

## Praktikum iz objektno orijentisanog programiranja (13S112POOP)

### Projektni zadatak – Java

Napisati skup klasa sa odgovarajućim metodama i konstruktorima za realizaciju softverskog sistema za rad sa tabelama. Implementacija projektnog zadatka treba da se oslanja na rešenje realizovano na programskom jeziku C++ u sklopu prvog projektnog zadatka.

Korisnik (naručilac) softvera, želi da softver pruži sledeće funkcionalnosti:

- Interakciju sa korisnikom putem grafičkog korisničkog interfejsa
- Kreiranje tabela uz mogućnost učitavanja iz CSV formata
- Osnovna manipulacija
  - Prikazivanje tabele
  - Formatiranje polja tabele
  - Rad sa funkcijama
- Čuvanje tabele
- Kraj rada

Za uspešno rešenje zadatka potrebno je izvršiti analizu zahteva. Kao rezultat analize, potrebno je dopuniti i precizirati funkcionalnu specifikaciju softverskog alata. Na osnovu specifikacije, potrebno je napisati sistem klasa u jeziku Java koje realizuju traženi softver. U nastavku su navedeni neki elementi specifikacije. Od studenata se očekuje da dopune one stavke koje nisu dovoljno precizno formulisane, odnosno dodaju nove stavke (tamo gde to ima smisla) ukoliko uoče prostor za unapređenje. Izmene i dopune specifikacije mogu da donekle odudaraju od zahteva naručioca softvera u onoj meri u kojoj to neće narušiti traženu funkcionalnost. Takođe, priloženi UML dijagram koji opisuje zahtevani softver se ne mora obavezno poštovati, već samo predstavlja skicu potencijalnog rešenja. Prilikom izrade specifikacije voditi računa o potencijalnom unapređenju softvera na osnovu naknadnih zahteva.

Prilikom izrade rešenja, od studenata se očekuje intenzivno korišćenje svih onih mogućnosti koje pružaju specifikacija i biblioteke jezika Java, kao što su kolekcije, algoritmi, regularni izrazi, iteratori, lambda izrazi i sl. **Rešenja koja ne vode računa o ovom aspektu neće moći da dobiju maksimalan broj poena.** Takođe, voditi računa o **objektno orijentisanom dizajnu rešenja**, čistoći, čitkosti i komentaranju programskog koda.

## Funkcionalna specifikacija

U nastavku je zadat deo korisničkih zahteva koje treba razraditi i, po potrebi, dopuniti tako da se dobije funkcionalna aplikacija.

### Interakcija sa korisnikom

Korisnik može da interaguje sa programom izborom u datom trenutku dostupnih opcija putem **JavaFX grafičkog korisničkog interfejsa**. Interakcija može da se vrši putem tastature ili miša. U zavisnosti od izabrane opcije i njenih parametara, program izvršava zadatak opciju ili ispisuje poruku greške. Poruka greške treba da bude što je moguće detaljnija da bi korisniku pomogla da grešku otkloni. Sve eventualne parametre koji su potrebni prilikom rada aplikacije je potrebno zatražiti od korisnika. Ukoliko korisnik ne zada ništa, koristiti vrednosti fiksirane u programu.

### Pokretanje programa

Prilikom pokretanja programa korisniku se nudi da kreira novu tabelu ili da učitava postojeću. Potrebno je podržati sledeće načine za učitavanje tabele:

- CSV format – učitavanjem iz CSV formata dobijaju se samo vrednosti polja u tabeli. Informacije o formatiranju tabele nije moguće učitati iz ovog formata.
- Potrebno je definisati format koji bi podržao čuvanje podataka iz tabele uz informacije o formatiranju. Format mora biti zasnovan na JSON formatu čija je specifikacija data u prilogu ovog dokumenta. Za parsiranje i generisanje fajlova u ovom formatu **dozvoljeno je** korišćenje gotovih biblioteka. **Format treba da odgovara formatu koji je korišćen u prvom projektnom zadatku.** Iz toga sledi da je tabele napravljene u prvom projektnom zadatku moguće otvoriti pomoću alata koji je predmet drugog projektnog zadatka i obrnutno.

Po učitavanju postojeće ili kreiranju nove tabele korisnički interfejs za vreme izvršavanja programa prikazuje trenutno stanje tabele. Kolone tabele se obeležavaju velikim slovima engleskog alfabeta. Pretpostaviti da će jedno slovo uvek biti dovoljno za prikazivanje svih kolona tabele. Redovi tabele se numerišu celim brojevima počevši od 1. Određivanje načina za prikaz tabele se ostavlja studentima pri čemu je potrebno voditi računa o lakoći korišćenja alata, a posebno o preglednosti.

