

6

Kontajnery

Pojmy zavedené v 5. prednáške₍₁₎

- logické výrazy
 - operátory &&, ||, !
- referencie
 - operatory ==, !=
 - hodnota null
- zánik inštancie
 - zánik pri kompozícii
 - zánik pri asociácii

Pojmy zavedené v 5. prednáške₍₂₎

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

Cieľ prednášky

- skupiny objektov – kontajnery
- nový prvok algoritmu – cyklus

- príklad: poznámkový blok

Jednoduché a zložené objekty

- jednoduché objekty
 - atribúty, parametre – primitívne typy
- zložené objekty
 - kompozícia – celok a časť
 - digitálne hodiny
 - kontajnery – vytváranie skupín prvkov jedného druhu
 - môže obsahovať ľubovoľný počet prvkov

Príklady kontajnerov

- zoznam študentov v študijnej skupine
- katalóg kníh v knižnici
- cestujúci v trolejbuse
- index – zoznam zapísaných predmetov
- zoznam klientov banky
- diár – zoznam poznámok

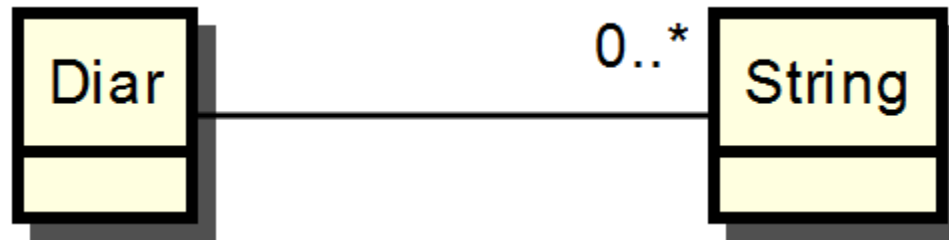
Poznámkový blok – Diár

- kontajner na poznámky
 - poznámka – text (reťazec)
- služby:
 - nové poznámky – doplnenie
 - nepotrebné poznámky – škrtnutie
 - vypísanie zvolenej poznámky
 - počet všetkých poznámok

Diár

```
+ new(): Diar  
+ vlozPoznamku(paPoznamka: String): void  
+ vypisPoznamku(paPoradoveCislo: int): void  
+ zmazPoznamku(paPoradoveCislo: int): void  
+ dajPocetPoznamok(): int
```

