



Prednáška 10: Základné analytické postupy na vytváranie objektovo-orientovaného modelu

Ján Lang

kanc. 4.34, jan.lang@stuba.sk, <http://www2.fiit.stuba.sk/~lang/zoop/>

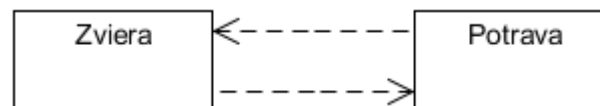
Ústav informatiky, informačných systémov a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave
5. decembra 2023

Polymorfizmus

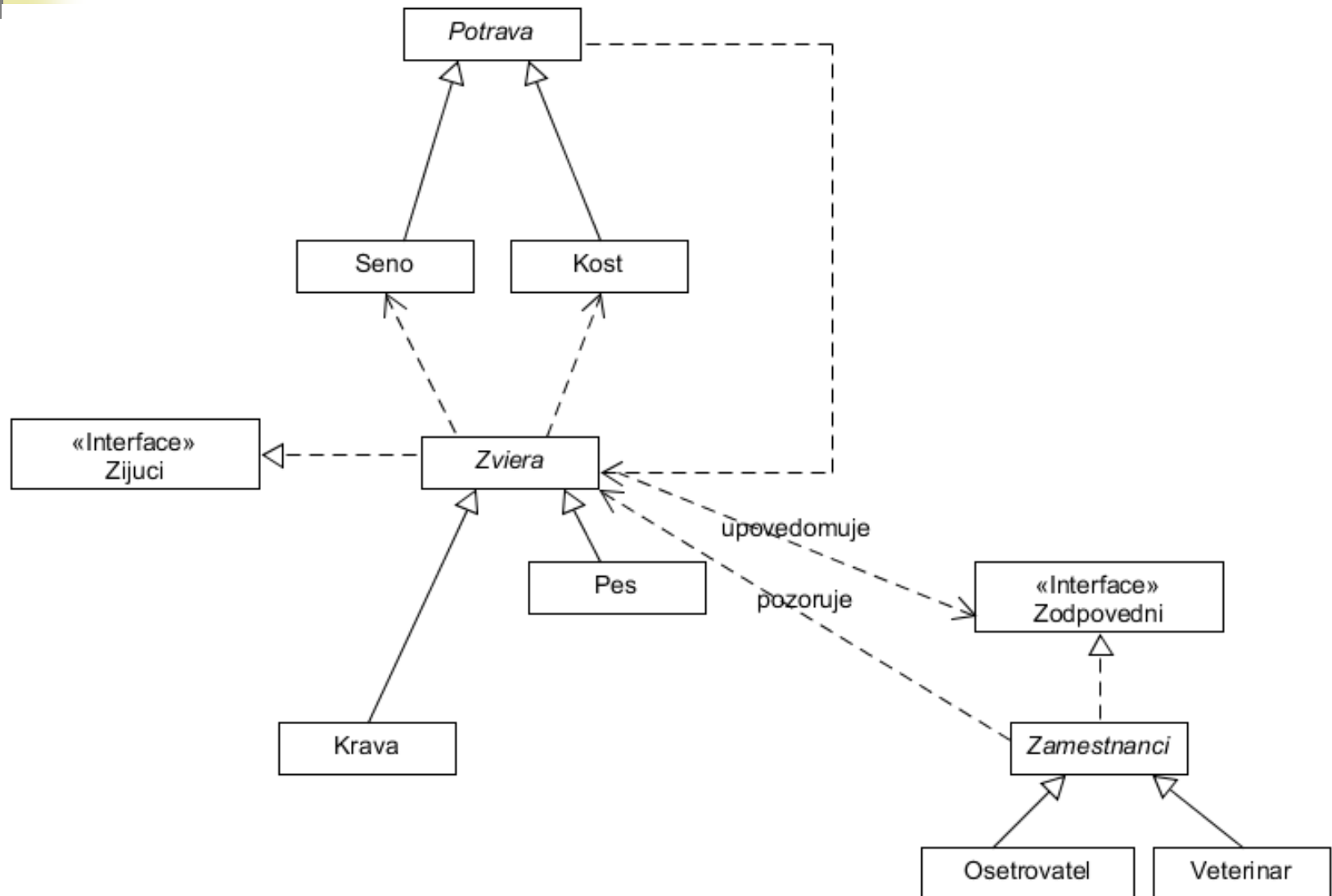
- Symetria vzájomnej závislosti

```
public static void main(String[] args) {  
    new Zviera().zjedz(new Potrava());  
    new Potrava().nakrm(new Zviera());  
}
```

```
<terminated> Hlavna (22) [Java Application] C:\H  
Som zviera a potrava mi chuti  
Zviera z je nakrmene
```



Prečo je tá krava tak dôležitá?





