

# Struktura seminarskog rada

---

- **Nacrt naslova rada**

*GraphIt*: Komponentni razvoj aplikacije za vizualizaciju podataka u formi grafa

- **Autori**

Đorđić Jovan

Đurić Anastasija

Petrov Dina

- **Publika**

Našu publiku čine softverski inženjeri, programeri i studenti tehničkih nauka zainteresovani za principe, proces, i konkretnu primenu komponentnog razvoja aplikacija, kao i pojedinci zainteresovani za vizualizaciju i grafički prikaz podataka.

- **Zašto je publici stalo do teme rada?**

Publiku interesuje način na koji je moguće implementirati individualne, međusobno nezavisne komponente za učitavanje podataka iz različitih proizvoljnih izvora, kao i vizuelizaciju tih podataka na više nivoa složenosti i detaljnosti. Od najvećeg interesa za publiku je predstavljanje primera konkretnog složenog i skalabilnog softverskog proizvoda, nastalog kombinovanjem pomenutih komponenti u zajedničku funkcionalnu celinu.

- **Šta publika očekuje da pročita u radu?**

Publika očekuje da joj se približi pristup komponentnom razvoju prateći konkretan primer aplikacije za grafičku vizualizaciju podataka iz proizvoljnih izvora.

- **Šta publika zna, a šta treba pojasniti u radu?**

Publika je upoznata sa ključnim elementima programskih jezika *Python* i *JavaScript* kao i sa osnovnim principima teorije grafova. U radu treba pojasniti korišćenje konkretnih vizualizacionih biblioteka za grafički prikaz podataka, kao i pravilan postupak kreiranja i sinhronizacije nezavisnih komponenti.

- **Kakav stav publika ima prema temi?**

Publika je motivisana da nauči kako da primenjuje principe komponentnog razvoja prilikom implementacije sopstvenih aplikacija.

- **Koja pitanja bi čitaoci mogli postaviti?**

Kako bi se mogli iskoristiti dati primeri komponenti u drugim aplikacijama? Kakva je otpornost aplikacije u slučaju otkazivanja pojedinačnih komponenti? Kako se definiše interfejs koji pojedinačne komponente implementiraju da bi međusobno

komunicirale. Kako pojedinac može da prilagodi ili unapredi komponentu svojim potrebama?

- Šta želite od publike?

Želimo da im pružimo uvid u značaj ponovne iskoristivosti softverskih proizvoda koja se postiže komponentnim razvojem.

- Svrha i motivacija rada

- Ključna poruka

Komponentnim razvojem kreirana aplikacija je robusna, skalabilna i fleksibilna. Razvoj, prilagođavanje, izmena i unapređivanje pojedinačnih komponentni potpuno je nezavisno, dok njihovo usklađivanje ostavlja utisak jedinstvene homogene koherentne celine. Aplikacija se može koristiti u situacijama kada je korisniku potreban grafički prikaz relacija između entiteta unutar velike količine podataka. Komponente omogućavaju uniformnu reprezentaciju podataka koji se mogu učitati iz bilo kojeg izvora i formata. Grafički prikazi podataka olakšavaju razumevanje i sagledavanje strukture podataka, kao i izvođenje zaključaka na osnovu prikaza.

- Svrha

Svrha rada je da prikaže proces razvoja softvera kroz primer aplikacije za grafičku vizualizaciju podataka koji mogu poticati iz različitih izvora. Primer se sastoji od dva ključna tipa komponenti - za učitavanje i za grafički prikaz podataka. Rad treba da objasni postupak kreiranja individualnih komponenti i njihovu integraciju i međusobnu sinhronizaciju.

- Motivacija

Čest problem pri razvoju softvera je fokusiranje na pisanje novog koda umesto integracije postojećih rešenja u nove sisteme. Sve više postoji potreba za omogućavanjem pouzdane i robusne ponovne iskoristivosti. Primenom gotovih rešenja znatno se ubrzava razvoj sistema i izbegava se “izmišljanje tople vode”. Razvoj baziran na komponentama nam nudi standardizovan način za rešavanje pomenutih problema.

