

УНИВЕРЗИТЕТ У БЕОГРАДУ  
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ



Еволуција софтвера  
Test First Development - Dictionary using Trie

Професор:  
Драган Бојић

Студент:  
Душан Стијовић  
2021/3335

## 1. Преузимање и покретање пројекта

### 1.1. Преузимање пројекта

Пројекат се налази на *GitHub*-у. Могуће га ја преузети на следећем [линку](#). Такође, изворни код се налази у фолдеру уз овај документ.

### 1.2. Покретање пројекта

- Прво је потребно превести преузети код.
- Како бисмо превели код потребно је имати:
  - Java 18
  - IntelliJ
- Отворити пројекат у *IntelliJ*, направити нову конфигурацију типа *Application* и покренути пројекат кликом на `run` дугме.

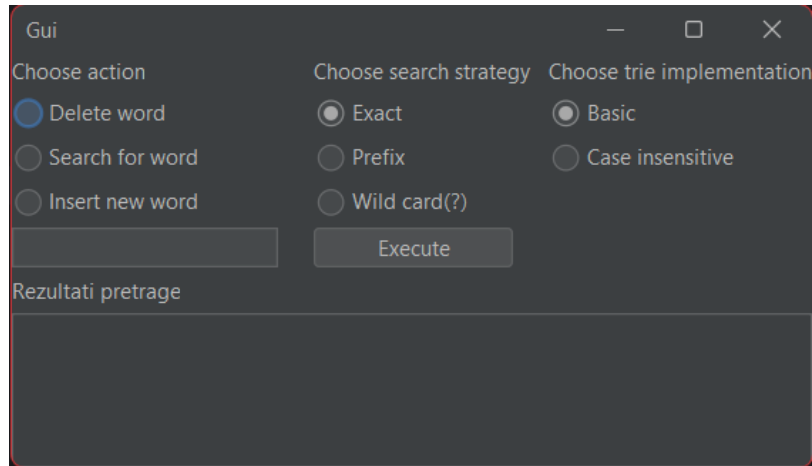
### 1.3. Покретање тестова

- Тестове је такође могуће покренути из *IntelliJ* тако што се направи нова конфигурација типа *JUnit* и кликне се на `run` дугме.

### 1.4. Напомена

Превођење и покретање је препоручљиво радити из *IntelliJ* јер је он коришћење за израду графичког корисничког интерфејса и он нам омогућава да се компоненте интерфејса повежу без потребе за додатним кодом. Наравно, могуће је у некој наредној верзији апликације променити ово тако да је све могуће одрадiti из терминала помоћу *maven-a*.

## 2. Опис апликације



- *Choose action* (Избор акције над речником)
  - *Delete word*
    - Брисање унете речи
  - *Search for word*
    - Претрага унете речи
  - *Insert new word*
    - Убацивање нове речи у речник
- *Choose search strategy*
  - Услов који ће се користити како би се проверило да ли се унета реч за претрагу поклапа са неком речи у речнику
  - *Exact*
    - Потпуно поклање
  - *Prefix*
    - За све речи које у речнику почињу као унета реч испуњено је поклапање
  - *Wild card (?)*
    - У унетој речи на местима где је могуће да се нађе било које слово потребно је ставити упитник. За све речи у речнику које имају исту дужину и иста остала слова (осим места са упитником, та места се прескачу) ће бити испуњено поклапање.
- *Choose trie implementation* (Избор имплементације)
  - *Basic*
    - Класична имплементација trie стабла
  - *Case Insensitive*
    - У овој имплементацији се слова унета као мала или као велика понашају на исти начин односно као да су унета мала слова.
- Након избора акције, стратегије за претрагу, као и имплементације стабла потребно је унети реч у input box и кликнути на дугме *Execute*
- Након тога уколико је акција била претрага у *Rezultati Pretrage* ће се појавити листа са речима које постоје у речнику а поклапају се са унетим параметрима
- Са десне стране дугмета *Execute* се исписује порука о статусу последње извршене акције

### 3. Опис реализације апликације

- Апликације је имплементирана користећи Test First методологију. Ово значи да су прво писани тестови а затим код који омогућава да ти тестови прођу. Ово је рађено у више итерација. Приликом писања тест кода писане су мале целине, а затим су те целине имплементирани. Приликом имплементације писано је само онолико кода колико је довољно да тестови прођу. Након сваког писања имплементације пуштани су сви тестови како би се осигурали да пролазе. Након више оваквих итерација одрађено је рефакторисање кода уколико је потребно. Рефакторисање је рађено како за код апликације тако и за код који тестира код апликације. Избачени су дупликати, издвојене су неке целине у функције, промењена су имена неких променљивих, неки алгоритми су промењени. Након неке мале промене пуштани су сви тестови како би се проверило да није поремећена нека функционалност апликације.

#### 3.1. Ток развоја апликације

- Прво је одрађена основна имплементација *Trie* стабла. Ова имплементација је користила листу речи. Прво су одрађене операције за потпуно преклапање, а затим за префикс.
- Након основне имплементације, одрађена је промена имплементације користећи стабло уместо листе. Прво је имплементирана класа за чворове стабла а затим је та класа уграђена уместо листе.
- Након овога направљена је имплементација *Case Insensitive* стабла. Овом приликом дошло је и до рефакторисања тестова и прављење основне класе за тестирање заједничких функционалности имплементације речника као и специфичне тест класе које су изведене из основне за *Basic* и *Case Insensitive* имплементацију.
- Затим је одрађено рефакторисање претраге и брисања. Уведене су стратегије поклапања уместо посебних функција за сваки вид преклапања. Имамо *Exact*, *Prefix* и *Wildcard* преклапање.
- Након овога започет је графички кориснички интерфејс. За сам дизајн интерфејса коришћен је уграђени *ScreenBuilder IntelliJ*-а. Помоћу њега је подешено где се налази која компонента, како се зове, да ли се може уносити текст, која опција је подразумевано селектована. Тестови за овај део графички корисничког интерфејса су проверавали да ли су компоненте које је потребно да постоје *null*, да ли су селектоване одговарајуће подразумеване опције у *radio button* групама, да ли текст може да се уноси, односно не уноси у одговарајуће компоненте.
- Након овога је започето писање тестова и интеграција модела у графички кориснички интерфејс. За ове потребе имплементиран је архитектурни шаблон *Model–View–ViewModel*. *ViewModel* је класа *TrieModelView* која представља све функционалности које би требало да буду подржане на графичком корисничком интерфејсу. Класа за графички кориснички интерфејс има примерак ове класе. Акције корисника се прослеђују овој класи, а сама логика за реаговање на дату акцију се налази у самој класи. Овим је олакшано тестирање графичког корисничког интерфејса. Односно, потребно је само тестирати да ли се одговарајућим акцијама корисника зову одговарајуће методе *TrieModelView* класе. Наравно, сама класа *TrieModelView* је имплементирана користећи TDD.

#### 4. Списак коришћених библиотека и алата

- Алати
  - *IntelliJ*
- Библиотеке
  - *JUnit*
    - За потребе тестирања
  - *FlatLaf*
    - Коришћен за тему графичког корисничког интерфејса