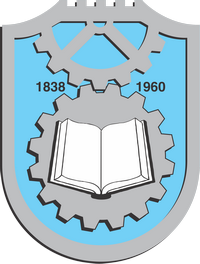
Универзитет у Крагујевцу

Факултет инжењерских наука



Пројектни рад

**Предмет:**

Софтверско инжењерство

**Тема:**

Графичка симулација Соларног система

**Студент: Професор:**

Давид Јеремић 624/2017 Ненад Филиповић

Тијана Шуштершич

Садржај

[Поставка и опис задатка - 3 -](#_Toc524039614)

[Опис изворног кода - 4 -](#_Toc524039615)

[Класа *Небеско Тело* - 4 -](#_Toc524039616)

[Класа *Соларни систем* - 7 -](#_Toc524039617)

[*UML* дијаграми - 11 -](#_Toc524039618)

[*Use Case Diagram* - 11 -](#_Toc524039619)

[Дијаграм секвенци - 12 -](#_Toc524039620)

[Дијаграм активности - 14 -](#_Toc524039621)

[Дијаграм стања - 15 -](#_Toc524039622)

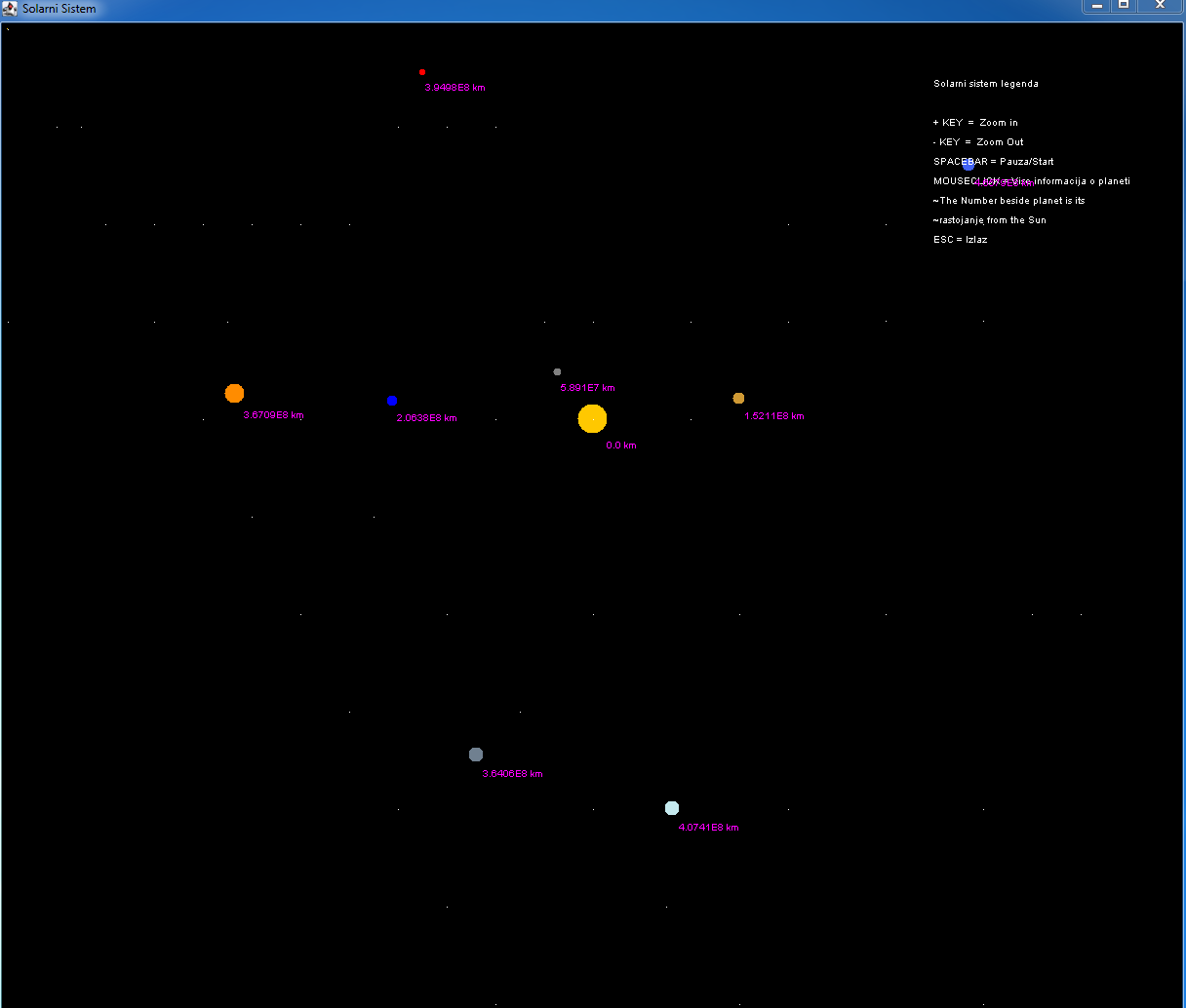
[Дијаграм класа - 16 -](#_Toc524039623)

