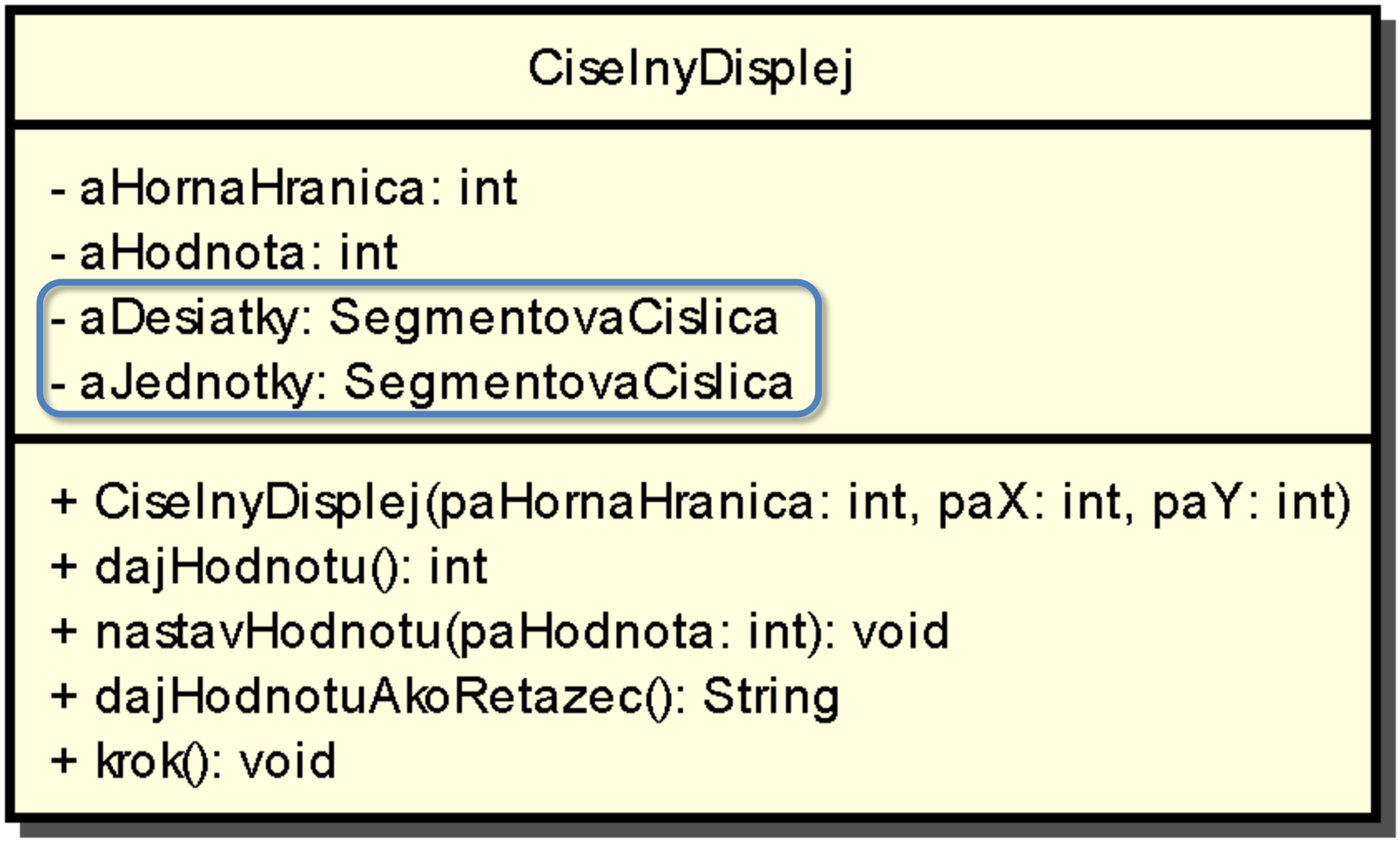
Segmentová číslica – skladanie



CiselnyDisplej – rozhranie



CiselnyDisplej – vnútorný pohľad



CiselnyDisplej – konštruktor

```
public CiselnyDisplej(int paHornaHranica,
                                    int paX, int paY)
{
    aHornaHranica = paHornaHranica;
    aHodnota = 0;
    aDesiatky = new SegmentovaCislica(paX, paY);

    int posunX = 120 + 10;
    aJednotky = new SegmentovaCislica(
                                    paX + posunX, paY);
}
```

CiselnyDisplej – krok

```
public void krok()
{
    aHodnota = (aHodnota + 1) % aHornaHranica;
    aDesiatky.zobraz(aHodnota / 10);
    aJednotky.zobraz(aHodnota % 10);
}
```

