

4. laboratorijska vježba: Java, XML i regularni izrazi

Cilj vježbe

Upoznavanje s programiranjem u programskom jeziku Java. Savladavanje osnova objektno orijentiranog dizajna. Rad s datotekama – učitavanje podataka i spremanje u XML. Rad s kolekcijama objekata. Rad s regularnim izrazima.

Priprema za vježbu

Proučiti – programski jezik Java:

- pisanje i prevođenje kôda (pročitati poglavlje *Getting Started* JavaTutorial-a)
- osnove jezika – tipovi podataka, operatori, izrazi, razredi, objekti, sučelja, nasljeđivanje, paketi (pročitati poglavlje *Learning the Java Language* JavaTutorial-a)
- osnovne ulazno/izlazne operacije s datotekama (pročitati poglavlje *Essential Classes/Basic I/O* JavaTutorial-a)
- skupove objekata (pročitati poglavlje *Collections* JavaTutorial-a)
- iznimke (pročitati poglavlje *Essential Classes/Exceptions* JavaTutorial-a)
- regularne izraze u programskom jeziku Java (pročitati poglavlje *Essential Classes/Regular Expressions* JavaTutorial-a)

Ponoviti – UML:

- dijagrame razreda – prikazivanje razreda, metoda, vrste veza između razreda te njihovo označavanje.

Zadatak za vježbu

Potrebno je izraditi samostalnu Java aplikaciju koja će učitavati podatke iz tekstualne datoteke i zatim te iste podatke zapisivati u XML datoteku.

Podaci u ulaznoj tekstualnoj datoteci odvojeni su znakom '|', npr:

```
Ime|Prezime|Broj telefona|Adresa|Pbr|Zemlja|Mail  
Ivo|Ivić|091 555-5555|Ilica 555|10000|Hrvatska|ivo.ivic@fer.hr
```

Pri parsiranju podataka iz ulazne datoteke koristiti regularne izraze!

Za osnovni element svake inačice postoje dvije različite varijante ulaznih podataka sa različito formatiranim podacima (npr. za inačicu 'Imenik', gdje je osnovni element Osoba, postoje ulazni podaci za Britance i Kanađane).

Potrebno je kreirati strukturu razreda – definirati razrede za:

- osnovni element (npr. za inačicu 'Imenik', osnovni je element Osoba)
- podvarijante osnovnog elementa (nasljeđuju osnovni element, npr. varijante elementa Osoba – Britanac i Kanađanin)
- svaki element brojnosti N (to su oni elementi iz inačice XML zapisa koji se mogu pojaviti više puta – u opisu inačica to je stupac "Brojnost")

Razred osnovnog elementa mora biti javan, apstraktan i definirati dvije metode:

- javnu metodu *toXML()* bez parametara, koja vraća objekt tipa String koji sadrži XML zapis tog objekta

- za oblikovanje izlaznog niza tipa `String` možete koristiti pomoćni razred `StringBuffer`
- zaštićenu apstraktnu metodu `parse(String redak)` bez povratne vrijednosti, koja parsira jedan zapis iz datoteke i popunjava svojstva objekta, te podiže iznimku `java.text.ParseException` ako dođe do greške pri parsiranju

Svaki razred koji nasljeđuje razred osnovnog elementa mora nadjačati implementaciju metode `parse` i u vlastitoj implementaciji metode izvesti parsiranje retka s podacima prema vlastitom obliku zapisa (npr. za inačicu 'Imenik' s osnovnim razredom `Osoba`, metoda `parse()` razreda `Kanadjanin` parsira podatke drugačije od metode `parse()` razreda `Britanac`). Za parsiranje podataka u metodama `parse()` koristite se regularnim izrazima.

Razredi koje predstavljaju element brojnosti `N` također moraju implementirati metodu `toXML()`.