### Osnovna manipulacija

Korisniku treba dozvoliti sledeće opcije:

- Izmena podataka
- Formatiranje podataka
- Čuvanje podataka
- Napuštanje programa

## Izmena podataka

Potrebno je korisniku omogućiti da izmeni vrednost određene ćelije. Ukoliko uneta vrednost nije u odgovarajućem formatu korisniku se ispisuje poruka o grešci, a vrednost ćelije ostaje nepromenjena. Ukoliko vrednost ćelije počinje znakom = potrebno ju je tretirati kao formulu. Nakon znaka = u formuli se mogu pojaviti identifikatori ćelija (primer: C5), numeričke konstante (primer: 9.81), osnovni aritmetički operatori (+, -, \* i /), zagrade i beline. Prilikom prikazivanja ćelije u kojoj se nalazi formula potrebno je izračunati i prikazati vrednost formule. U slučaju greške ispisati ERROR crvenom bojom. Usvojiti da je vrednost formule uvek numeričkog tipa. **Za izračunavanje vrednosti formula potrebno je koristiti rešenje sa prvog projektnog zadatka. Koristeći JNI treba pozvati funkciju koja je deo C++ projekta, a koja obavlja potrebno izračunavanje. Funkciji je neophodno proslediti celokupnu tabelu (najbolje u CSV tekstualnom formatu), a rezultat njenog izvršavanja treba da bude sadržaj koji je potrebno prikazati korisniku nakon formatiranja (opet najbolje u CSV formatu).**

## Formatiranje podataka

Potrebno je podržati sledeće formate ćelija: datum, tekstualni i numerički. Datum se zadaje u formatu **dd.mm.gggg**. Prilikom postavljanja formata korisniku omogućiti da može da izabere jednu ćeliju (primer: C5) ili celu kolonu (primer: C). Numerički format se dodatno parametrizuje brojem decimala koje se prikazuju. Greška je ukoliko vrednost neke od ćelija kojima se postavlja format nije u skladu sa zahtevanim formatom (primer: u ćeliji se nalazi ceo broj, a korisnik želi da postavi datum kao format). Tekstualni format nema ograničenja.

## Čuvanje podataka i kraj rada

Korisnik može da zahteva da se tabela sačuva pri čemu je potrebno da zada format i putanju destinacionog fajla.

Korisnik može da zahteva kraj rada programa. Od korisnika se traži potvrda za napuštanje programa. Ukoliko korisnik nije sačuvaio tabelu koja je trenutno otvorena potrebno mu je ponuditi da to uradi pre napuštanja programa.

## Generisanje dokumentacije

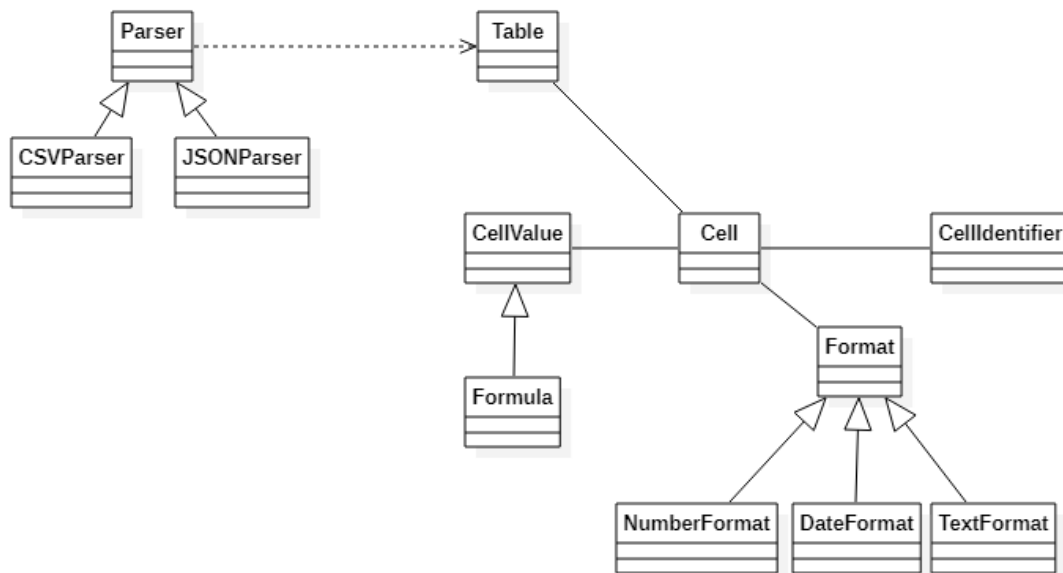
Potrebno je jednu klasu (koja ima veći broj metoda) detaljno opremiti dokumentacionim komentarima i generisati dokumentaciju za ovu klasu korišćenjem alata Javadoc.

