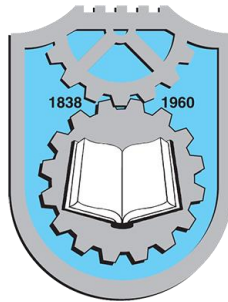


Универзитет у Крагујевцу
Факултет инжењерских наука



Софтверски инжењеринг

Апликација за приказ сегментисане области
и рачунање њене површине на медицинским сликама

Студент:
Александра Милошевић 571/2016

Предметни професор:
Проф. Др. Ненад Филиповић

Крагујевац 2019.

Садржај

1. Поставка задатка и детаљан опис апликације	3
1.1 Дефинисање пројекта	3
1.2 Опис коришћења апликације	3
2. Опис делова програма	4
2.1 Увод	4
2.2 Класа <code>RacunanjePovrsine</code>	4
2.3 Класа <code>Segmentiranje</code>	5
2.4 Коришћење апликације	6
3. UML дијаграми	7
3.1 Дијаграм класа	8
3.2 Дијаграм активности	9
3.3 Дијаграм случајева коришћења	10
3.4 Дијаграм стања	11
3.5 Дијаграм секвенци	12
4. Литература	13

1. Поставка задатка и детаљан опис апликације

1.1 Дефинисање пројекта

У овом задатку потребно је направити апликацију која израчунава површину костију са 2D снимака DICOM формата медицинских слика, као и означити сегментисану област од интереса.

1.2 Опис коришћења апликације

Корисник учитава медицинску слику .dcm формата која је смештена у жељеном фолдеру на жељеној локацији. На основу боје пиксела на слици, апликација израчунава површину костију и враћа је кориснику. На слици се означава сегментисана област од интереса тј. површина костију, и чува се под истим називом, али под другом екстензијом, након извршетка програма. Та слика се из .dcm формата конвертује у слику .jpg формата (због коришћења RGB боја) и чува на жељеној локацији. DICOM слике немају опцију RGB боја. Корисник може бесконачан број пута поново покретати програм, али приликом једног покретања можемо користити само једну слику.

2. Опис делова програма

2.1 Увод

За израду овог пројектног задатка коришћен је Python програмски језик (3.7 верзија). Неопходно је коришћење додатних библиотека и пакета: pydicom, numpy, cv2 (слика 1). Pydicom је чисти Python пакет за рад са DICOM датотекама као што су медицинске слике, извештаји и радиотерапијски објекти. Pydicom олакшава читање ових сложених датотека у природне python-ske структуре ради једноставне манипулације. Измењени скупови података могу се поново уписати у датотеке DICOM формата. NumPy је библиотека за програмски језик Python, која додаје подршку великим, вишедимензионалним низовима и матрицама, заједно са великом колекцијом математичких функција високог нивоа за рад на овим низовима. Pandas је библиотека отвореног кода, BCD лиценцирана библиотека која пружа високе-перформансе, једноставне за употребу структуре података, и алате за анализу података за програмски језик Python. OpenCV је водећа библиотека отвореног кода за рачунарски вид, обраду слике и машинско учење, а сада нуди убрзање GPU-а за рад у реалном времену.

```
1  import pydicom
2  import numpy as np
3  import cv2
```

Слика 1 – Коришћене библиотеке

2.2 Класа RacunanjePovrsine

```
5  class RacunanjePovrsine(object):
6      def __init__(self, ime):
7          self.slika = ime
8      def racunanje(self):
9          filename = pydicom.dcmread(self.slika)
10         n=0
11         m=0
12         k=200
13
14         np.set_printoptions(threshold=np.inf)
15
16         for row in filename.pixel_array:
17             for cell in row:
18                 n=n+1
19         print(n)
20         for row in filename.pixel_array:
21             for cell in row:
22                 if cell>=k:
23                     m=m+1
24         print(m)
25         print(m/n*100)
```

Слика 2 – Класа `RacunanjePovrsine`

У овој класи врши се учитавање медицинске слике формата `.dcm` и рачуна се површина сегментисане области од интереса тј. површина костију, која се изражава у процентима приликом покретања програма и исписује на екрану.