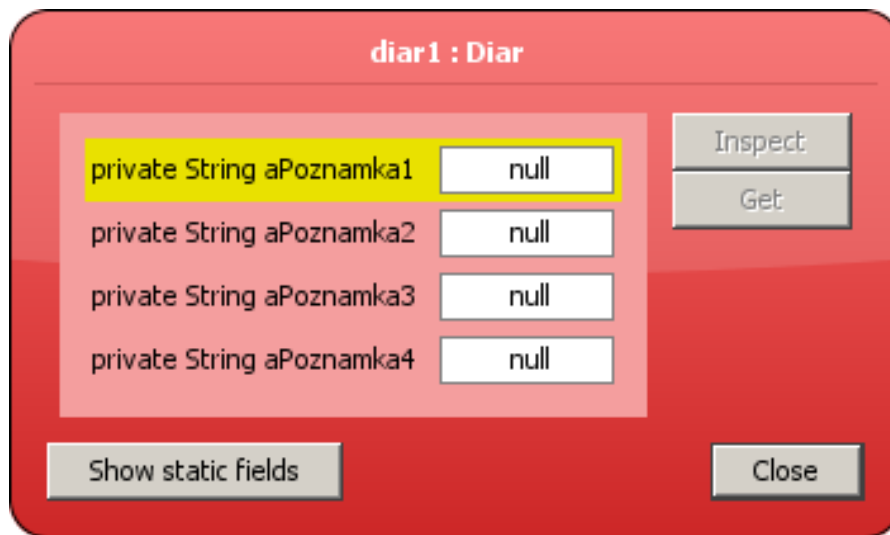

Diár – vzťah s poznámkou



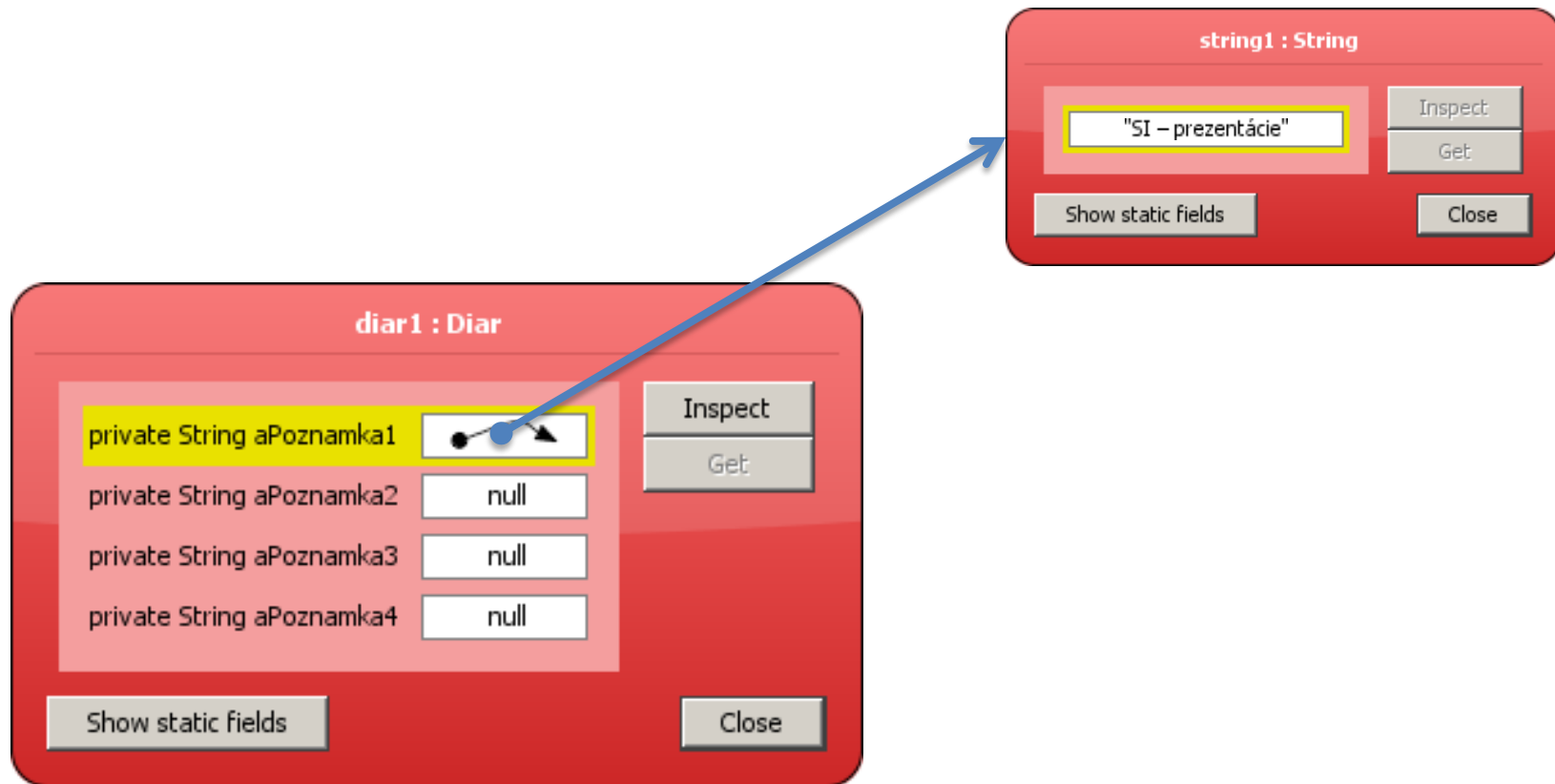
Diár – s doterajšími prostriedkami

- každá poznámka – atribút
- pevne daný počet poznámok
- voľná poznámka == null

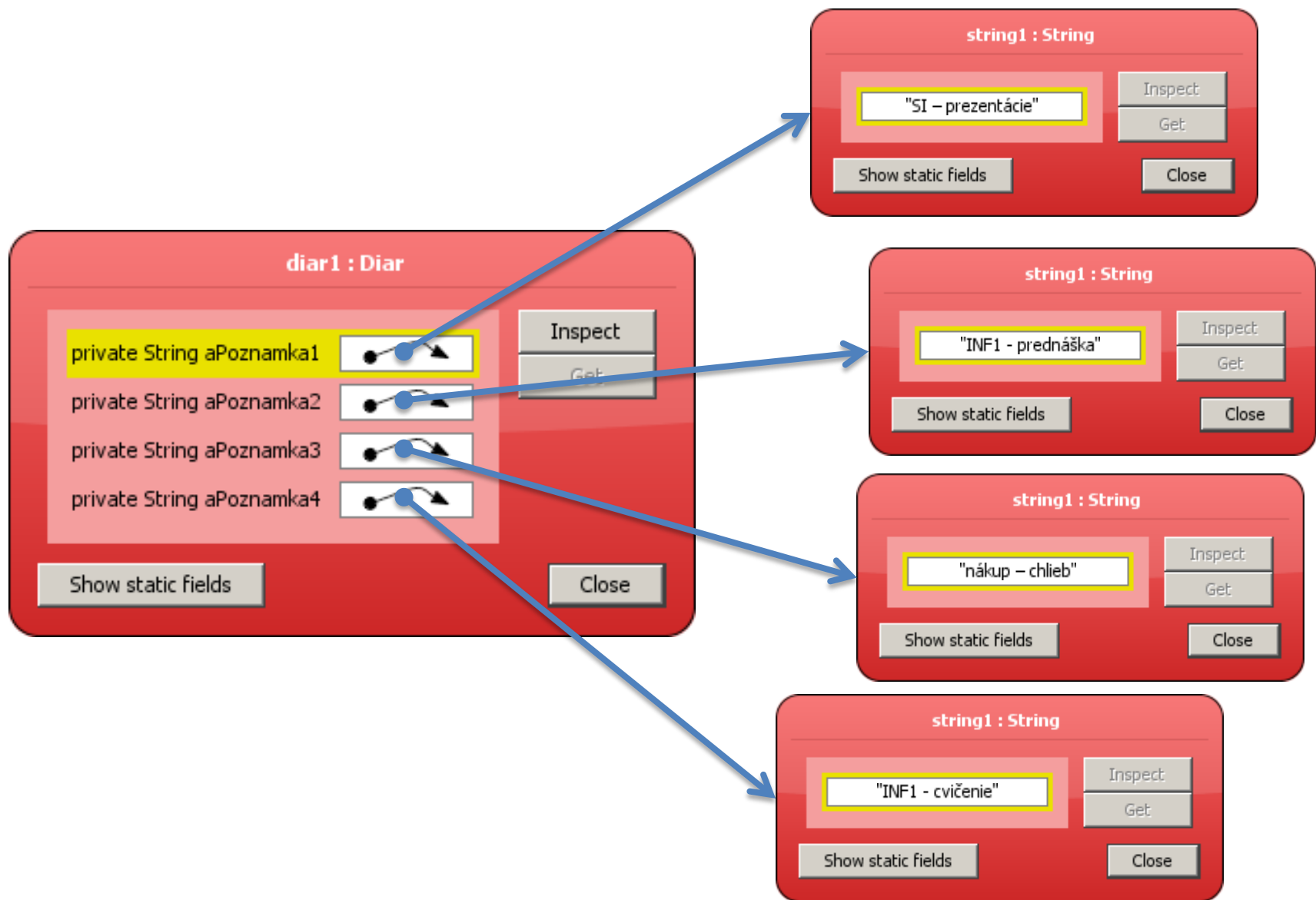
Prázdný diár – objektový diagram