Asociácia – spolupráca objektov

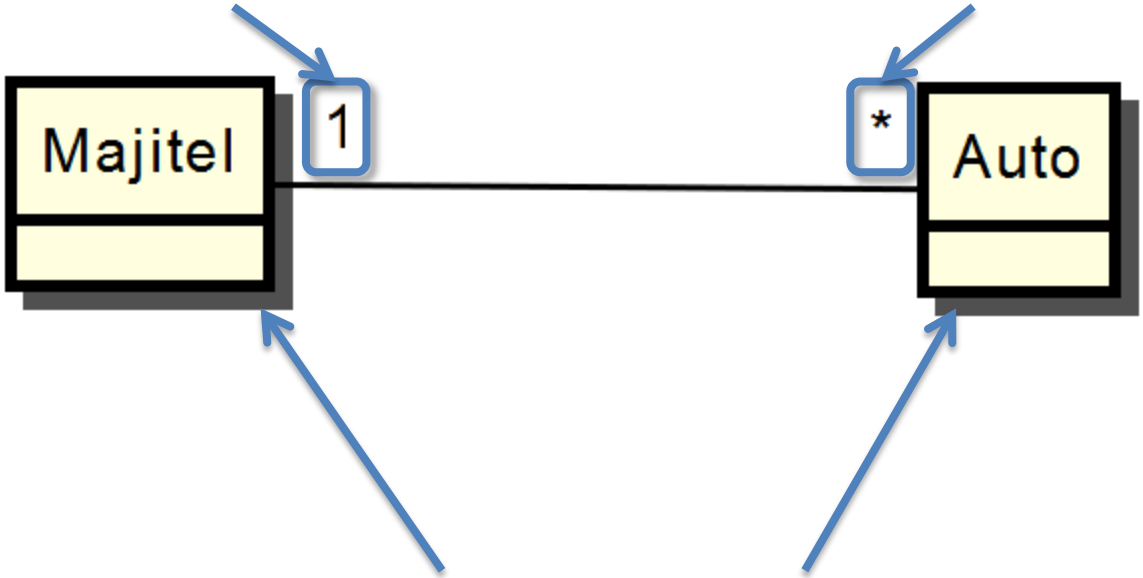
- druhá možnosť implementácie

- asociácia – ľubovoľná spolupráca dvoch objektov
  - príklady: klient a banka, učiteľ a študent