2.3 Класа `Segmentiranje`

У овој класи имамо функцију бојење која садржи функцију `DicomtoRGB` за претварање боја у RGB формат боја, да бисмо могли лакше да препознамо боје. DICOM слике немају боје у RGB формату, због чега и вршимо конверзију.

```
27 class Segmentiranje(object):
28     def __init__(self, ime):
29         self.slika = ime
30
31     def bojenje(self):
32
33         def DicomtoRGB(filename, bt, wt):
34
35             img=np.array(filename.pixel_array, np.int16)
36             clipped_img = np.clip(img, bt, wt)
37
38             rgb_add = clipped_img - bt
39
40             rgb_mult = np.multiply(rgb_add, 255/(wt-bt))
41
42             rgb_int = np.around(rgb_mult, 0).astype(np.uint8)
43
44             rgb_img = np.stack([rgb_int]*3, axis=-1)
45             return rgb_img
46         filename = pydicom.dcmread(slika)
47
48         image=DicomtoRGB(filename, bt=0, wt=1400)
```

Слика 3 – Први део класе `Segmentiranje`

```
50 len=image.shape
51 for a in range(0, len[0]):
52     first=0
53     last=0
54     for b in range(0, int(len[1]/2)):
55         if (first==0):
56             if (image[a,b]<=[250,250,250]).all() and (image[a,b+1]>[250,250,250]).all() and (image[a,b+2]>
57                 [250,250,250]).all() and (image[a,b+3]>[250,250,250]).all():
58                 first=b+1
59         if (first!=0):
60             if (image[a,b]>[250,250,250]).all() and (image[a,b+1]<=[250,250,250]).all():
61                 last=b
62     for i in range(first, last):
63         image[a,i]=[0,33,166]
```

Слика 4 – Други део класе `Segmentiranje`

```

64     first=0
65     last=0
66     for b in range(int(len[1]/2),len[1]-3):
67         if (first==0):
68             if (image[a,b]<=[250,250,250]).all() and (image[a,b+1]>[250,250,250]).all() and (image[a,b+2]>
               [250,250,250]).all() and (image[a,b+3]>[250,250,250]).all():
69                 first=b+1
70         if (first!=0):
71             if (image[a,b]>[250,250,250]).all() and (image[a,b+1]<=[250,250,250]).all():
72                 last=b
73     for i in range(first,last):
74         image[a,i]=[0,33,166]
75
76     cv2.imwrite(slika+".jpg", image)
77
78     print("Nova .JPG slika: "+slika+".jpg")

```

Слика 5 – Трећи део класе Segmentiranje

На крају класе преводимо слику екстензије .dcm у .jpg и чувамо је под истим називом и на жељеној локацији коју уносимо приликом покретања програма.

```

76     cv2.imwrite(slika+".jpg", image)
77
78     print("Nova .JPG slika: "+slika+".jpg")
79
80     if __name__ == "__main__":
81         print("unesite ime slike ili putanju do slike!")
82         slika=input()
83         povrsina=RacunanjePovrsine(slika)
84         povrsina.racunanje()
85         novaSlika=Segmentiranje(slika)
86         novaSlika.bojenje()

```

Слика 6 – Део кода за покретање апликације

Овај део кода се прво извршава. Стварају се објекти класа и позивају се функције које користимо.

2.4 Коришћење апликације

У овој апликацији потребно је израчунати површину сегментисане области од интереса тј. површину костију и означити је. На почетку уносимо назив слике коју користимо или путању до ње ако се не налази у истом фолдеру у коме је и програм.

Да бисмо израчунали површину костију, користимо функцију за израчунавање површине. Уз помоћ две `for` петље вршимо рачунање укупног броја пиксела слике, и уз помоћ друге две `for` петље број пиксела беле боје која означава кости. Подешена је променљива $k=200$ да бисмо направили границу боја, тј. за белу боју користимо само вредности пиксела изнад 200.