## Testiranje rada programa

Prilikom testiranja rada programa moguće je koristiti postojeće alate za rad sa tabelama kao što je Microsoft Excel. Tabele napravljene u ovom alatu je moguće sačuvati u CSV formatu. Takođe, ovaj alat nudi mogućnost učitavanja tabela iz CSV formata te će biti u stanju da otvori svaku CSV datoteku generisanu ovim projektnim rešenjem. Skup funkcionalnosti koje su predmet ovog projekta predstavlja podskup funkcionalnosti koje nudi Microsoft Excel.

## Dijagram klasa

Na osnovu prethodne funkcionalne specifikacije formiran je sledeći dijagram klasa. Dijagram klasa nije detaljan, te ga treba tumačiti kao skicu koja načelno ukazuje na arhitekturu softvera. Studenti mogu da koriste ovaj dijagram kao referencu i, po potrebi, prošire ga da bi ga usaglasili sa eventualnim dopunama specifikacije.



Prilikom implementacije rešenja, obratiti pažnju na objektno orijentisani dizajn i intenzivno koristiti kolekcije i algoritme standardne biblioteke jezika Java i lambda funkcije gde god je to moguće.

## Specifikacija JSON formata

JSON (JavaScript Object Notation) je format za razmenu podataka. Lako se parsira i generiše, a čak je i ljudima pogodan za čitanje i pisanje. U pitanju je u potpunosti tekstualni format, nezavisan od bilo kog programskog jezika. JSON je izgrađen na osnovu dve strukture:

- Kolekcija parova ključeva i vrednosti - **objekat**, heš tabela, record, struct,...

Primer objekta je dat u nastavku:

```
{  
    "voce": "Jabuka",  
    "velicina": "Velika",  
    "boja": "Crvena"  
}
```

U vitičastim zagradama se, odvojeni zarezom, navode parovi ključeva i vrednosti. Ključ i vrednost se razdvajaju dvotačkom.

- Uređena lista vrednosti - **niz**, lista, vektor, sekvenca,...

Primer niza je dat u nastavku:

```
[ "Crvena", "Zelena", "Plava", "Bela"]
```

Ove dve strukture se mogu proizvoljno kombinovati čime se dobijaju složene strukture. Primer je dat u nastavku:

```
{ "kviz": {  
    "sport": {  
        "p1": {  
            "pitanje": "Kojim sportom se bavi Novak Djokovic?",  
            "odgovori": ["Tenis", "Odbojka", "Kosarka"],  
            "tacan": "Tenis"  
        }  
    },  
    "matematika": {  
        "p1": {  
            "pitanje": "5 + 7 = ?",  
            "odgovori": ["10", "11", "12", "13"],  
            "tacan": "12"  
        }  
    }  
  }  
}
```

Kako biste lakše razumeli JSON format otvorite sledeći link u bilo kom browser-u:

<https://query1.finance.yahoo.com/v8/finance/chart/aapl?period1=1616072670&period2=1617531870&interval=1d>

Iskopirajte celokupan sadržaj, a zatim posetite sajt <http://jsonviewer.stack.hu/>. Ovaj sajt nudi lak pregled JSON sadržaja. Nalepite (*Paste*) kopirani sadržaj u tekstualno polje i kliknite na dugme *Viewer* u gornjem levom uglu. Primer upotrebe je prikazan na slici 5.

The screenshot shows the JSON Viewer application. The 'Viewer' tab is selected. The JSON data is pasted into the text area. The tree view on the right shows the structure of the JSON data, including 'chart', 'result', 'meta', 'timestamp', 'indicators', 'quote', and 'low'. A table on the far right displays the 'low' values for each timestamp.

Name	Value
0	120.3199...
1	119.6800...
10	122.4899...
2	120.2600...
3	122.1399...
4	120.0699...
5	119
6	118.9199...
7	120.7300...
8	118.8600...
9	121.1500...

Slika 5 Prikaz JSON sadržaja

## **Bodovanje projekta**

Osnovne funkcionalnosti (učitavanje, čuvanje u CSV formatu, izmena i prikaz podataka): **15** poena

JSON format: **5** poena

Formatiranje tabele: **5** poena

Rad sa formulama - JNI: **10** poena

**Napomena:** Studenti koji nisu radili prvi projektni zadatak rad sa formulama treba da implementiraju na programskom jeziku Java.

Na broj osvojenih poena utiče i:

- korišćenje mogućnosti koje pruža biblioteka jezika Java (kolekcije, algoritmi, regularni izrazi, iteratori, lambda izrazi)
- objektno orijentisani dizajn rešenja,
- čistoća, čitkost i komentarisanje programskog koda