Rozhrania

- Skúsme pridať novú metódu do existujúceho rozhrania bez zásahu do existujúcich tried, ktoré ho implementujú



Rozhrania

- Skúsme pridať novú metódu do existujúceho rozhrania bez zásahu do existujúcich tried, ktoré ho implementujú
- **public interface I**
- **public void m();**
- ...tiež
- **public class A implements I**
- **public class B implements I**



Rozhrania

- Skúsme pridať novú metódu do existujúceho rozhrania bez zásahu do existujúcich tried, ktoré ho implementujú
- **public interface I**
- **public void m();**
- ...tiež
- **public class A implements I**
- **public class B implements I**

Existujúci
systém



Rozhrania

- Skúsme pridať novú metódu do existujúceho rozhrania bez zásahu do existujúcich tried, ktoré ho implementujú
- ...čiže máme prístup k rozhraniu I
- ALE nemáme dosah na triedy A, B
- **public void newMethod();**
- = problém...



Rozhrania

- Skúsme pridať novú metódu do existujúceho rozhrania bez zásahu do existujúcich tried, ktoré ho implementujú
- **Modifying one interface in a JDK framework breaks all classes that extend/implement the interface**



Rozhrania

- Skúsme pridať novú metódu do existujúceho rozhrania bez zásahu do existujúcich tried, ktoré ho implementujú

TO ČO NEMÔŽEME

- Modifying one interface in a JDK framework breaks all classes that extend/implement the interface

TO ČO CHCEME

- to extend interfaces in a backward-compatible way.



Rozhrania

- Skúsme pridať novú metódu do existujúceho rozhrania bez zásahu do existujúcich tried, ktoré ho implementujú

SOLUTION

- *Default methods* can be provided to an interface without affecting implementing classes as it includes an implementation
- ... It allows interface to define implementation which will use as the default in a situation where a concrete class fails to provide an implementation for that method



Rozhrania

- Skúsme pridať novú metódu do existujúceho rozhrania bez zásahu do existujúcich tried, ktoré ho implementujú

SOLUTION

- *Default methods* can be provided to an interface without affecting implementing classes as it includes an implementation
- ... It allow interface to define implementation which will use as the default in a situation **where a concrete class fails to provide an implementation for that method**



Rozhrania

- Interface – „Default Method“ (Defender methods) implementation
- Skúsme pridať novú metódu do existujúceho rozhrania bez zásahu do existujúcich tried, ktoré ho implementujú
- Default method v java API 8
- ...napr. v rozhraní - **Interface Iterable<T>**
- <https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html>

All Methods

Instance Methods

Abstract Methods

Default Methods

Modifier and Type

Method and Description

default void

`forEach(Consumer<? super T> action)`

Performs the given action for each element

```
for (T t : this)
    action.accept(t);
```



Abstract class vs. Interface

- ...ako vyzerá rozdiel po zavedení default method v Java 8?
- The abstract class can define constructors. They are more structured and can have a state associated with them.
- While in contrast, *default method* can be implemented only in the terms of invoking other interface methods, with no reference to a particular implementation's state = čo už...



Abstract class vs. Interface

- **Namespace konflikt**
- Dve rovnomené default method-y v dvoch rozhraniach
- Since Java classes can implement multiple interfaces and each interface can define a *default method* with the same method signature, the inherited methods can conflict with each other
- Jedno z riešení, je poskytnúť v triede vlastnú implementáciu default metod-y
- `<interface>.super.defaultMethod();`



Regular vs. default method

- ...ako vyzerá rozdiel po zavedení default method v Java 8?
- *Default Method* is different from the regular *method*
 - × In the sense that *default method* comes with *default* modifier
 - × Additionally, methods in classes can use and modify method arguments as well as the fields of their class, but *default method*, on the other hand, can only access its arguments as interfaces do not have any state



Základné analytické postupy

- Len informatívne na príklade v EA
- Nepotrebuje riešiť do svojho zadania, nie je podmienkou na uplatnenie vo vašom zadaní!



TODO nezabudnite

- Aj vy môžete pomôcť vylepšiť tento predmet študentom pre nasledujúci akademický rok. Vaše odporúčanie, komentár či otázka.
...cez spätnoväzobný formulár.