Користећи `for` петље пролазимо кроз матрицу пиксела, тј. кроз редове и колоне у којима су пиксели са слике. На екрану се штампа број пиксела костију и површина костију на слици у процентима, која се добија дељењем белих пиксела са укупним бројем пиксела помноженим са 100. Сада је потребно означити сегментисану област од интереса. Уз помоћ конвертовања боја слике у RGB систем можемо лако мењати боје пиксела. Слика се дели на две половине тако да кости у односу на ту равну буду симетричне једна другој. Прво се са леве стране врши бојење, па онда са десне тако што се налази прва тачка беле боје, па онда последња тачка беле боје. Те две тачке се спајају и боје црвеном бојом. Провера се врши проласком кроз редове слике од почетка са леве стране до половине слике и редом се означава црвеном бојом област од интереса. Тако се исто дешава и са провером од половине до краја слике са десне стране. На крају, добијамо означену сегментисану област од интереса на слици црвеном бојом која се чува у формату `.jpg` под истим називом. На екрану се исписује порука да се сачувала нова слика.

3. UML дијаграми

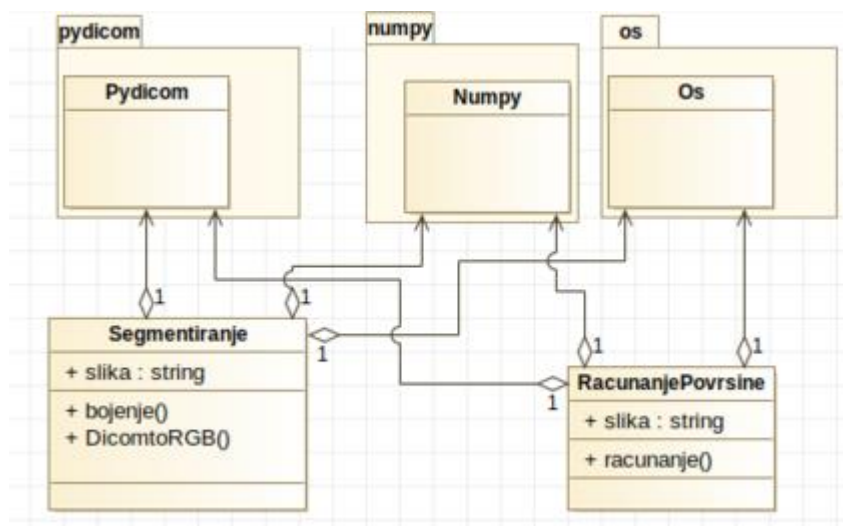
The Unified Modeling Language или скраћено УМЛ је скуп графичких нотација који нам помаже у опису и дизајнирању софтверског система. Како је УМЛ језик доста обиман ми ћемо се фокусирати на најчешће коришћене графичке елементе и дијаграме у пракси.

Описаћемо следеће дијаграме:

- дијаграм класа
- дијаграм активности
- дијаграм случаја коришћења
- дијаграм стања
- дијаграм секвенци

