



ZÁPADOČESKÁ  
UNIVERZITA  
V PLZNI

# **KIV/PIA - Samostatná práce**

## **Modul zkrstag pro OpenCMS**

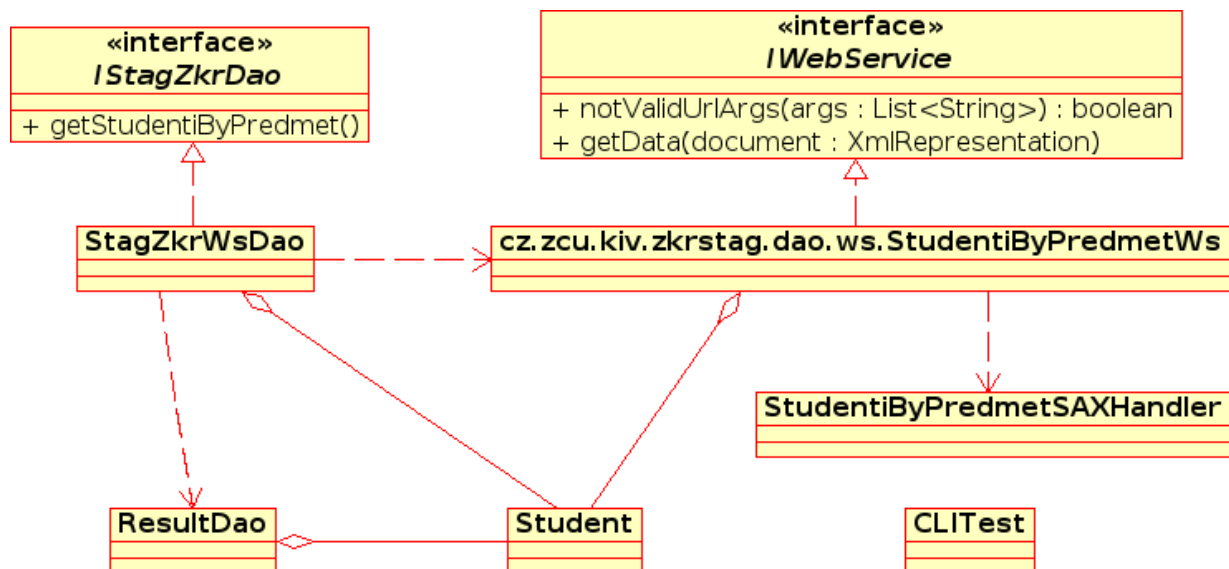
**Martin Žibrický (A07077)**  
matesz@students.zcu.cz

# Obsah

1	Programátorská dokumentace.....	3
1.1	Diagram tříd knihovny p-sim.....	3
1.2	Vytvořená síť ze simulačních objektů.....	4
1.3	Definované objekty.....	4
1.4	Popis procesů.....	5
1.4.1	ReqSource.....	5
1.4.2	Server.....	5
2	Výsledky.....	5
3	Uživatelská dokumentace.....	6
3.1	Instrukce ke spuštění.....	6
3.2	Ukázkový konzolový výstup aplikace.....	6
4	Závěr.....	6
4.1	Nedostatky.....	6
4.2	Zhodnocení.....	6

# 1 Programátorská dokumentace

## 1.1 Diagram tříd



Ilustrace 1.1: Diagram Tříd

## 1.2 Rozhraní a Třídy

Typ	Popis
IStagZkrDao (interface)	Rozhraní pro přístup k datům ze studijní agendy
IWebService (interface)	Rozhraní definující metody pro validaci parametrů webové služby a zpracování XML odpovědi webové služby. Rozhraní je nutno implementovat pro každou webovou službu STAGu.
StagZkrWsDao	Implementace rozhraní IStagZkrDao
StudentiByPredmetWs	Implementace rozhraní IWebService pro webovou službu <code>getStudentiByPredmet</code> .
Student	Třída reprezentující data pro jednoho studenta.
StudentiByPredmetSAXHandler	SAX parser pro web service <code>getStudentiByPredmet</code>
ResultDao	Typ objektu, který získá webový frontend, volající danou web service
CLITest	Pomocná třída pro testování modulu <code>cz.zcu.kiv.zkrstag</code> . Testování probíhá výpisy na konzoli.

