

## **Správa o realizácii projektu**

**URBARIUM** - správa finančných tokov urbárskeho pozemkového spoločenstva

### **I. Zadanie projektu:**

Projekt je zameraný na vytvorenie systému na organizáciu peňažných operácií v rámci činnosti urbárskeho pozemkového spoločenstva (spol. súkromných vlastníkov podielov lesov). Do činnosti spoločenstva spadá: ťažba drevnej hmoty, zalesňovanie, čistenie pasienkov, oprava ciest, prenájom pozemkov, nákup strojov, materiálu a nehnuteľností.

Ďalej systém ponúka riadenie služieb spoločenstva pre verejnosť - medzi ktoré patrí predaj a doprava drevnej hmoty (guľatina, stolárske, vlákňinové, palivové drevo), výdaj povoleniek pre súkromnú ťažbu dreva.

Systém je navrhnutý pre užívateľov s rôznymi funkciami a prístupovými právami. Sú to:

*Zákazník* – využíva služby spoločenstva pre verejnosť: nákup a doprava rôzneho druhu dreva, žiadosti o povolenky na ťažbu dreva.

*Podielnik* – člen spoločenstva, vlastník určitej časti pozemku má prehľad o hospodárení spoločenstva, má možnosť využívať služby za zvýhodnené ceny (podľa veľkosti svojho podielu). Môže ponúknuť na predaj svoj podiel ostatným členom spoločenstva.

*Ekonom* – zodpovedá za správu peňažných transakcií, akými sú prerozdeľovanie zisku/straty podielnikom podľa podielu, vyplácanie zamestnancov, vystavovanie faktúr za objednané služby.

*Hospodár* – vybavuje požiadavky zákazníkov a podielnikov, schvaľuje žiadosti o povolenky.

## II. Štruktúra systému vo forme diagramu najdôležitejších tried

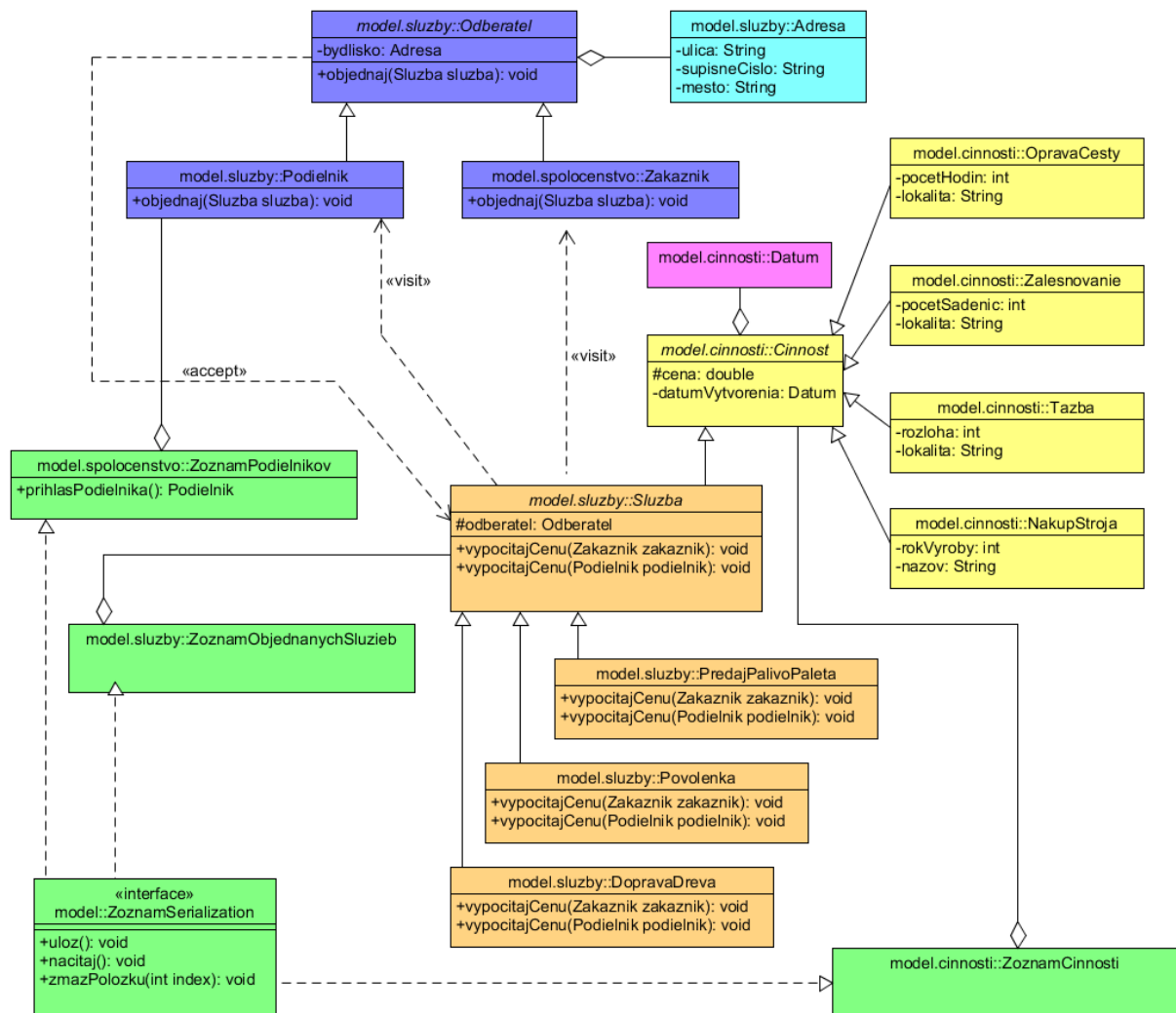


Diagram znázorňuje najdôležitejšiu časť programu použitého návrhového vzoru Visitor.