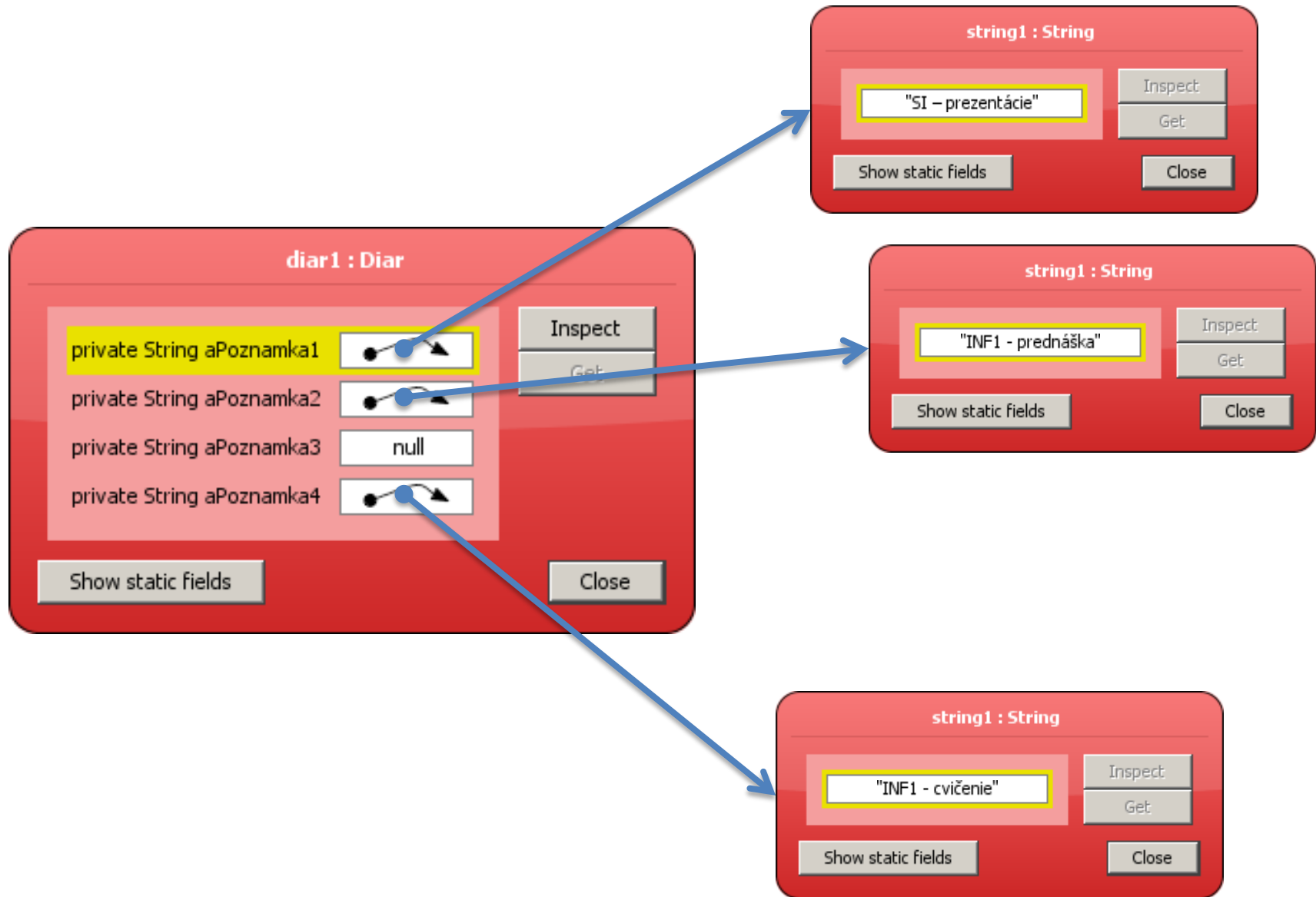
Diár s 1. poznámkou



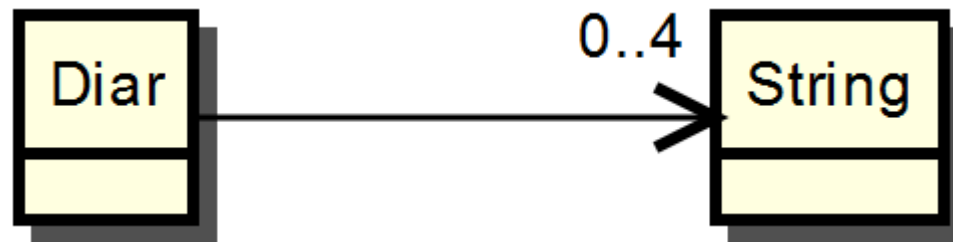
Diár so 4 poznámkami - plný



Diár po vymazaní tretej poznámky



Diár – diagram tried



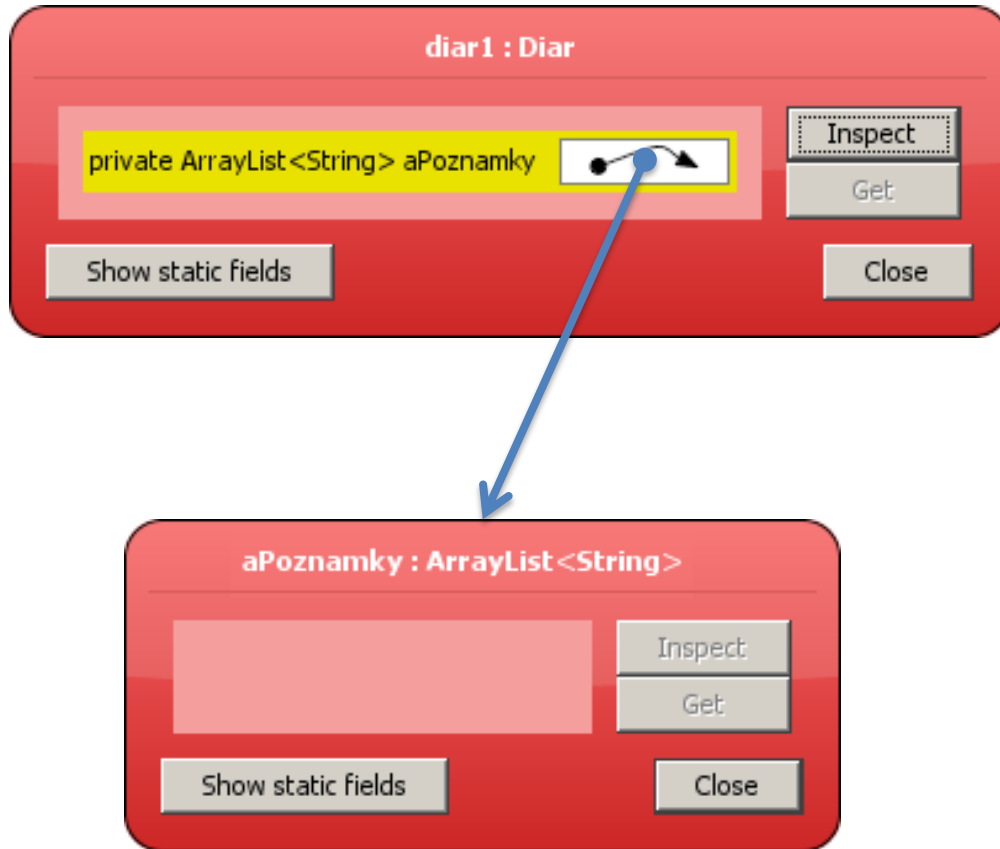
Diár – ďalšie podmienky

- neurčená horná hranica
- počet poznámok sa mení
- počet je ľubovoľný – aj žiadna