- charakteristika asociácie – nezávislé životné cykly oboch objektov

Asociácia v UML

jeden majiteľ môže mať viac aut

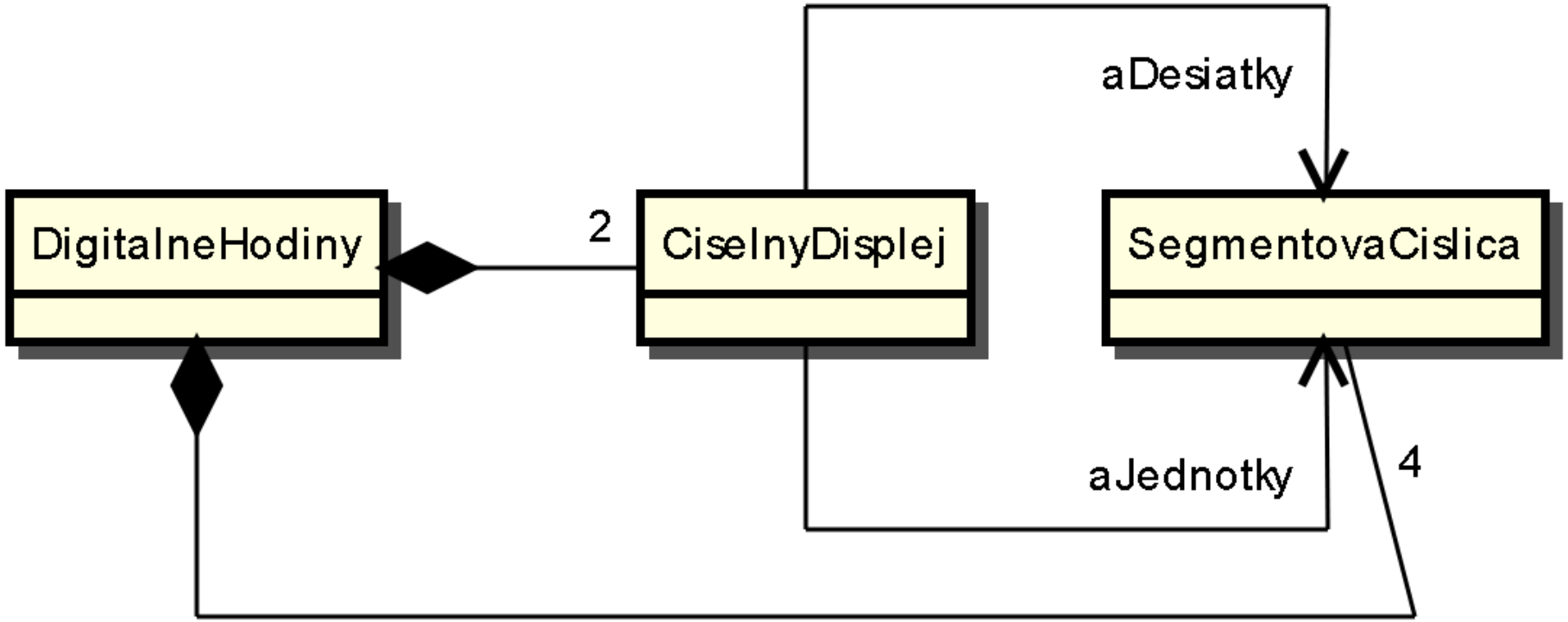


rovnocenné objekty

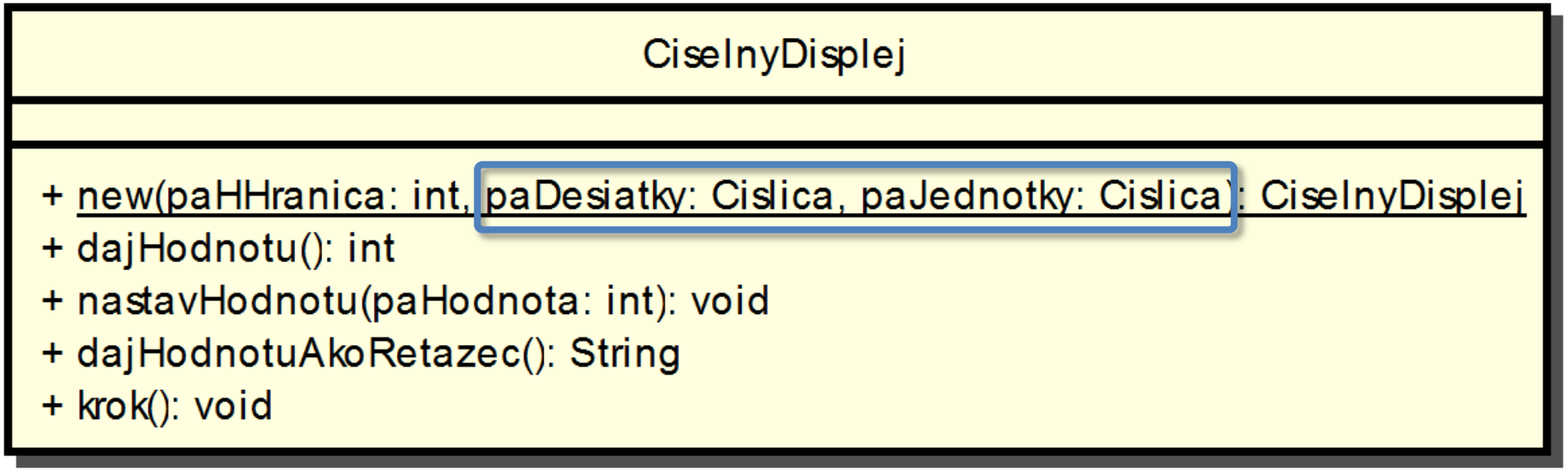
Riešenie zobrazenia asociáciou

- asociácia – spolupráca
  - inštanciu SegmentovaCislica vytvárajú DigitalneHodiny
  - CiselnyDisplej spolupracuje so SegmentovaCislica
  - CiselnyDisplej dostane číslicu ako parameter konštruktora
- pozíciu číslic určujú hodiny