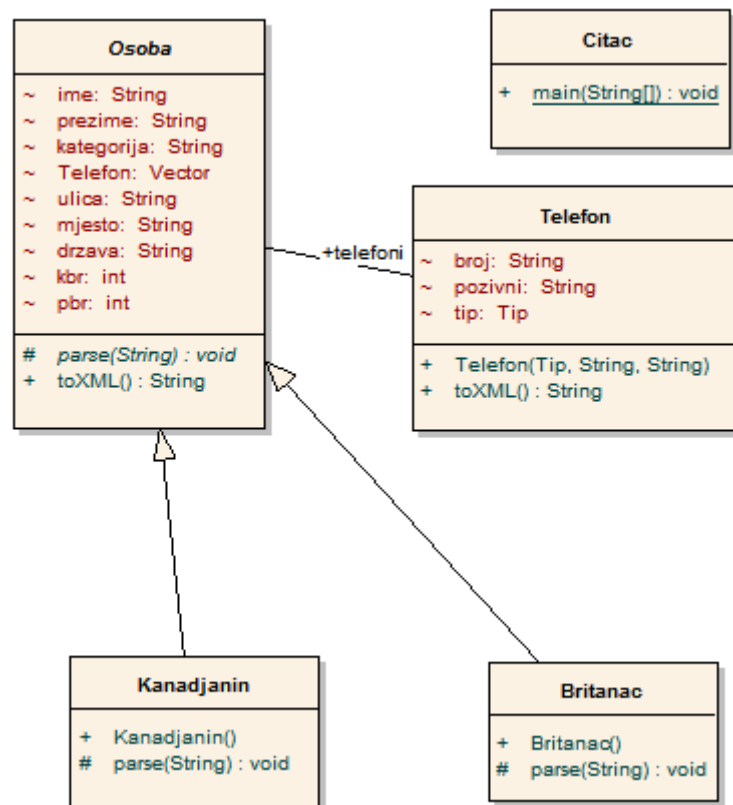
Također treba implementirati razred `Citac`, koja će sadržavati samo metodu `main()`. U toj metodi treba otvoriti datoteke čija imena su zadana kao parametar komandne linije (npr. `Britanac.txt`, `Kanadjanin.txt`). Na osnovu učitanih podataka iz datoteka, metoda `main()` treba stvoriti kolekciju objekata koristeći razred `java.util.ArrayList`. Kolekcija treba sadržavati sve objekte tipa osnovnog elementa. Za svaki zapis iz datoteke treba stvoriti odgovarajući objekt i dodati ga u kolekciju ili dopuniti postojeći objekt (koji se već nalazi u kolekciji) novim informacijama.

Na primjer, ako osoba ima više telefonskih brojeva, to je u datoteci zapisano u više redaka. U kolekciji podataka smije postojati samo jedan objekt razreda `Osoba` za tu konkretnu osobu, ali on treba sadržavati sve podatke o telefonskim brojevima te osobe.

Razredi objekata koje treba međusobno uspoređivati (u gornjem primjeru to je barem razred `Osoba`) trebaju nadjačati javnu metodu `equals(Object obj)` razreda `java.lang.Object` koja vraća istinosnu vrijednost (`boolean`).

Kad su svi podaci uneseni u kolekciju, treba generirati izlaznu XML datoteku slijednom obradom (pomoću metode `toXML()`) svih objekata pohranjenih u kolekciji objekata.

Imena ulaznih i izlaznih datoteka primaju se preko komandne linije. U metodi `main()` potrebno je zapisati i korijenski element nadređen osnovnom elementu svake inačice (npr. za inačicu 'Imenik' to je element `<imenik>` koji sadrži elemente `<osoba>`)



Sve ulazne podatke treba pretvoriti i pohraniti u formatu definiranom inačicom laboratorijskih vježbi. Nedostajuće obvezne podatke nadomjestiti po volji odabranim vrijednostima.

Predaja vježbe

Predaja vježbe podrazumijeva:

1. predočavanje datoteka s izvornim kôdom, eventualna objašnjenja
2. pokretanje aplikacije sa zadanim izvornim podacima
3. provjeru valjanosti generirane XML datoteke DTD-om iz 2. lab. vježbe
4. eventualne promjene u kôdu

Vježba se predaje na lokalnom računalu. Datoteke koje treba predati i prikazati su:

- Citac.java
- <imeOsnovnogElementa>.java
- <imePotomkaOsnovnogElementa>.java (1 ili više, ovisno o inačici)
- <imeElementaBrojnostiN>.java (1 ili više, ovisno o inačici)

U ovoj će vježbi biti potrebne i druge datoteke (npr. DTD iz 2. lab. vježbe). Na poslužitelj Moodle treba postaviti ZIP arhivu sa svim datotekama potrebnima za izvođenje i provjeru.

Ispitno gradivo vježbe

Ispitno gradivo uključuje sve navedeno u pripremi za vježbu, te detaljno razumijevanje napisanog rješenja i snalaženja u pravcima istog.

Primjeri pitanja:

1. Skicirajte dijagram razreda svoje inačice.
2. Validirajte izlaznu XML datoteku.
3. Koja metoda se poziva prilikom instanciranja objekta, ako razred nema definiran konstruktor?
4. Koje metode su dostupne za skup objekata koji ste koristili?
5. Kako iterirate kroz objekte u vašem skupu?
6. Napišite regularni izraz za zadani niz znakova.

Linkovi i literatura

JavaTutorial u HtmlHelp formatu - <http://www.allimant.org/javadoc> (The Java Tutorial, Nov 10, 2006 edition)

J2SE dokumentacija u HtmlHelp formatu - <http://www.allimant.org/javadoc> (J2SE 5 documentation ili J2SE 6 documentation)

J2SE API i dokumentacija (online) - <http://java.sun.com/javase/reference/api.jsp>