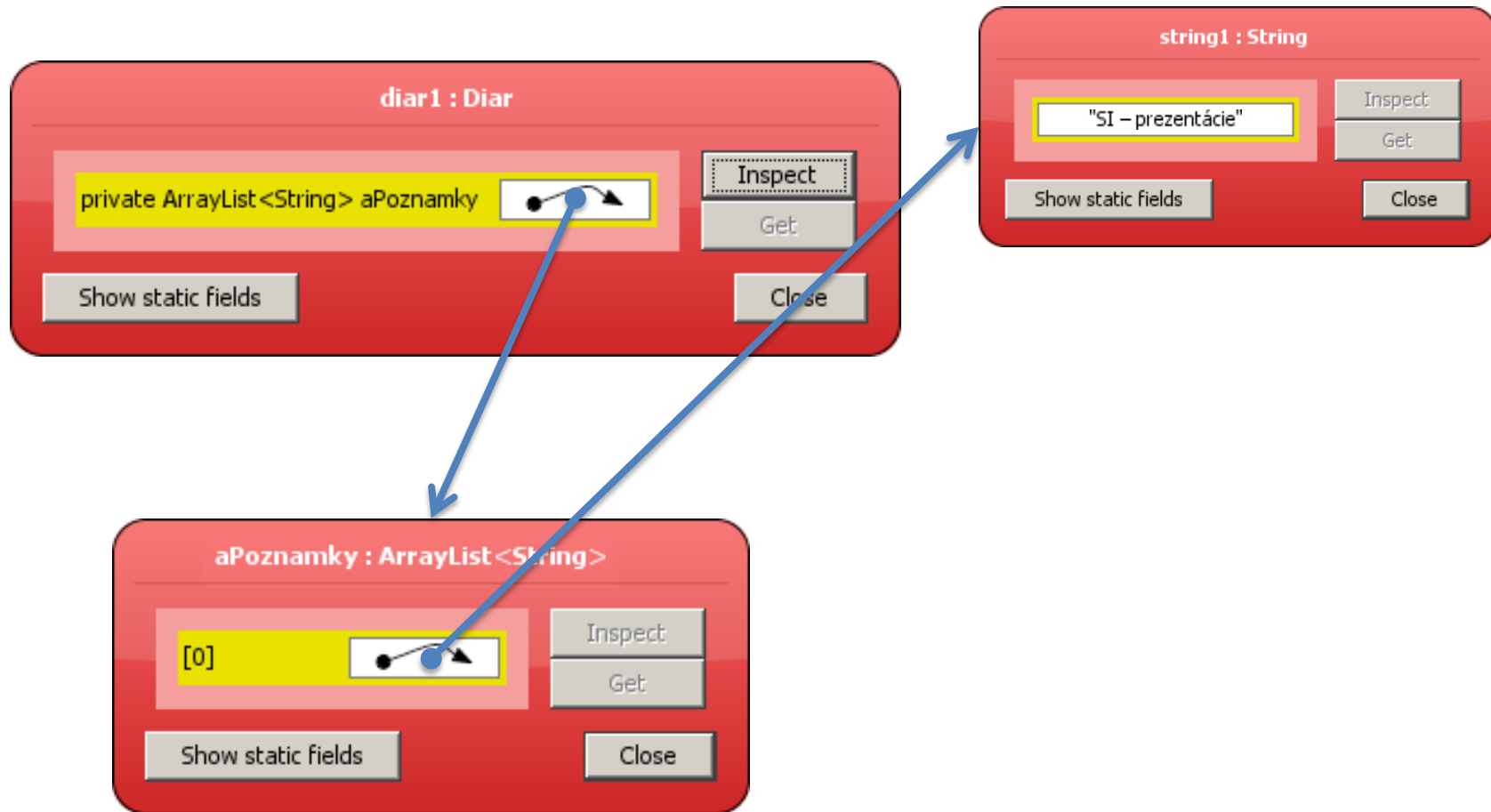
ArrayList – Java

- `ArrayList<TypPrvkov>`
 - `TypPrvkov` – ľubovoľný objektový typ
- kontajner pre zvolený typ prvkov
- môže obsahovať ľubovoľný počet prvkov
- ktorýkoľvek prvok sa dá vymazať
- ku prvkom sa dá pristupovať pomocou poradového čísla – indexu
 - číslovanie začína od 0