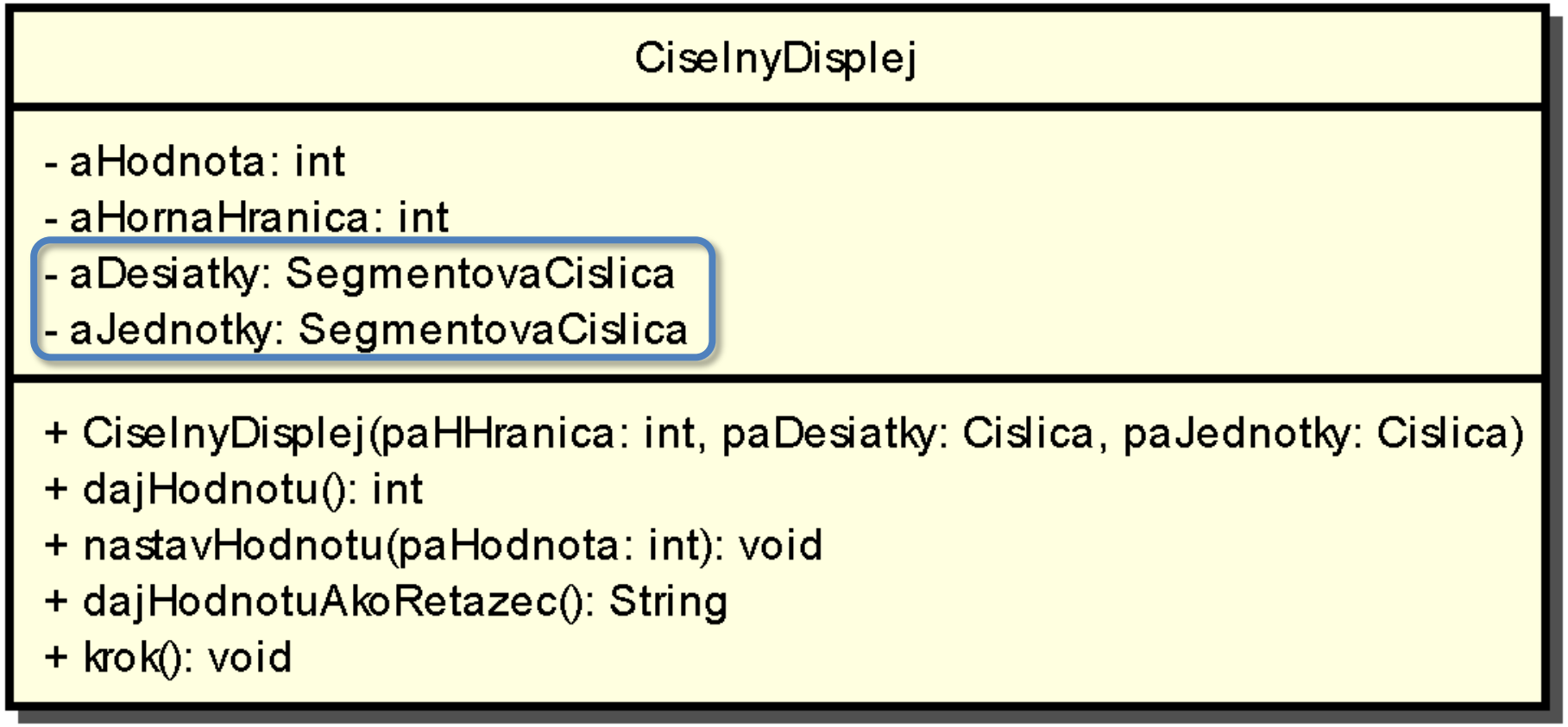
SegmentovaCislica – asociácia



CiselnyDisplej – rozhranie



CiselnyDisplej – vnútorný pohľad



CiselnyDisplej – konstruktor

```
public CiselnyDisplej(int paHornaHodnota,
                      Cislica paDesiatky, Cislica paJednotky)
{
    aHornaHranica = paHornaHranica;
    aHodnota = 0;
    aDesiatky = paDesiatky;
    aJednotky = paJednotky;
}
```

CiselnyDisplej – krok

```
public void krok()
{
    aHodnota = (aHodnota + 1) % aHornaHranica;
    if (aDesiatky != null && aJednotky != null) {
        aDesiatky.zobraz(aHodnota / 10);
        aJednotky.zobraz(aHodnota % 10);
    }
}
```

Hodnota null

- null – objektový literál
- null = referencia neodkazuje na žiadny objekt
- pre referenciu na inštanciu ľubovoľnej triedy

Relačné výrazy s objektmi(1)

- relačné operátory pre čísla
  - < (menšie ako), <= (menšie, alebo rovné), > (väčšie ako), >= (väčšie, alebo rovné), == (rovné), != (nerovné)
- relačné operátory pre objekty
  - == (rovné), != (nerovné)
- porovnanie referencií na objekty

```
referencia1 == referencia2
```

Relačné výrazy s objektmi(2)

- == porovnanie identity dvoch objektov
  - true – dve referencie na ten istý objekt
  - false – dve referencie na dva rôzne objekty

- != opak operátora ==

Porovnanie hodnôt objektov

- porovnanie hodnôt/stavov dvojice objektov?

- porovnanie dvojice rovnakého typu
- porovnanie zhody ich stavov
- špeciálna správa – metóda equals

Porovnávanie reťazcov – String

- správa equals
- testuje zhodnosť dvoch reťazcov
- hlavička metódy

```
public boolean equals(String paRetazec)
```

- príklad:

```
System.out.println("Zilinska".equals("univerzita"));
```

String – „==“ a equals(1)

- dve rôzne referencie na ten istý reťazec
- dva rôzne reťazce – rovnaký obsah

- relačný operátor „==“
  - „==“ – porovnanie referencií na reťazce
  - true – referencie na ten istý reťazec
- správa equals
  - equals – porovnanie obsahov reťazcov
  - true – rovnaký obsah dvoch reťazcov

String – „==“ a equals(2)

- reťazec1 == reťazec2 – true
- ⇒ reťazec1.equals(reťazec2) – true

- POZOR: opačná implikácia neplatí

- reťazec1.equals(reťazec2) – true
- ⇒ reťazec1 == reťazec2 – true

String – „==“ a equals(3)

```
String nazov = "Zilinska univerzita";
// "zilinska univerzita"
String nazovA = nazov.toLowerCase();
// "zilinska univerzita"
String nazovB = nazov.toLowerCase();
```

- dva rôzne objekty s rovnakým stavom
  - nazovA.equals(nazovB) – true
  - nazovA == nazovB – false