3.1 Дијаграм класа

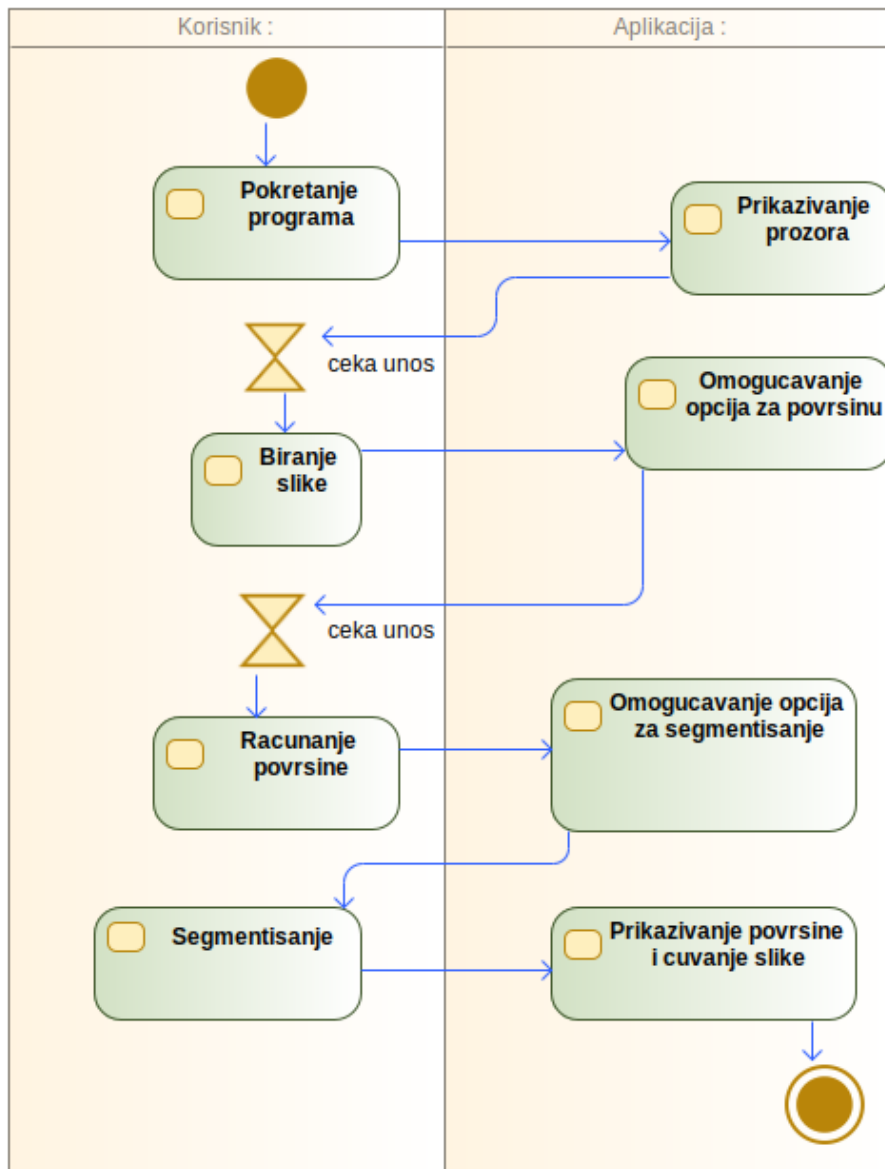
Дијаграм класа (део УМЛ-а) је врста структурног дијаграма у софтверском инжињерингу, који описује структуру састава објашњавајући класе унутар састава, њихове атрибуте и односе.



Слика 7 – Дијаграм класа задате апликације

3.2 Дијаграм активности

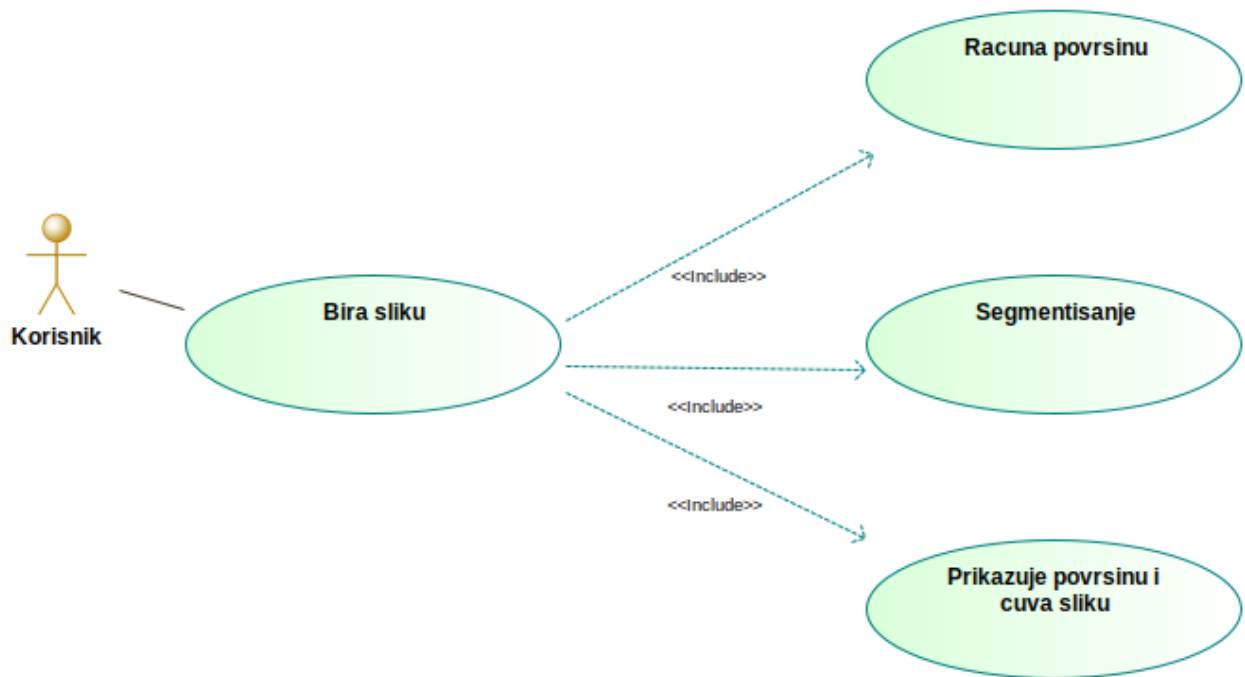
Активност је извршење неког задатка, било да се ради о физичкој активности или извршавању кода. Једноставно речено, дијаграм активности приказује редослед активности. Дијаграми активности омогућавају приказ условног понашања и паралелног извршавања. Почнимо од описа основних ознака:



Слика 9 – Дијаграм активности задате апликације

3.3 Дијаграм случајева коришћења

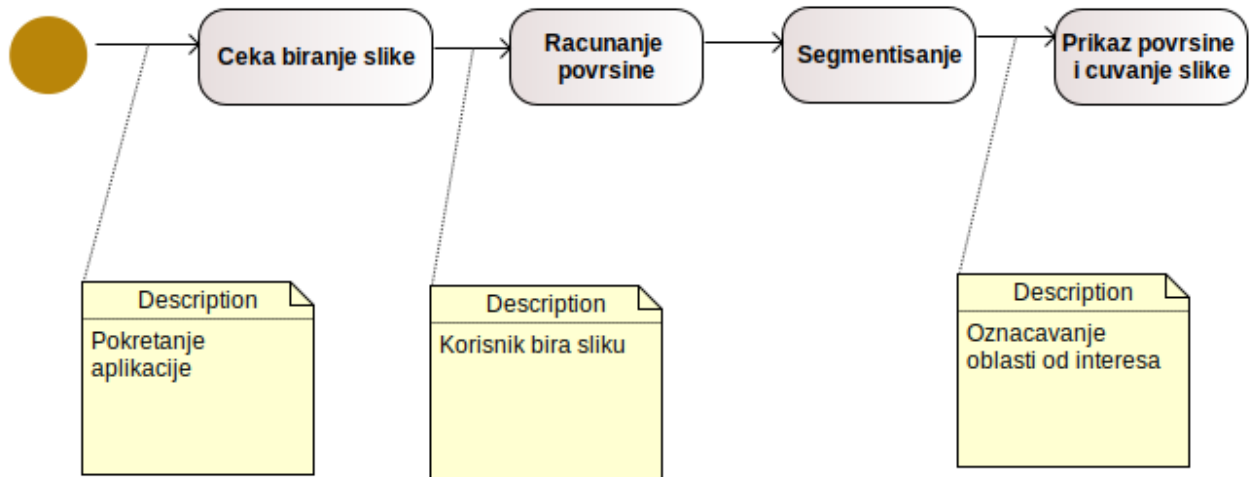
Дијаграм случајева коришћења (енгл. Use case diagram) приказ је интеракције корисника са системом који показује однос између корисника и различитих случајева коришћења у којима је корисник укључен. Дијаграм случаја коришћења може идентификовати различите типове корисника система и различите случајеве коришћења и често ће бити праћен и другим типовима дијаграма. Случајеви коришћења представљени су круговима или елипсама.