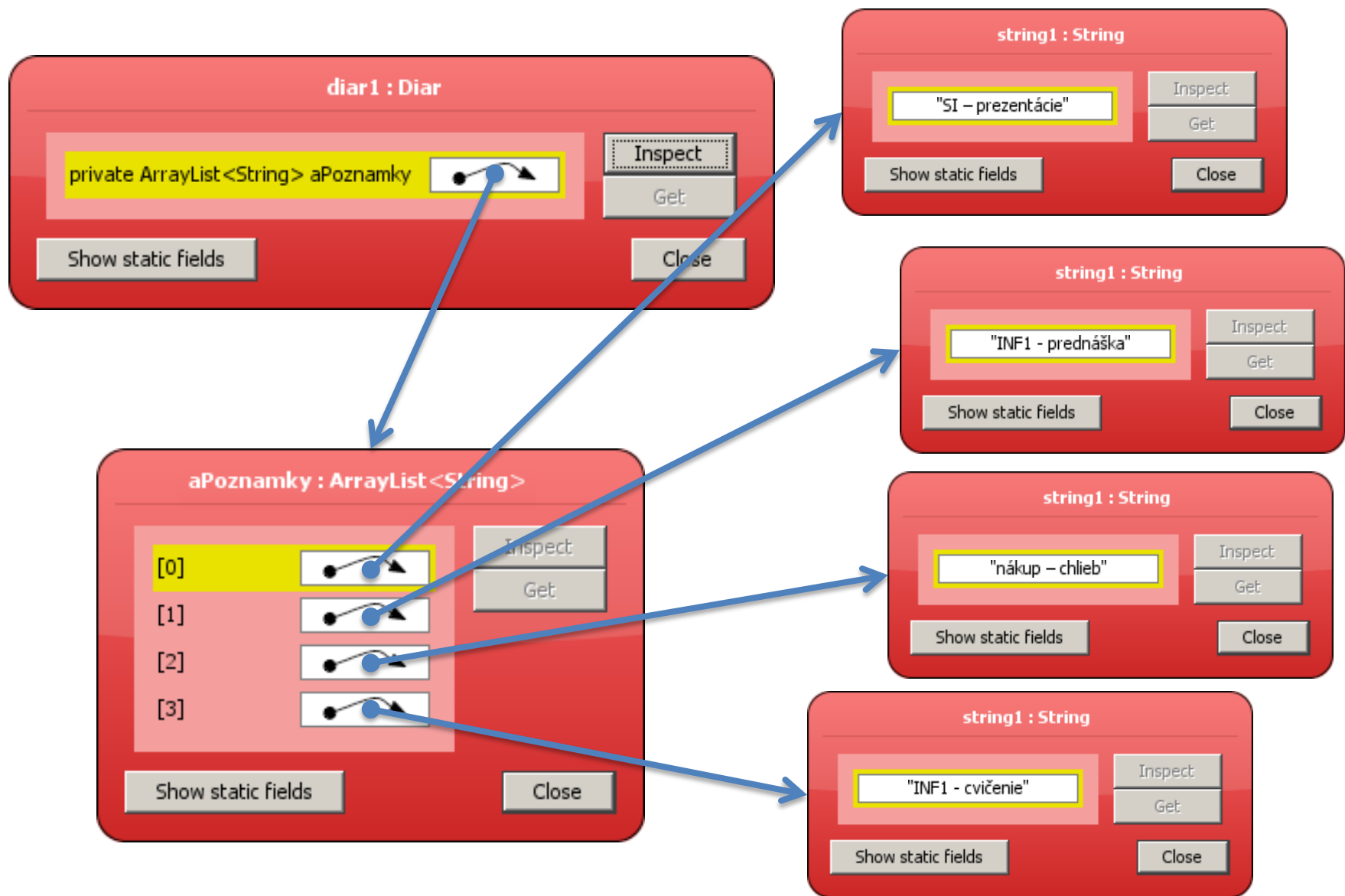
Prázdný diár



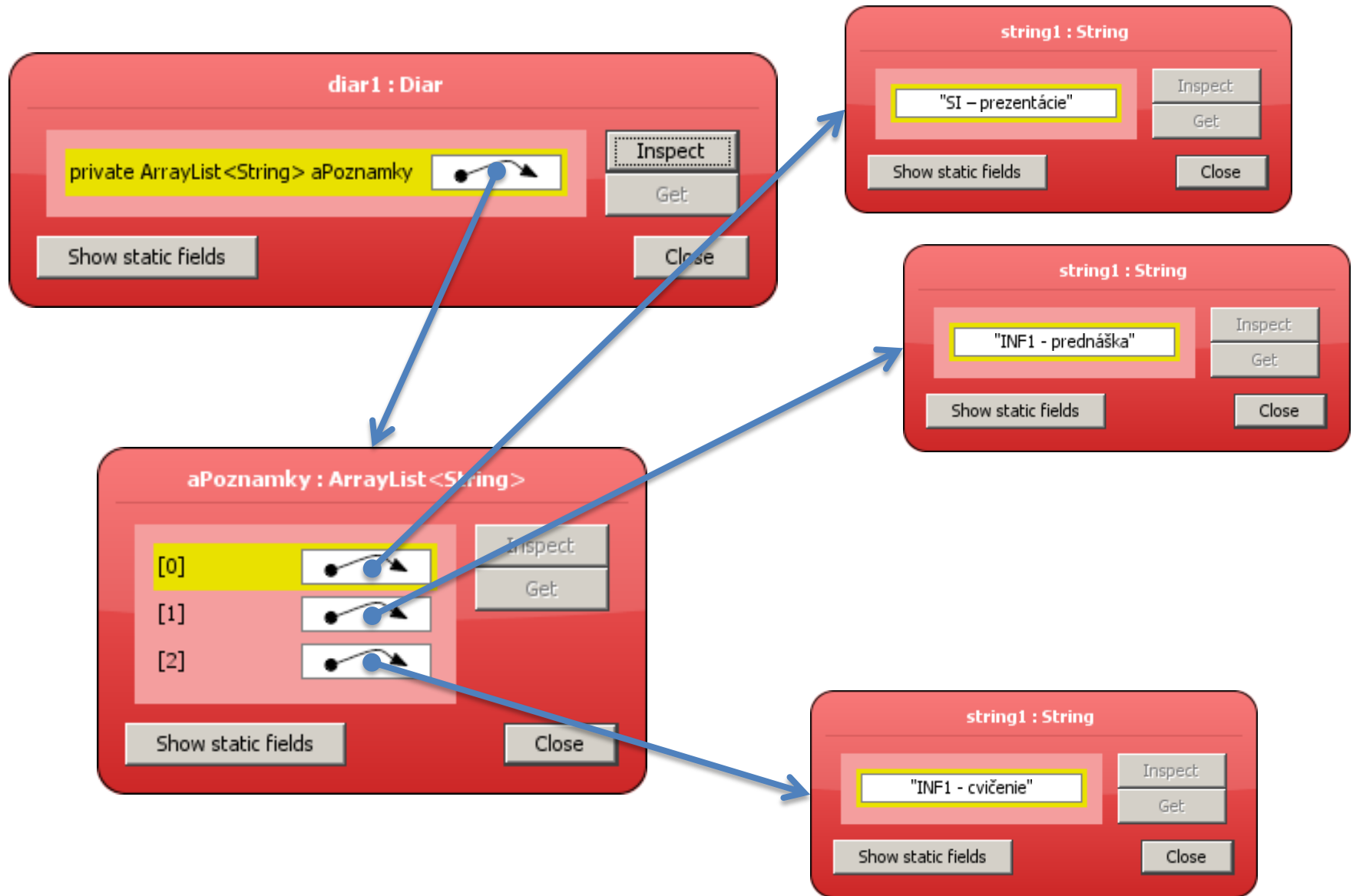
Diár s jednou poznámkou



Diár so 4 poznámkami



Diár po vymazaní tretej poznámky



ArrayList<String> – rozhranie

ArrayList<String>

+ new(): ArrayList<String>
+ add(paPrvok: String): boolean
+ get(palIndex: int): String
+ remove(palIndex: int): String
+ size(): int

Generické triedy – Java

- ArrayList je jedna z generických tried

NazovTriedy<zoznamParametrov>

- parametrami musia byť objektové typy
 - typové parametre

Generické triedy – Java

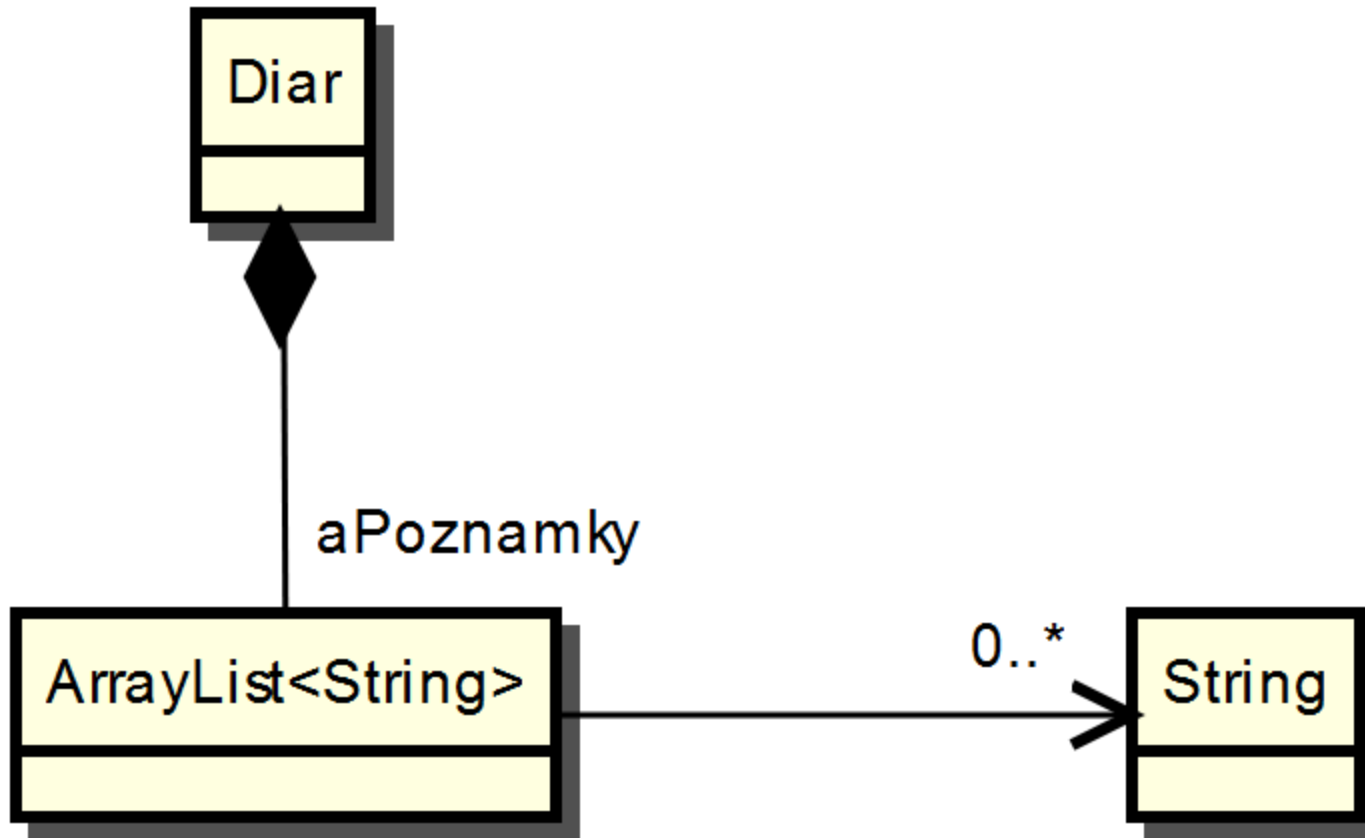
- deklarácia premennej:

```
NazovTriedy<zoznamParametrov> premenna;
```