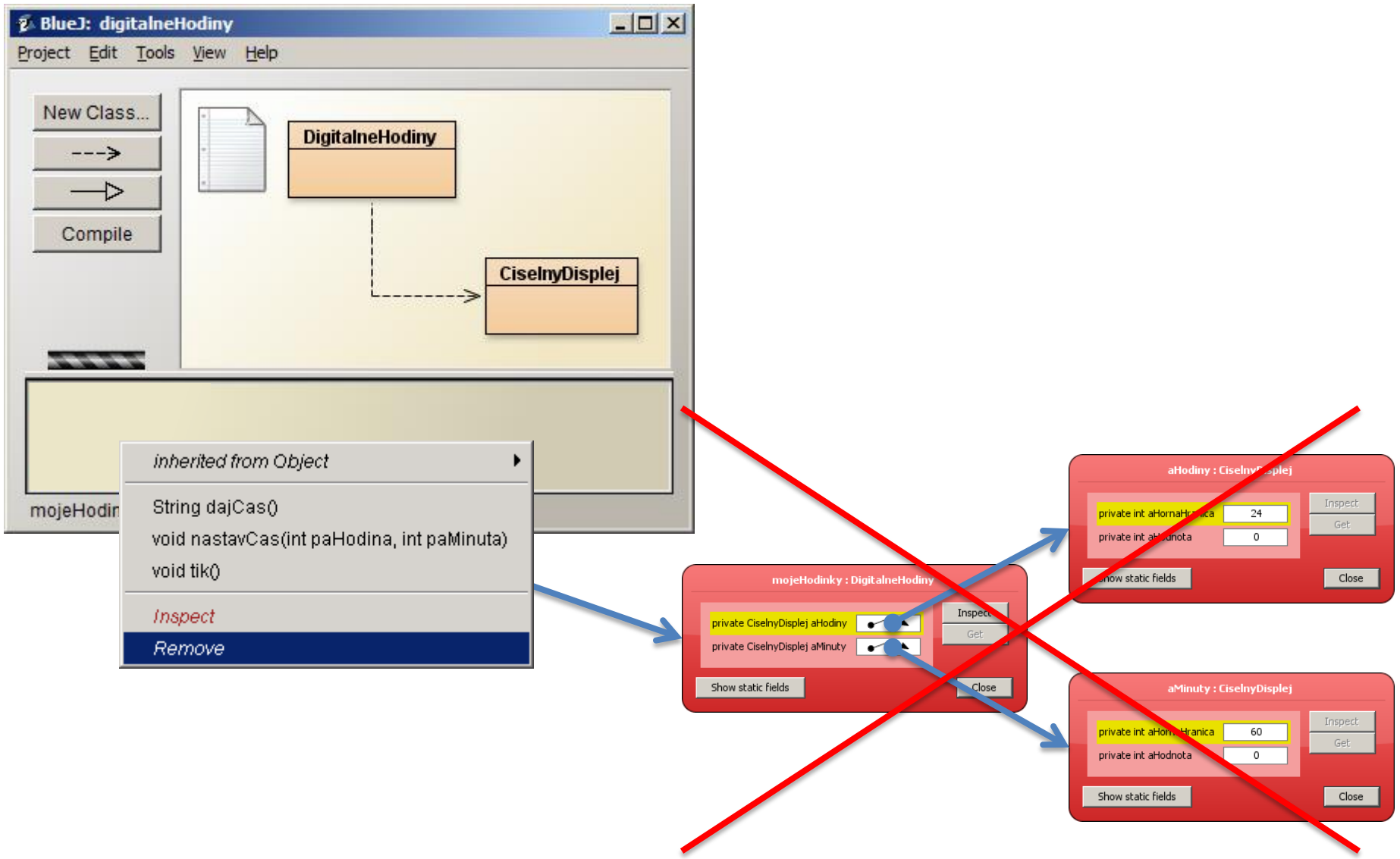
Zánik inštancie

- životný cyklus inštancie
  - vznik inštancie – špeciálna správa „new“ triede
  - poskytovanie služieb – prijímanie správ a reakcie na ne
  - zánik inštancie – ???