[Литература - 17 -](#_Toc524039624)

# Поставка и опис задатка

Тема овог пројектног задатка јесте графичка симулација Соларног система. Све планете треба да су у реалном односу брзина и величина.

Апликација је рађена тј. написана у програмском језику **Java**. Писана је у окружењу *Visual Studio Code*, а најбоље је покренути апликацију у окружењима као што су *Eclipse* и *Intellij Idea.* Да би покренули апликацију, потребно је имати **Јava-ину** библиотеку ***Swing*** која служи за креирање GUI.



*Слика 1. Покренута апликација*

Корисник може мануелно да користи опције које су омогућене у апликацији преко пречица које се налазе на тастатури. Кликом на тастер *SPACEBAR*, корисник добија опцију *Пауза*  за паузирање наше апликације и опцију *Старт* за стартовање паузиране апликације. Потом, кликом на тастере *PLUS KEY* и *MINUS KEY,* можемо зумирати и одзумирати наше планете на апликацији. Ако кликнемо директно на неку од планета или на сунце левим тастером миша можемо добити више информација о планети на коју смо кликнули тастером. За крај, кликом на тастер *ESC* излазимо из наше апликације.

Звезде су беле тачке које се насумично крећу по екрану.

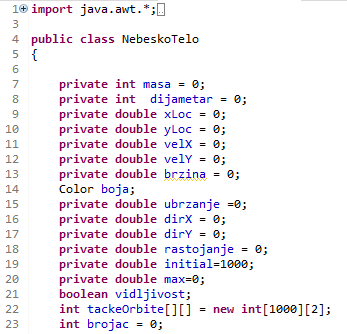
# Опис изворног кода

Централна класе наше апликације су:

* Небеско тело
* Соларни Систем

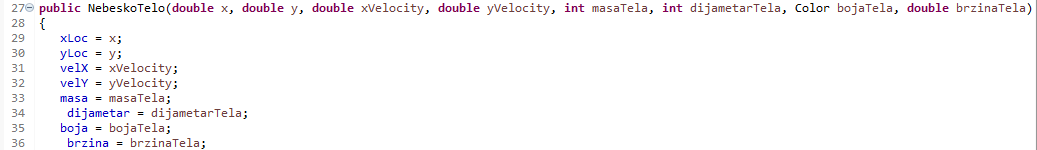
## Класа *Небеско тело*

Коришћењем класе Небеско Тело, апликација исцртава и конструише планете и звезде.Све планете имају своју масу, брзину, растојанје, дијаметар. На почетку класе смо дефинисали објекте, то можемо приметити на следећим линијама кода(слика 2):



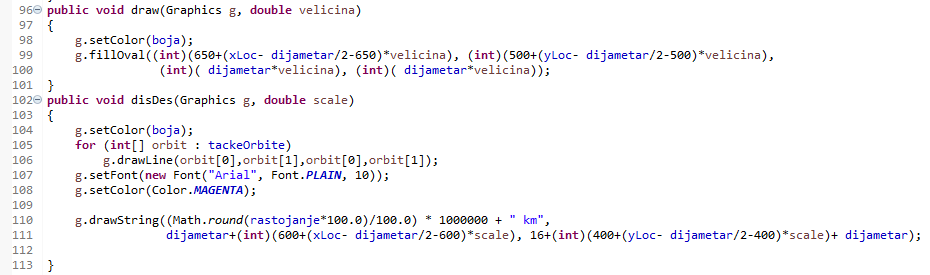
*Слика 2. Класа Небеско Тело*

Накок тога имамо конструктор за објекте планета(слика 3).