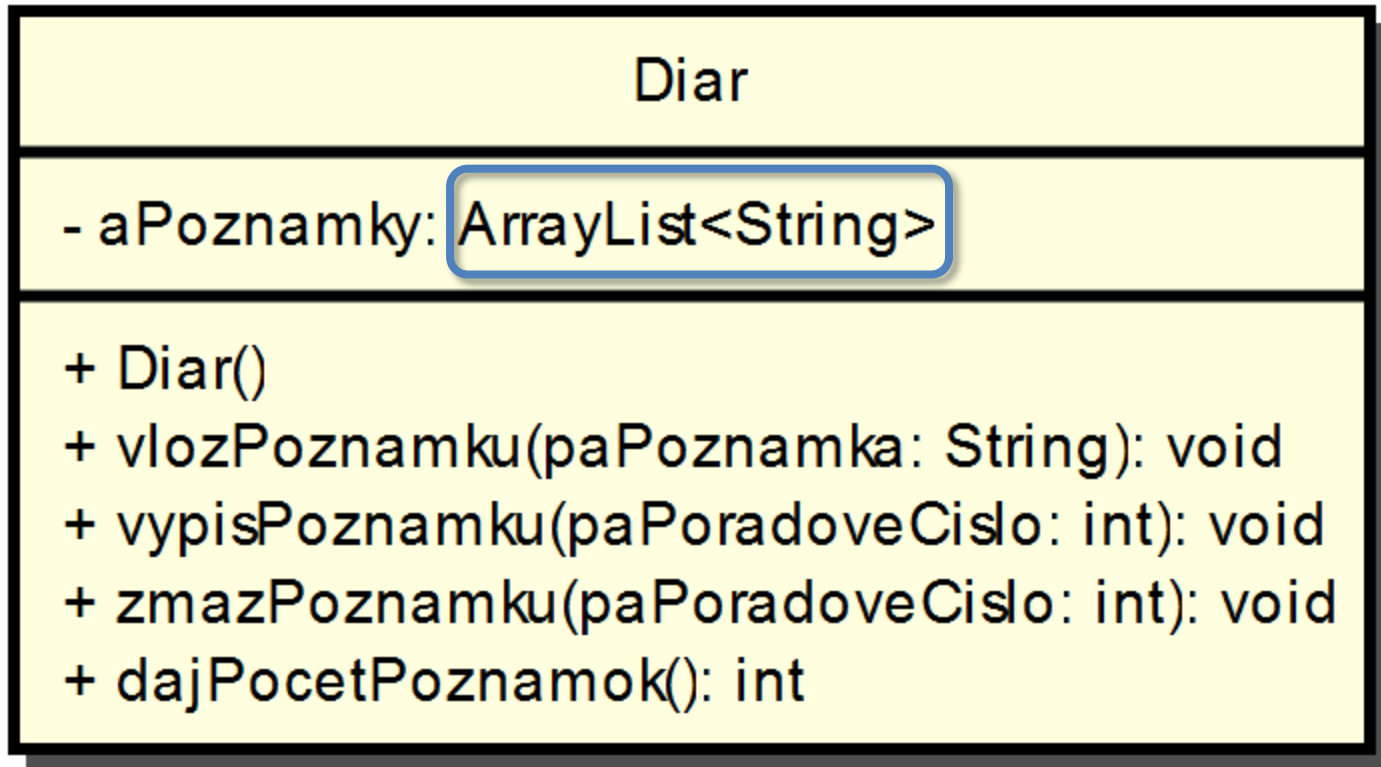
- vytvorenie inštancie:

```
new NazovTriedy<zoznamParametrov>  
(parametreKonstruktora)
```


Diar – diagram tried



Diár – vnútorný pohľad



Diár – definícia triedy, atribút

```
public class Diar
{
    private ArrayList<String> aPoznamky;

    ...
}
```

Diár – konštruktor

```
public Diar()  
{  
    aPoznamky = new ArrayList<String>();  
}
```

Diár – vloženie poznámky

```
public void vložPoznamku(String paPoznamka)
{
    aPoznamky.add(paPoznamka);
}
```

Diár – zobrazenie poznámky

```
public void zobrazPoznamku(int paPoradoveCislo)
{
    if ((paPoradoveCislo >= 0)
        && (paPoradoveCislo < dajPocetPoznamok())) {
        System.out.println(aPoznamky.get(paIndex));
    }
}
```

Diár – počet poznámok

```
public int dajPocetPoznamok()  
{  
    return aPoznamky.size();  
}
```

Diár – zmazanie poznámky

```
public void zmazPoznamku(int paPoradoveCislo)
{
    if ((paPoradoveCislo >= 0)
        && (paPoradoveCislo < dajPocetPoznamok())) {
        aPoznamky.remove(paIndex);
    }
}
```


Zobrazenie všetkých poznámok

- dokážeme zobrazíť ktorúkoľvek poznámku
- ďalšia požiadavka:
 - výpis všetkých poznámok do okna terminálu
- výpis musíme teda postupne zopakovať pre každú poznámku v kontajneri