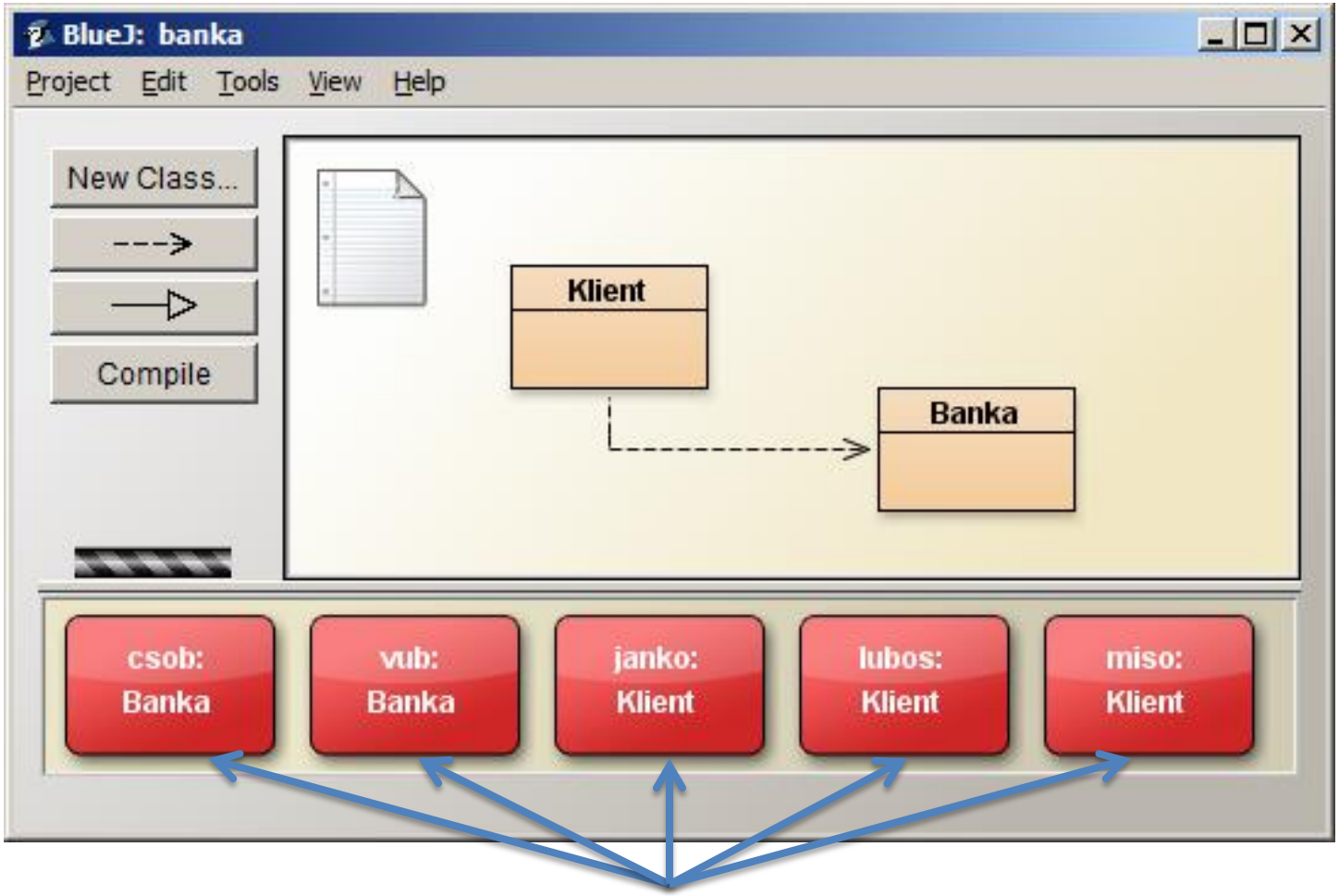
Zánik inštancie

- o zrušenie inštancie sa stará zberná služba – garbage collector
- kedy?
  - zberná služba ruší objekt v prípade, že naň neexistuje žiadna referencia
  - null = referencia neodkazuje na žiadny objekt