Слика 10 – Дијаграм случајева коришћења задате апликације

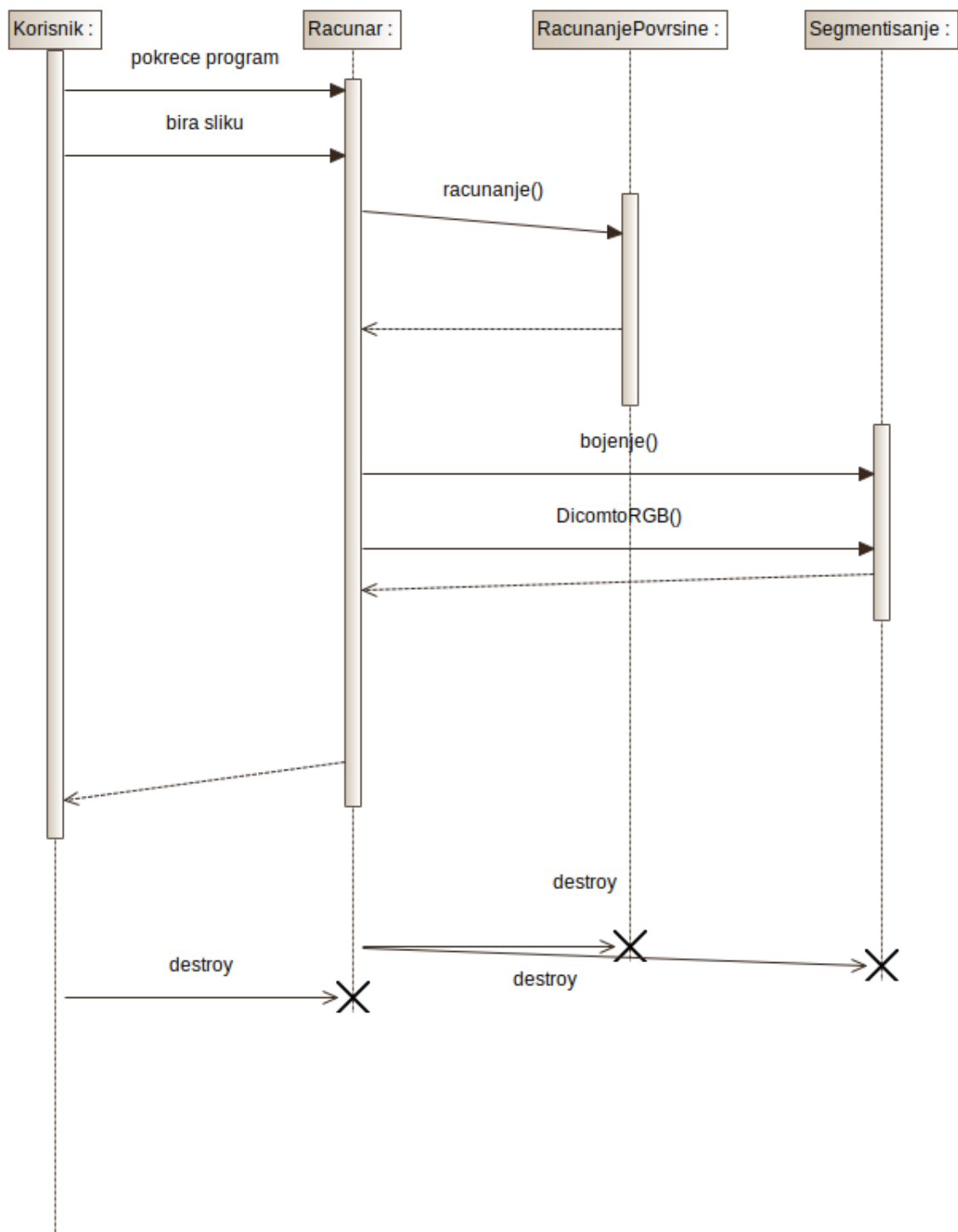
3.4 Дијаграм стања

Дијаграм стања је дијаграм који се користи на пољу компјутерских наука, који представља понашање система који је састављен од коначног броја стања. Постоје многи облици дијаграма стања, који се незнатно разликују и имају различиту семантику.



Слика 11 – Дијаграм стања

3.5 Дијаграм секвенци



Слика 12 – Дијаграм секвенци

4. Литература

https://hr.wikipedia.org/wiki/Dijagram_klasa