Diár – zobraz všetky poznámky

```
public void zobrazVsetko()  
{  
    System.out.println(aPoznamky.get(0));  
    System.out.println(aPoznamky.get(1));  
    System.out.println(aPoznamky.get(2));  
    System.out.println(aPoznamky.get(3));  
}
```

- zopakovanie časti algoritmu podľa zadaných pravidiel
 - vypísanie všetkých poznámok v diári
 - sčítanie čísel od 1 po dané číslo
 - vyhľadanie knihy v knižnici
- rôzne typy cyklov – rôzne pravidlá
- pravidlá sa vyhodnocujú počas vykonávania algoritmu procesorom

Cyklus foreach

- jeden z cyklov
- pravidlo:
 - vykonaj pre každý prvok kontajnera

```
for (TypPrvku prvok : kontajner) {  
    // telo cyklu  
}
```

- !!! kontajner sa nesmie v tele cyklu meniť

Diár – výpis všetkých poznámok

```
public void zobrazVsetko()  
{  
    for (String aktualna : aPoznamky) {  
        System.out.println(aktualna);  
    }  
}
```

Rozšírenie možností diára

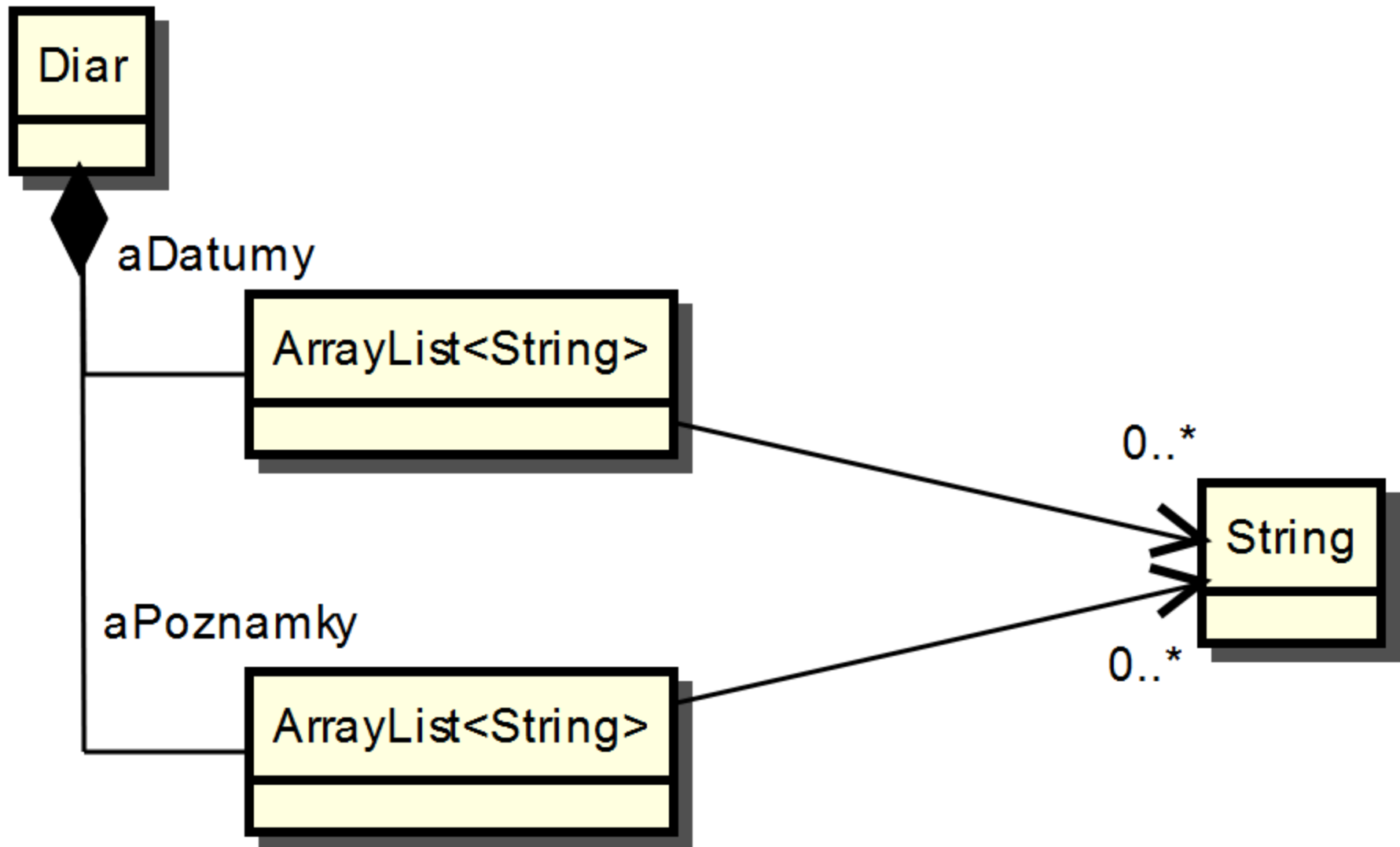
- aktuálna verzia – poznámka = len text
- požiadavka – ukladanie dátumu v poznámke

Nový diár - rozhranie

Diar

- + new(): Diar
- + vlozPoznamku(paDatum: String, paText: String): void
- + vypisPoznamku(paPoradoveCislo: int): void
- + zmazPoznamku(paPoradoveCislo: int): void
- + dajPocetPoznamok(): int
- + zobrazVsetko(): void