Zánik objektov pri kompozícii

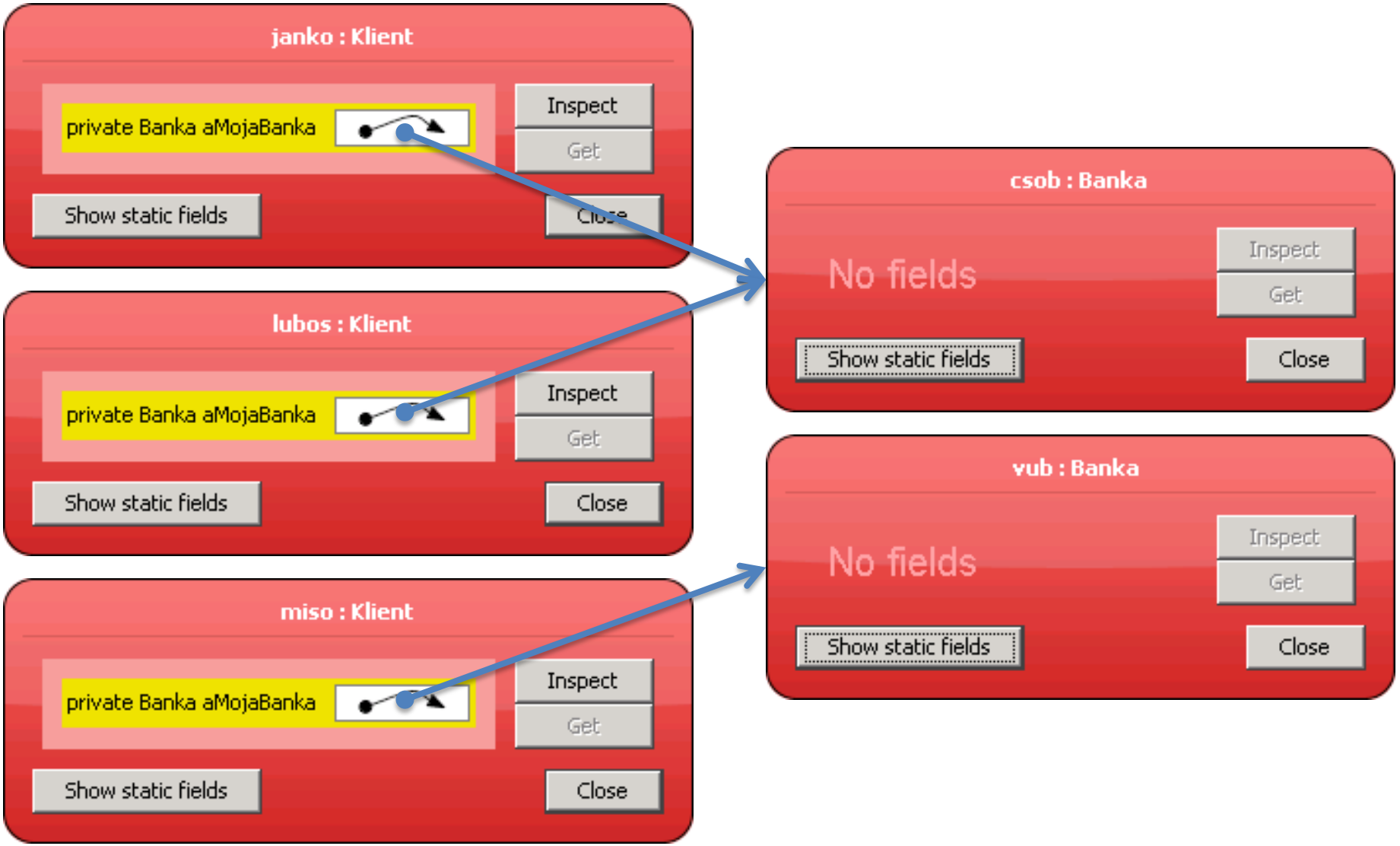


Referencie v BlueJ

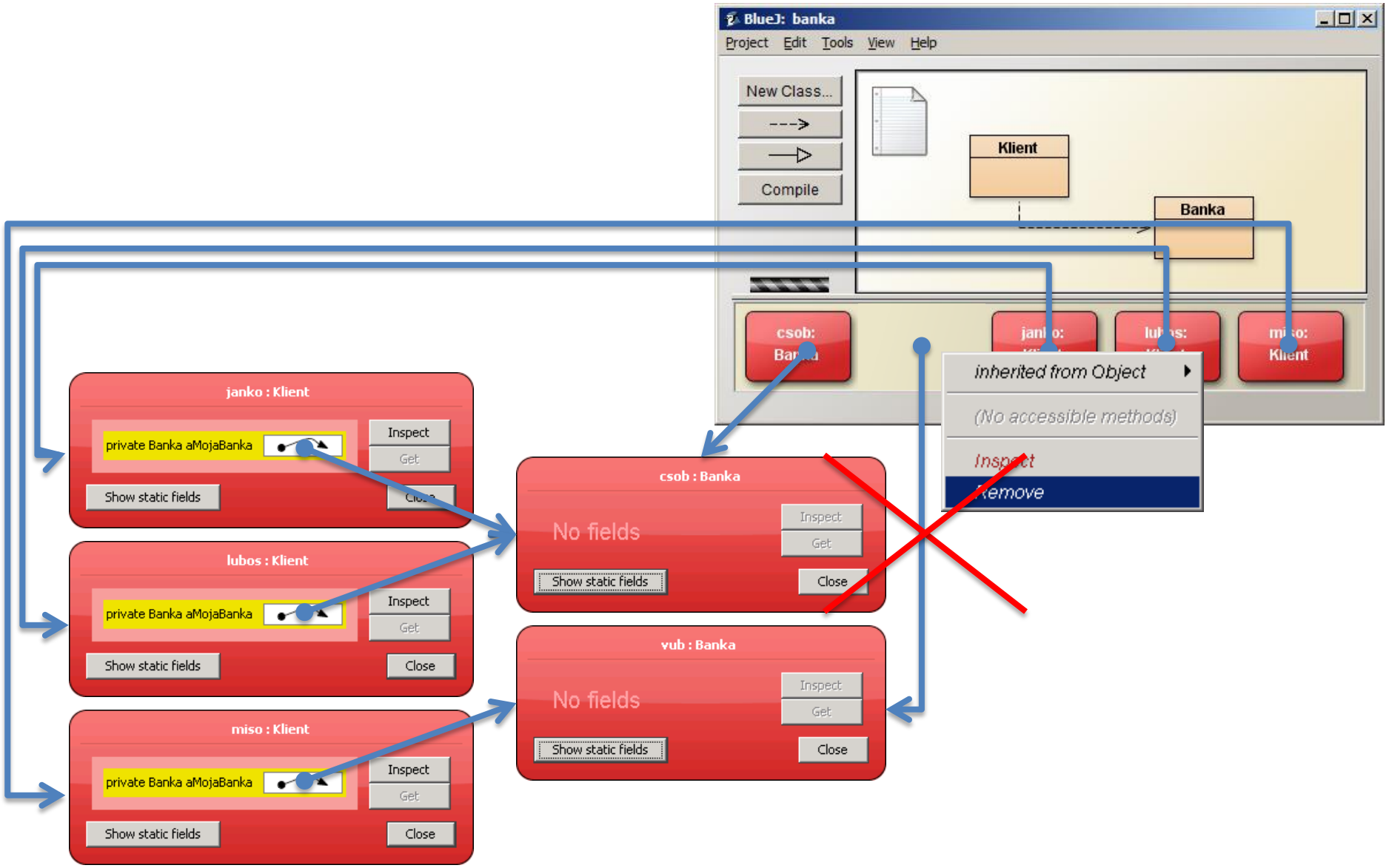


referencia

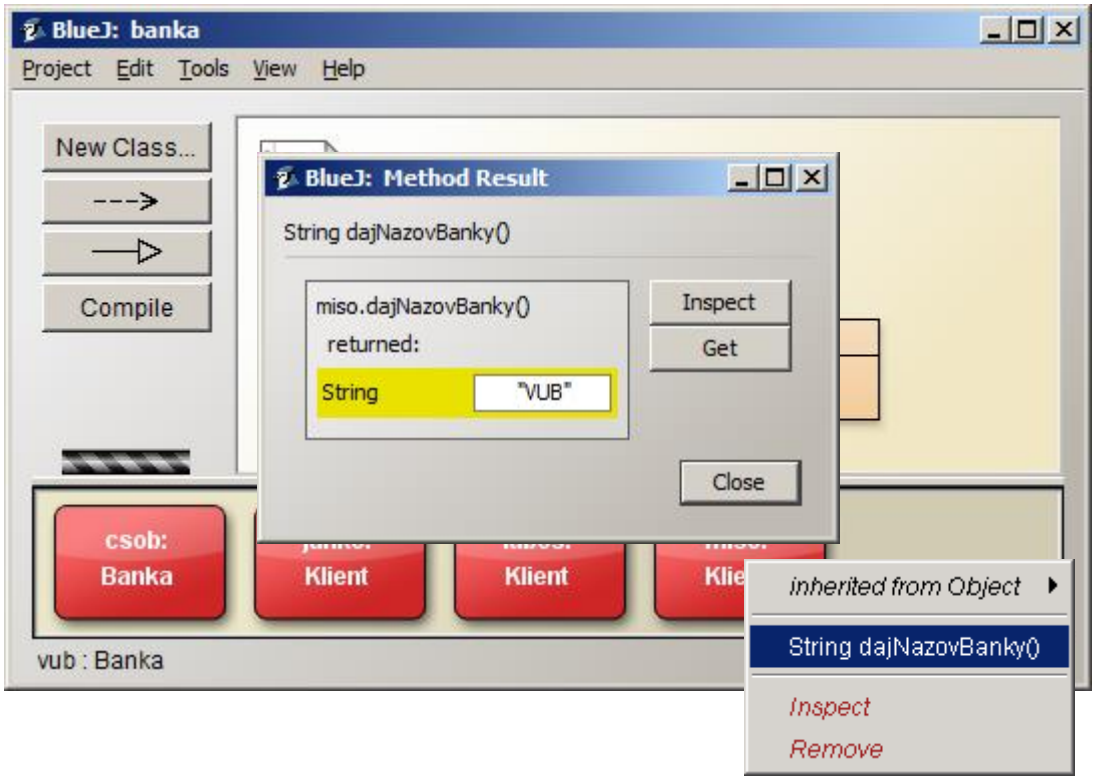
Objektový diagram príkladu



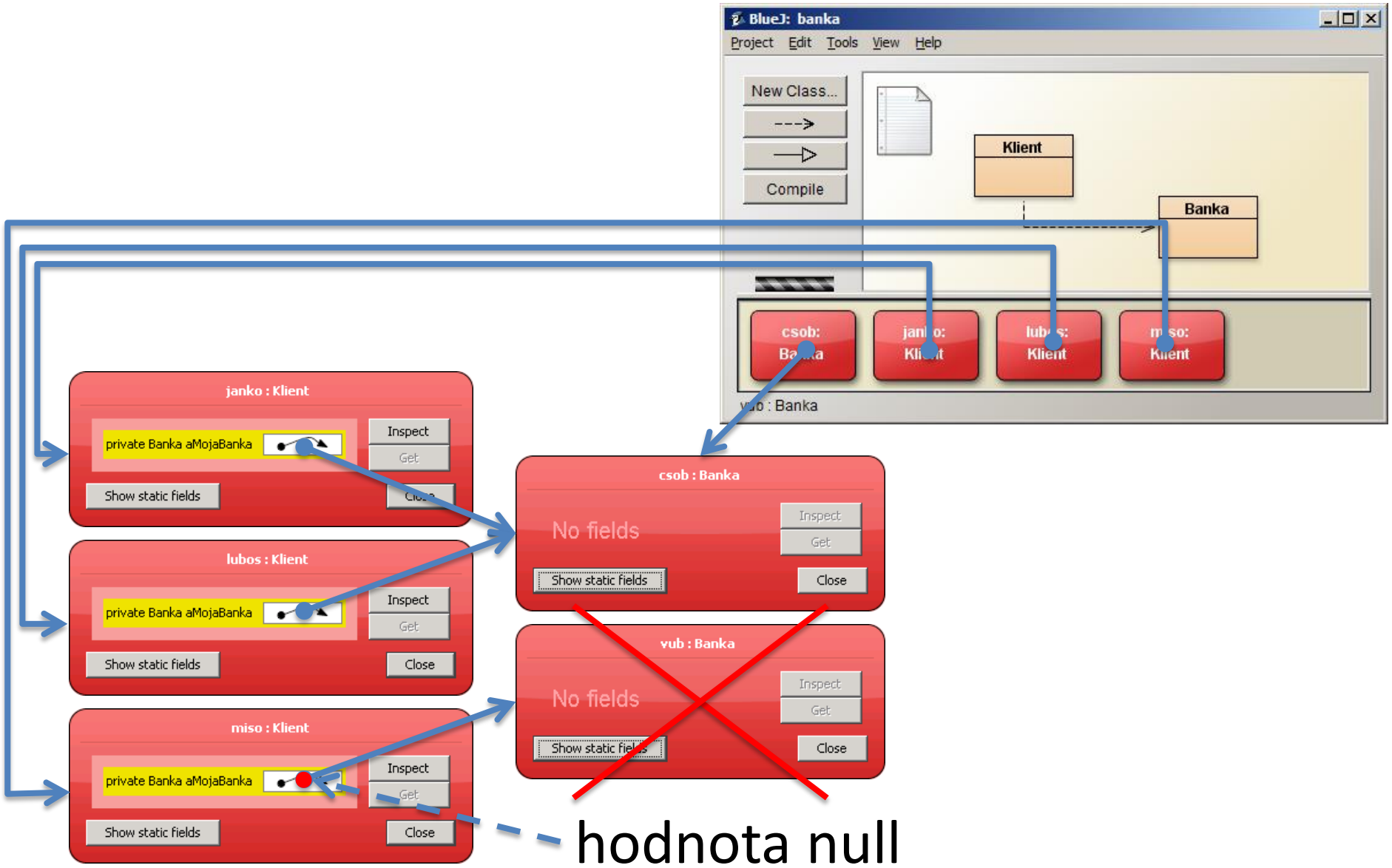
Zánik objektov pri asociácii(1)



Zánik objektov pri asociácii(2)



Zánik objektov pri asociácii(2)



Hodnota null – úlohy(1)

- null = referencia neodkazuje na žiadny objekt
- inicializácia objektovej premennej
- zánik inštancie
  - priraďovací príkaz
  - premenna = null;

Hodnota null – úlohy(2)

- adresát nie je určený
  - podmienka pre poslanie správy
  - premenna != null

Vďaka za pozornosť