- Organizacija rada

- Apstrakt

**Sažetak rada:** Opis problema, način rešavanja i zaključak - podela sistema na komponente i njihova međusobna komunikacija uz prednosti koje ovaj pristup donosi.

- komponenta
    - *plugin*
    - CBD (*Component-Based Development*)

- Uvod

- **Motivacija** - kratak opis tematike i problema

- **Case study: GraphIt** (sažet opis našeg primera komponentne aplikacije)
- **Glavni zaključci**
- **Organizacija po poglavljima** - pregled sadržaja rada
- **Razvoj baziran na komponentama**
  - Mane *custom made* softvera
  - Modularizacija - *divide and conquer* pristup rešavanju kompleksnih problema
  - Prednosti i cilj: visoka kohezija i slaba sprega, ponovna iskoristivost (reusability), kupovina komponenti od nezavisnih proizvođača -> brzi razvoj kvalitetnog softvera
  - Šta su komponente?
  - Integracija kao preduslov ponovne iskoristljivosti komponenti
  - Preduslovi uspešne komponentizacije (standardi i marketplaces)
  - Platforma (CORE) - obezbeđuje infrastrukturu u koju se komponente ugrađuju - pruža skup bazičnih servisa koje komponente mogu da koriste
  - Servisi - skup funkcionalnosti koje komponenta pruža drugim komponentama i čije korišćenje je definisano ugovorom, koji propisuje način upotrebe (npr. interfejsi i protokoli), i ograničeno politikom upotrebe servisa.
  - Komponenta u ulozi pružaoca i/ili konzumenta servisa.
- **Implementacija aplikacije za vizualizaciju podataka**
  - **Komponentni razvoj u Python-u**  
Osnove *setuptools* alata.  
Primer podešavanja komponente.
  - **Organizacija i modeli entiteta aplikacije**  
Dijagram klasa.  
Dijagram komponenti.
  - **Core komponenta**  
Platforma (jezgro aplikacije) koja upravlja komponentama za učitavanje i prikaz.
  - **Komponente za učitavanje i parsiranje podataka**
    - XML loader
    - File system loader
    - Deezer & Spotify loader
  - **Komponente za prikaz učitanih podataka**
    - Simple visualisation
    - Complex visualisation
    - Tree view
    - Bird view

- **Demonstracija upotrebe aplikacije *GraphIt***

Primer upotrebe.

Slike ekrana.

- **Prednosti komponentnog razvoja**

Šta nam je omogućio komponentni razvoj u našoj aplikaciji.

Skalabilnost aplikacije, otpornost na greške, fleksibilnost i visoka kohezija..

- **Izazovi prilikom programiranja preko komponentnog razvoja**

Interfejsi, slaba sprega između komponenti.

- **Zaključak**

**Rezime rada:** podela složenog sistema na komponente, benefiti koje donosi razvoj baziran na komponentama.

**Pravci daljeg razvoja:** moguća unapređenja *GraphIt* aplikacije (opcije za čuvanje podataka, uvođenje novih komponenti ili familija komponenti, dodavanje i izmena podataka u realnom vremenu, analiza podataka - zaključci i predikcije na osnovu učitanih podataka).

**Ponovna iskoristivost komponenti korišćenih za izgradnju *GraphIt* aplikacije.**

- **Literatura**

[1] C. Szyperski, *Component Software: Beyond Object-Oriented Programming*, Addison-Wesley Longman Publishing Co., Inc., 2002

Osnovi principi i teorija vezana za komponentni razvoj

[2] Lau, Kung-Kiu, and Simone Di Cola. *An Introduction to Component-Based Software Development*. Vol. 3. # N/A, 2017.

Uvod u komponentni razvoj, dijagrami

[3] <https://d3js.org/>

Dokumentacija D3.js biblioteke za grafički prikaz podataka

[4] <https://docs.python.org/3/>

Python dokumentacija

[5] <https://setuptools.readthedocs.io/en/latest/setuptools.html#developer-s-guide>

Setuptools dokumentacija