Nový diár – možné riešenie



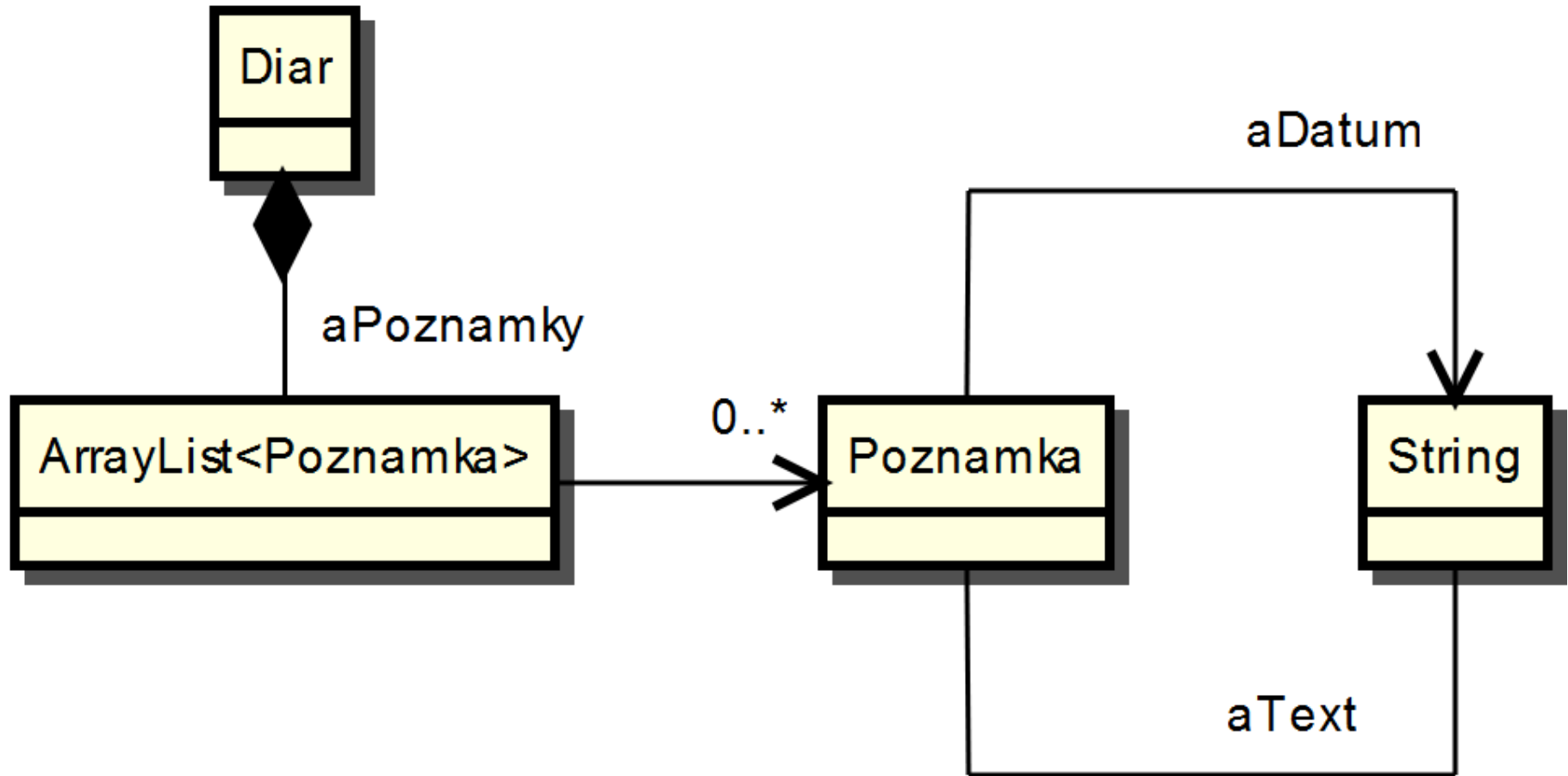
Nový diár – problémy

- samostatné dátumy + samostatné poznámky
- nie je možné jednoducho spárovať dátum a poznámku
- komplikované vypisovanie dátumov a poznámok
- ...

Riešenie – nová trieda Poznamka

- iba jeden zoznam – zoznam poznamok
- každá poznámka obsahuje
 - dátum
 - text

Nový diár – lepšie riešenie



Nový diár – vnútorný pohľad

Diar

- aPoznamky: ArrayList<Poznamka>

+ Diar()

+ vlozPoznamku(paDatum: String, paText: String): void

+ vypisPoznamku(paPoradoveCislo: int): void

+ zmazPoznamku(paPoradoveCislo: int): void

+ dajPocetPoznamok(): int

+ zobrazVsetko(): void

ArrayList<Poznamka> – rozhranie

ArrayList<Poznamka>

+ new(): ArrayList<Poznamka>
+ add(paPrvok: Poznamka): boolean
+ get(paIndex: int): Poznamka
+ remove(paIndex: int): Poznamka
+ size(): int

Trieda Poznamka – rozhranie

Poznamka

+ new(paDatum: String, paText: String): Poznamka
+ dajDatum(): String
+ dajText(): String
+ zobrazPoznamku(): void

Nový diár – definícia triedy, atribút

```
public class Diar
{
    private ArrayList<Poznamka> aPoznamky;

    ...
}
```

Nový diár – konštruktor

```
public Diar()  
{  
    aPoznamky = new ArrayList<Poznamka>();  
}
```


Nový diár – vloženie poznámky

```
public void vlozPoznamku
    (String paDatum, String paText)
{
    Poznamka poznamka =
        new Poznamka(paDatum, paText);
    aPoznamky.add(poznamka);
}
```

Nový diár – vloženie poznámky

```
public void vložPoznamku
```

```
(String paDatum, String paText)
```

```
{
```

```
aPoznamky.add(new Poznamka(paDatum, paText));
```

```
}
```

anonymný objekt



Anonymný objekt

- anonymná inštancia
 - nemá názov \Rightarrow nie je priradená do premennej
- Použitie:
 - ako parameter správy
 - ako adresát správy

Anonymný objekt

```
public void vložPoznamku  
    (String paDatum, String paText)  
{  
    aPoznamky.add(new Poznamka(paDatum, paText));  
}
```

anonymný objekt ako parameter správy

Nový diár – zobrazenie poznámky

```
public void zobrazPoznamku(int paPoradoveCislo)
{
    if ((paPoradoveCislo >= 0)
        && (paPoradoveCislo < dajPocetPoznamok())) {
        aPoznamky.get(paIndex).zobrazPoznamku();
    }
}
```

Anonymný objekt

```
public void zobrazPoznamku(int paPoradoveCislo)
{
    if ((paPoradoveCislo >= 0)
        && (paPoradoveCislo < dajPocetPoznamok())) {
        aPoznamky.get(paIndex).zobrazPoznamku();
    }
}
```

anonymný objekt ako adresát správy

Nový diár – výpis všetkých poznámok

```
public void zobrazVsetko()
{
    for (String aktualna : aPoznamky) {
        aktualna.zobrazPoznamku();
    }
}
```

Ďalšia požiadavka

- teraz môžeme škrtiť záznamy po jednom
- vymazanie celého obsahu diára
 - vytrhnutie popísaných strán
 - práca pokračuje s prázdnyim diárom

- pre všetky položky diára (poznámky):
 - vymaž položku zo zoznamu

Nový diár – zmaž všetky poznámky

```
public void zmazVsetko()
{
    for (String aktualna : aPoznamky) {
        aPoznamky.remove(aktualna);
    }
}
```

Chyba pri mazaní

```
72  
73     public void zmazVsetko()  
74     {  
75         for (Poznamka aktualna : aPoznamky) {  
76             aPoznamky.remove(aktualna);  
77         }  
78     }  
79 }  
80
```

java.util.ConcurrentModificationException:
null (in java.util.AbstractList\$Itr)

saved

kontajner sa nesmie v tele cyklu meniť

Cyklus while

- Další z cyklov
- Pravidlo:
 - vykonávaj telo cyklu kým platí podmienka

```
while (podmienka) {  
    // telo cyklu  
}
```

Nový diár – zmaž všetky poznámky

```
public void zmazVsetko()
{
    while (aPoznamky.size() > 0) {
        aPoznamky.remove(0);
    }
}
```

Vďaka za pozornosť