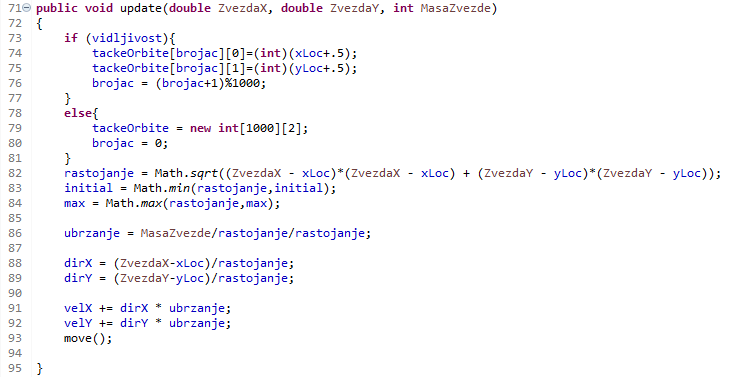
*Слика 3. Конструктор класе Небеско Тело*

Јако је битно напоменути функције које служе за цртање и конструисање планета и звезда(слика 4).



*Слика 4. Функција за исртавање планета*

Такође важна функција *update* за конструисање звезда(слика 5).



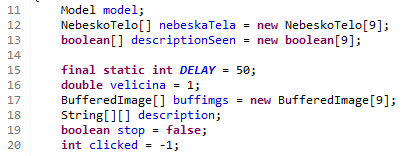
*Слика 5. Функција за конструисање звезда*

Аргументи ове функције су координате Звезда и њихова маса. У функцији рачунамо растојање звезда, њихово убрзање и позиције.

## Класа *Соларни систем*

У класи Соларни Систем убацују се сви параметри небеског тела, као и слике(слика 6). У бесконачној петљи се стално ажурирају координате сваке од планета.

У овој класи смо користили два низа, један који служи за учитавање слика планета, а други који служи за планете.

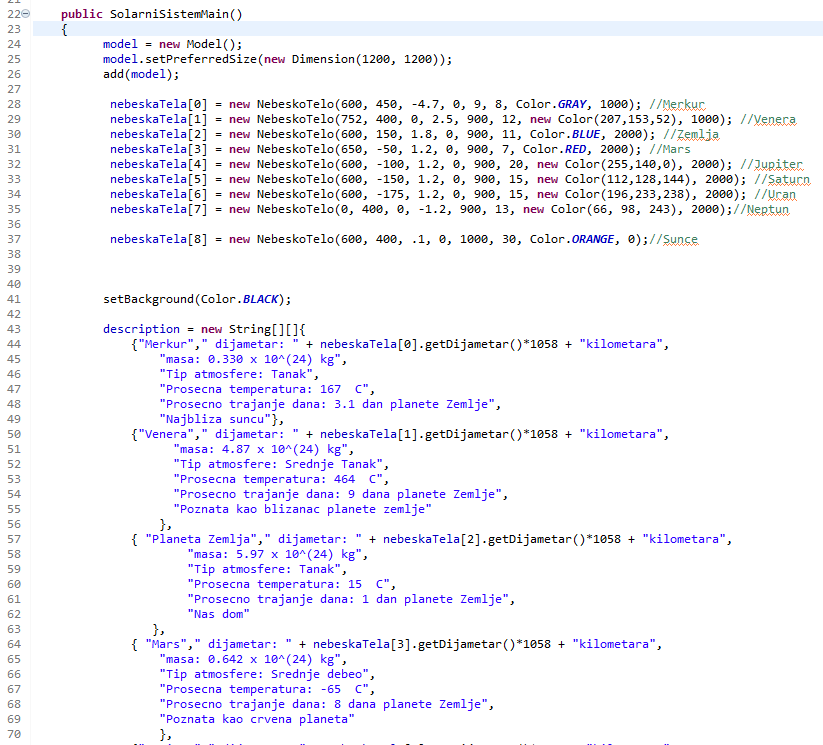


*Слика 6. Атрибути и низови*

Као што смо рекли један низ садржи типове података *buffimgs,* а други инстанце класе Небеско Тело.