## 1.3 Princip fungování

- Modul využívá pro komunikaci s webovými službami STAGu knihovnu *Restlet* ([www.restlet.org](http://www.restlet.org)).
- Modul umožňuje vytvořit jen jednu instanci třídy `StagZkrWsDao`. Je to takto řešeno proto, poněvadž je zajištěno, že webový server KIVu bude vytvářet jen jedno spojení k serveru STAGu. Předpokladem je, že modul nebude využívat mnoho uživatelů současně a mělo by

- se tudíž vystačit s jedním spojením k serveru studijní agendy.
- Rozlišují se 2 druhy chyb:
  - chyba spojení (connError)
    - Jsou možné dvě situace, kdy tato chyba
  - chyba vstupu (inputError)
    - chyba

Všechny vyjímky jsou odchyceny v *Dao* a pokud nastane nějaká ze dvou vyjímek, tak se v objektu typu *ResultDao* nastaví atributy *connError* případně *inputError* na hodnotu *true*.

## 2 Uživatelská dokumentace

### 2.1 Postup zprovoznění

Jako hlavní inspirace by měli sloužit třídy *StagZkrWsDao* a *StudentiByPredmetWs*. Tyto třídy implementují potřebná rozhraní pro webovou službu *getStudentiByPredmet*.

Postup bude popisován pro webovou službu *getStudentiByPredmet*. Předpokládejme, že web service se jmenuje *getStudentiByPredmet*.

1. Na stránce *stag-ws.zcu.cz* si vybereme službu typu REST a najdeme její cestu na serveru STAGu.
2. Cestu přidáme do souboru *zkrstag.properties*:  
`ws.path.StudentiByPredmet = /ws/services/rest/student/getStudentiByPredmet`
3. Do *IStagZkrDao* a *StagZkrWsDao* přidáme metodu (parametry odpovídají parametrům URL):  
`public ResultDao getStudentiByPredmet(String katedra, String zkratka, String rok, String semestr);`
4. Vytvoříme třídu *StudentiByPredmetWs*, která bude implementovat rozhraní *IWebService*.
5. Implementace metody *getStudentiByPredmet* spočívá v:
  1. inicializace proměnné s výsledky
  2. přidání všech parametrů pro web service do kolekce typu *List<String>*

```
List<String> args = new ArrayList<String>();
```

```
args.add(katedra);
```

```
args.add(zkratka);
```

```
args.add(rok);
```

```
args.add(semestr);
```

3. vytvoření objektu typu *IWebService*

```
IWebService ws = new StudentiByPredmetWs();
```

4. validace parametrů

```
if (notValidArgs(ws, args))
```

```
return this.result;
```

5. spojení parametrů do jednoho řetězce ve tvaru:

```
String query = "katedra=" + katedra;
```

```
query += "&zkratka=" + zkratka;
```

```
query += "&rok=" + rok;
```

```
query += "&semestr=" + semestr;
```

6. získání path k webové službě ze souboru *zkrstag.properties*

```
updateUrl("StudentiByPredmet", query);
```

7. vytvoření spojení se serverem studijní agendou

```
if (notValidConnection())
```

```
return this.result;
```

8. získání potřebné reprezentace xml a parsování xml

- je potřeba dávat pozor na to, jestli chceme SAX

```
if (notValidParsing(ws, response.getEntityAsSax()))
```

- `return this.result;`
  - nebo DOM reprezentaci
    - `if (notValidParsing(ws, response.getEntityAsDom()))`
- `return this.result;`
6. Dále je potřeba provést ve třídě *StudentiByPredmetWs* implementaci odpovídajících metod.
  7. V neposlední řadě vytvoříme jsp stránky:
    - přidáme potřebná vstupní políčka (formulář) do souboru *zkratky.jsp*
    - vytvoříme obsluhu formuláře podle vzoru souboru *studenti\_predmetu-seznam.jsp*

## 3 Závěr

### 3.1 Známé nedostatky

- Vytvořený modul nemá implementováno nastavení času (*timeout*), po kterém se spojení se serverem STAGu považuje za nerealizovatelné. Pokud se nelze spojit se serverem STAGu, tak bude klient čekat donekonečna na navázání spojení!
- Chybí podpora lokalizace.
- Automatický výber semestru ZS/LS v hlavním formuláři podle aktuálního kalendářního data není implementován.
- Implementována jen jedna služba STAGu.

### 3.2 Zhodnocení

Navrhl jsem a realizoval čtení dat z jedné webové služby STAGu. Načítání seznamu studentů na předmětu funguje bez větších problémů.

V současném stavu není modul ještě připraven pro produkční nasazení. Hlavním důvodem je v současné době absence práce s nastavením timeoutu. Při dalším vývoji bude toto hlavní bod, kterým je potřeba se zabývat.

Pokus o vytvoření modulu pro redakční systém OpenCMS byla zajímavá, a věřím že i užitečná, zkušenost. Zejména přínosem bylo oživení a rozšíření znalostí v oblasti J2EE. Pokud mi to čas dovolí, chtěl bych dopracovat započatý modul do finální podoby.