

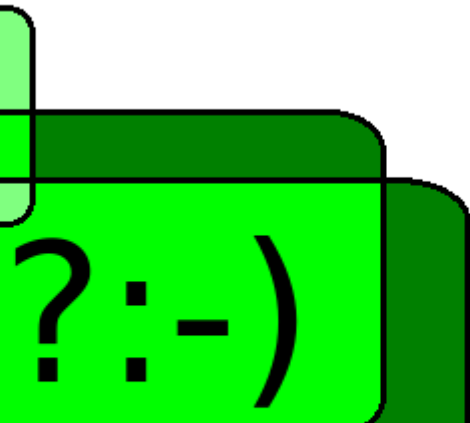
Deklarativno programiranje



Laboratorijske vježbe

Izv. prof. dr. sc. Markus Schatten

Prezentacija #9



Ažuriranje baze znanja

- U Flora-2 ažuriranje se ostvaruje primitivima:
 - **insert{ ... }, insertall{ ... }**
 - **delete{ ... }, deleteall{ ... }**
 - **erase{ ... }, eraseall{ ... }**
 - **insertrule{ ... }, deleterule{ ... }**
- **{ ... }** ima oblik **{ izraz | upit }** pri čemu je **izraz** izraz koji će se ubaciti/izbrisati iz baze znanja, a **upit** opcionalni upit koji taj izraz mora zadovoljavati

Upute

- Naredbe u nastavku upisujete izravno u konzolu.
- Kopiju konzole stavite u datoteku ime_prezime.txt

Primjeri

```
insert{ marica:osoba }.
```

```
?x:osoba.
```

```
insert{ marica:osoba@gol }.
```

```
?x:osoba@?modul.
```

```
delete{ marica:osoba }.
```

```
?x:osoba.
```

```
insertrule{ (?x:covjek :- ?x:osoba)@gol }.
```

```
?x:covjek@?modul.
```

Dinamički moduli

```
newmodule{ go3 }.
```

Dinamički moduli

```
newmodule{ go3 }.
```

```
insert{ ?x:osoba@go3 | ?x:osoba@go1 }.
```

Dinamički moduli

```
newmodule{ go3 }.
```

```
insert{ ?x:osoba@go3 | ?x:osoba@go1 }.
```

```
?x:osoba@go3.
```

Perzistentni moduli

- Kako bi sačuvali podatke u Flora-2 koristi se integrirani modul **persistentmodules**
- Omogućava nam spajanje željenog modula na (relacijsku) bazu podataka u kojoj se pohranjuju podaci modula

Priprema – ODBC SQLite

Provjerite da je instaliran sljedeći paket:

`sudo apt install libsqliteodbc`

Priprema – ~/.odbc.ini

Na kraj datoteke /home/foi/.odbc.ini provjerite da postoji zapis:

[sqllitedb]

Description=Moj SQLite

Driver=SQLite3

Database=/tmp/sqlldb.db

Username = foi

Password = vjezbe

Upute

- Spojit ćemo FLORA-2 stroj na SQLite bazu podataka putem ODBC sučelja
- Sljedeće naredbe upisujemo na FLORA-2 konzoli
- Provjerite pristupne podatke da budu u skladu s podacima u /home/foi/.odbc.ini
- Pripazite u .odbc.ini je SQLite veza definirana prema privremenoj datoteci! Ako hoćete da rezultati ostanu i nakon ponovnog pokretanja računala izmijenite putanju u .odbc.ini do neke datoteke u Vašem home direktoriju!

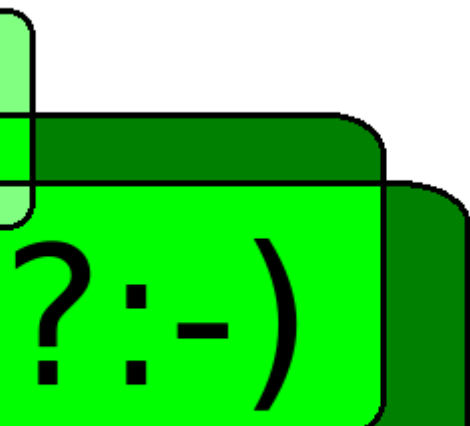
Primjer



```
[persistentmodules>>pm].  
newmodule{ mod1 }.  
mod1[attach(sqllitedb,?_,foi,vjezbe)]@pm.  
insert{ a:c[ x->y ]@mod1 }.  
\halt.
```

Ovom se naredbom konzola ugasila. Pokrećemo je ponovno u meniju Flora-2 > Start Flora2 process.

```
[persistentmodules>>pm].  
newmodule{ mod1 }.  
mod1[attach(sqllitedb,?_,foi,vjezbe)]@pm.  
?a[ ?x->?y ]@mod1.
```



Pojašnjenje

- Prvo smo učitali modul za perzistentnost
- Zatim smo kreirali novi modul (mod1)
- Tada smo modul povezali s bazom podataka u pozadini
- Onda smo ažurirali bazu znanja novom činjenicom
- I na kraju ugasili proces konzole.
- Kad smo ponovno pokrenuli konzolu i kreirali novi modul te ga spojili na istu bazu podataka, vidimo nakon upita da su podaci ostali pohranjeni.
- **NAPOMENA:** U takvom perzistentnom modulu pohranjuju se samo dinamički dodani podaci (pomoću insert, delete), podaci u datoteci se ne pohranjuju u bazu podataka!