Класа Соларни Систем садржи два конструкотора:

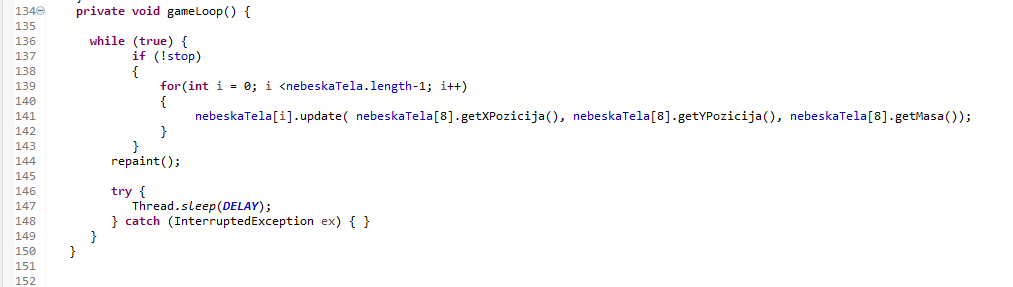
1. Један који се користи за подразумеване величине и параметре
2. Други који се користи за мануелно коришћење



*Слика 7. Конструктор*

*Слика 8. Учитавање слика и текста*

Модел је постављен на величину 1200 пиксела пута 1200 пиксела. Између линија 28 и 37, иницијализују се планете и сунце. Линија 41 поставља црну позадину. Након тога између линија 43 и 98, налази се опис за сваку планету и сунце који се касније појављује на екрану апликације. Касније, између линија 101 и 109 се учитавају слике планета.

*Слика 9. Петља*

Функција *gameLoop* је бесконачна петља, где се стално ажурирају координате планета. А линија 144 је намењена за поновно цртање планета.

 *Слика 10. Исцртавање збезда*

Класа модел служи за цртање планета и сунца на линији 163. У *for* петљи се исртавају звезде на насумичним местима.

# *UML* дијаграми

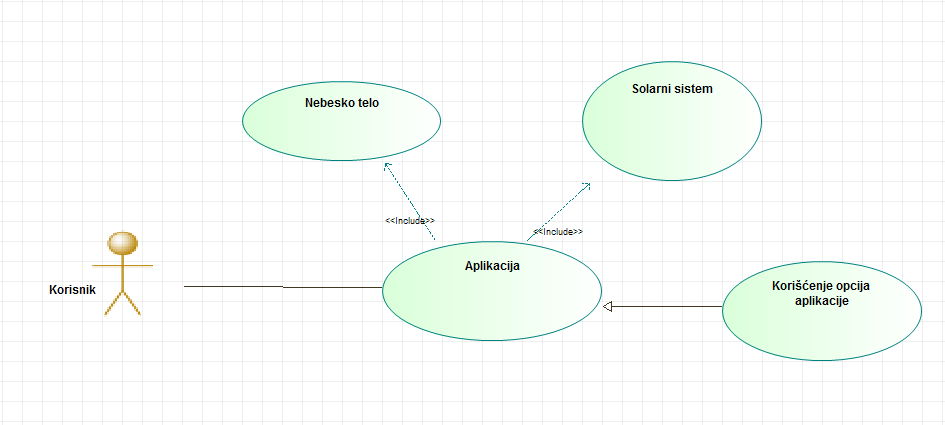
## *Use Case Diagram*

* Дијаграм случајева коришћења (use -case) приказује скуп случајева коришћења и актера
* Типично се користи да специфира неку фукционалност и понашање неког субјекта
* Дијаграм визуелизује понашање система, подсистема или чак класе и интерфејса

Елементи дијаграма су:

* случајеви коришћења
* актери
* релације
* пакети

Дијаграм случајева коришћења на нашем пројекту изгледа овако(слика 11):

*Слика 11. Use case diagram*

## Дијаграм секвенци

Дијаграм секвенци приказује комуникацију између скупа објеката, која се остварује порукама које објекти међусобно размењују у циљу остваривања очекиваног понашања. Детаљно описује како се операције изводе – које поруке се шаљу и када.