*Element* = *Odberatel*                      *accept()* = *objednaj()*  
*Visitor* = *Sluzba*                         *visit()* = *vypocitajCenu()*

*Odberatelom* môže byť *Podielnik* alebo *Zakaznik*. Každý agreguje *Adresu*.

*Sluzba* môže byť *PredajPalivoPaleta*, *Povolenka* alebo *DopravaDreva*.

Metóda *Odberateľa* *objednaj()* volá metódu *Sluzby* *vypocitajCenu()*. Algoritmus výpočtu pritom závisí od typu *odberateľa* aj *objednanej služby*.

*Sluzba* je podtypom *Cinnosti* spoločnosti, ktorými sú tiež *OpravaCesty*, *Zalesnovanie*, *Tazba*, *NakupStroja*. Všetky obsahujú *Datum* vytvorenia.

*Zoznamy Podielnikov*, *Cinnosti* alebo *objednávok Sluzby* sú uložené vo vlastných triedach. Ktorým predpisuje správanie *ZoznamSerialization*.

### III. Plnenie kritérií

#### a. Hlavné kritéria:

**Dedenie:** dve základné hierarchie dedenia

*Odberatel*    ◀ *Podielnik*  
                  ◀ *Zakaznik*

*Cinnost*      ◀ *Tazba*  
                  ◀ *Zalesnovanie*  
                  ◀ *OpravaCesty*  
                  ◀ *NakupStroja*  
                  ◀ *Sluzba*                    ◀ *PredajPalivoPaleta*  
  ◀ *DopravaDreva*  
  ◀ *Povolenka*

**Polymorfizmus:** použitie návrhového vzoru *Visitor*. V podtriedach triedy *Odberatel* sa vždy objedná daná služba vykonaním metódy *objednaj()*. A v podtriedach triedy *Sluzba* sa vždy vykoná metóda *vypocitajCenu()*, pričom výpočet prebehne podľa špecifického algoritmu. Možnosť pridania ďalších služieb, bez zmeny súčasných tried.

*Element* = *Odberatel*                    *accept()* = *objednaj()*  
*Visitor* = *Sluzba*                        *visit()* = *vypocitajCenu()*

#### **Agregácia:**

*Odberatel* agreguje *Adresa*  
*Cinnost* agreguje *Datum*  
*Sluzba* agreguje *Odberatel*

**Zapuzdrenie:** napr. trieda *Podielnik*:

```
private int pocetCiastok;                    //hodnotu nie je možné meniť
private String login;                        //hodnotu nie je možné meniť
private String heslo;                        //hodnotu nie je možné meniť
public int getPocetCiastok();                //hodnota je verejná
protected String getLogin();                //hodnota je viditeľná len triede
protected String getHeslo();                zodpovednej za prihlásenie
```

#### b. Ďalšie kritériá:

1. použitie návrhového vzoru *Visitor*
2. ošetrenie mimoriadnych stavov, nachádzajú sa v balíku *vynimky*
3. poskytnutie grafického používateľského rozhrania - vzor *MVC*
4. použitie *AspectJ*. Trieda *LogPodielnik* sleduje prihlásenia *Podielnik*, zoznam prihlásení serializuje a vypisuje v *HlavneOkno* (zasahuje do GUI)
5. použitie *RTTI*. V triede *VyberSluzby* za behu programu zisťuje typ vybraného objektu a vypisuje vhodný popis služby. V triede *VyberCinnosti* za behu programu zisťuje typ vybraného objektu a vypisuje vhodný popis cinnosti.

#### **IV. Zoznam odovzdaných pracovných verzí**

##### **6. cvičenie: - 31.3.2016**

Pridanie:

Triedy odberateľov:

- Zakaznik
- Podielnik

Triedy činnosti spoločnosti:

- Sluzba:
  - PredajDreva
  - DopravaDreva
  - Povolenka
- Tazba
- Zalesnovanie
- NakupStroja
- OpravaCiest

Triedy zoznamov:

- ZoznamPodielnikov
- ZoznamCinnosti
- ZoznamObjednavokSluzieb

GUI :

- HlavneOkno

##### **7. cvičenie: - 7.4.2016**

Pridanie:

Triedy vo view

- OknoHospodar
- OknoPodielnik
- OknoZakaznik
- PridaniePodielnikForm
- PrihlaseniePodielnikForm
- NovaCinnostForm

Triedy v controler

- PrihlaseniePodilnika
- PridaniePodielnik
- NacitanieZoznamov
- UlozenieZoznamov
- VyberCinnosti
- VyberSluzby

##### **9. cvičenie: - 21.4.2016**

Trieda Aspect : LogPodielnik

//logovanie Podielnika: datum+login

implementácia návrhového vzoru Visitor

Element = Odberateľ accept() = objednaj()  
Visitor = Sluzba visit() = vypocitajCenu()

##### **10. cvičenie: - 5.5.2016**

Implementácia vytvorenie novej činnosti Hospodárom

##### **11. cvičenie: - 12.5.2016**

- použitie reflexivnosti- vytvorenie objektu novej činnosti za behu programu na základe vybraného typu objektu
- zmena triedy LogPodielnik, pridanie serializácie a zásahu aspektu aj do GUI

##### **12. cvičenie: - 16.5.2016**

- pridanie RTTI pri výbere služieb a činnosti v triedach: VyberCinnosti, VyberSluzby
- pridanie výpočtu zostatku na účte a zobrazenia v okne EkonomOkno