Primjer

- Implementirat ćemo mali poslužitelj za rad s Flora-2 bazom znanja preko REST sučelja (korištenjem Python3 odnosno modulima pyxf, flask i json)

Priprema

- Ukoliko moduli nisu već instalirani, potrebno je instalirati sljedeće:

```
sudo pip3 install pexpect flask json
```

```
sudo pip3 install git+https://github.com/AILab-FOI/pyxf
```

Također, potrebno je dodati novi zapis (DSN) u .odbc.ini, npr.

```
[poslužitelj]
```

```
Description=Moj SQLite
```

```
Driver=SQLite3
```

```
Database=/home/foi/poslužitelj.db
```

```
Username = foi
```

```
Password = vjezbe
```

Upute

- Proučite datoteku **flora-server.py** dostupnu na:
<https://tinyurl.com/flora2-server>
- Podesite prava izvođenja skripte:
`chmod +x flora-server.py`
- Pokrenite pomoć skripte da vidite koje su njezine mogućnosti:
`./flora-server.py --help`
- Zatim provjerite je li trebata promijeniti neku opciju za pokretanje, te pokrenite poslužitelj npr.

`./flora-server.py --usr vjezbefoi`

Upute

- Ako je sve u redu, pokrenut je poslužitelj koji je dostupan na lokalnoj adresi (localhost)
- U proizvoljnom web pregledniku možete iskušati sljedeće upite:

`http://localhost:9876/query/insert{ a[b->c]@BazaZnanja }`

`http://localhost:9876/query/%3Fx[%3Fy->%3Fz]@BazaZnanja`

U drugom primjeru je ? zamjenjen 'escape' znakovima %3F jer ? Ima posebno značenje u URL adresi.

Web aplikacija - primjer

- Sada možemo implementirati jednostavnu web aplikaciju koja koristi bazu znanja koristeći Python 3, JavaScript i HTML
- Aplikacija će služiti kao pojednostavljeno sučelje za postavljanje upita nad bazom znanja
- Skinite sve potrebne datoteke u arhivi s adrese:
<https://tinyurl.com/flora2-app>
- Ekstrahirajte arhivu, podesite prava i proučite datoteke.

Upute

- Sučelje aplikacije dostupno je na adresi:

`http://localhost:9876`

- Iskušajte postaviti nekoliko upita, npr.:

`insert{ a[c->d]@BazaZnanja }`

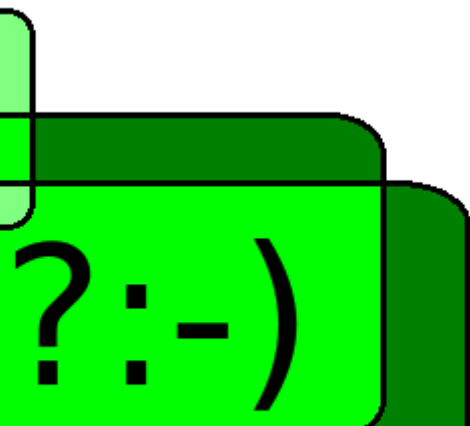
`?x[?y->?z]@BazaZnanja`

- **NAPOMENA:** Ovakav oblik implementacije u kojem se korisniku izravno omogućuje unos programskog koda koji se izvodi je veliki sigurnosni propust. U realnoj aplikaciji se dio koji se odnosi na kod upita svakako stavlja na poslužiteljsku stranu, dok klijent može samo koristiti aplikativno sučelje (API).

Zadatak



- Kreirajte perzistentnu bazu znanja ime_prezime.db (potrebno je ili dodati novi zapis u .odbc.ini ili promijeniti postojeći za SQLite) – perzistentni modul neka bude kolegiji
- Bazu znanja o kolegijima, nastavnicima i studentima iz prošlog zadatka učinite perzistentnom tako da definicije objekata dodate u bazu znanja
- Uz pomoć aplikacije za postavljanje upita iz prethodnog primjera spojite se na tu bazu znanja te napravite upite koji će na web stranici prikazati unesene kolegije, nastavnike i studente



Comic relief

<https://tinyurl.com/ml-parent>