Дијаграм секвенци је један од дијаграма интеракције

• Интеракција – понашање које обухвата скуп порука које се размењују

између скупа објеката у неком контексту са неком наменом

• Порука – спецификација комуникације између објеката која преноси

информацију

• Пријем поруке изазива акцију – извршење наредбе

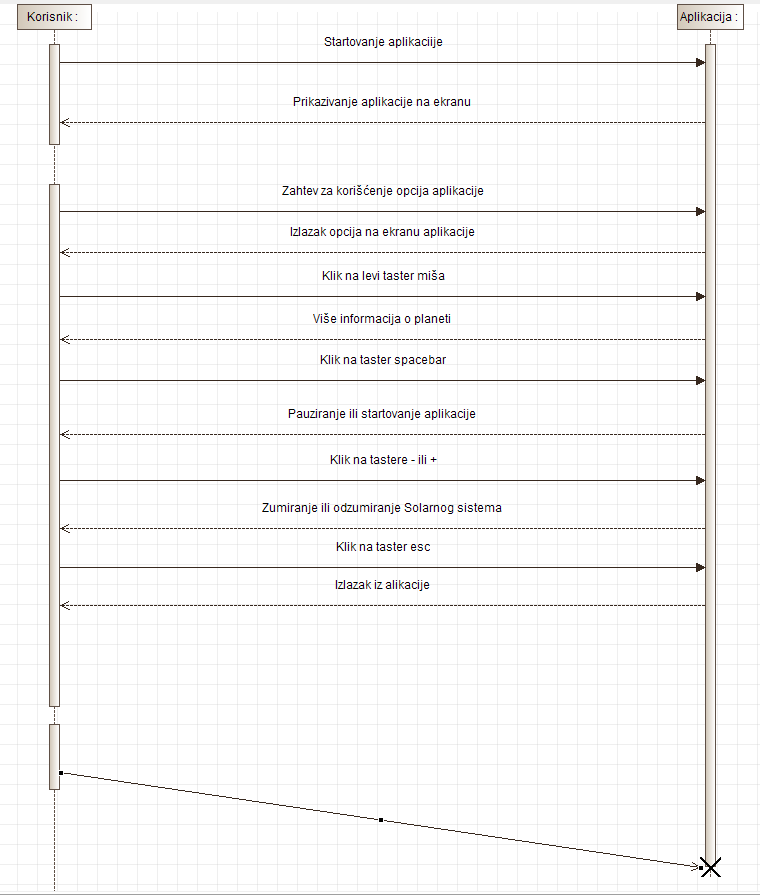
Ако су дијаграми случајева употребе претходно дефинисани – дијаграм секвенци је

једна од његових реализација – показује редослед:

• Догађаја – спољашњи улазни догађај – генерише учесник

• Операција – оџив на догађај у систему

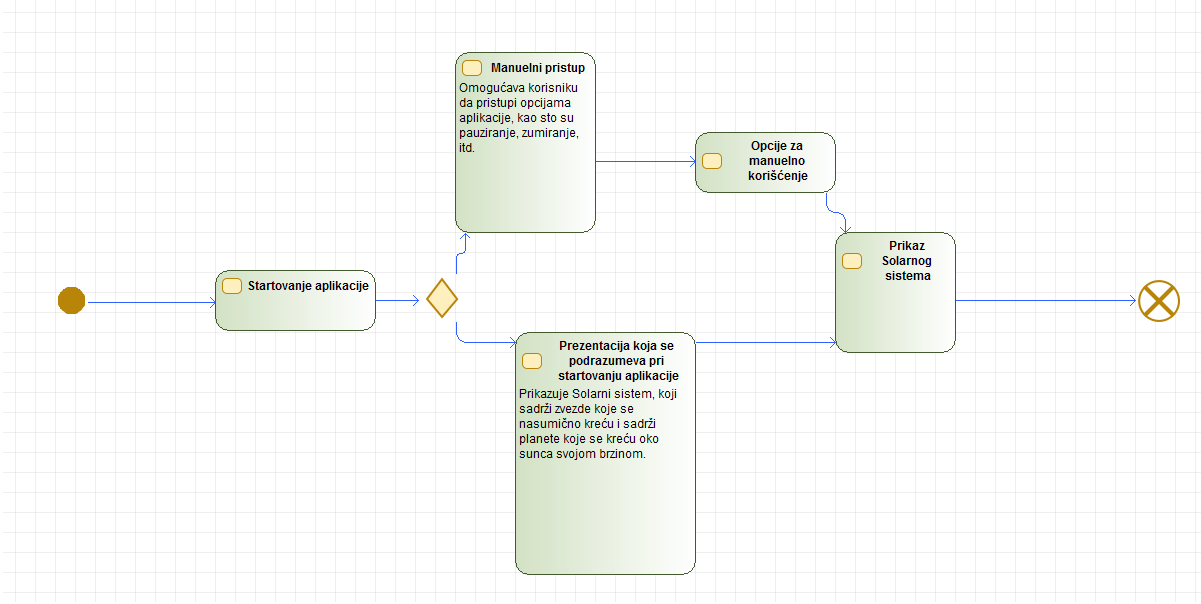
На следећој слици можемо видети овај дијаграм(слика 12):

 *Слика 12. Дијаграм секвенци*

## Дијаграм активности

Дијаграм активности је намењен моделирању динамичких аспеката система. Дијаграм активности приказује ток активности коју извршавају објекти и евентуално и ток објеката између корака активности.

Активност је спецификација параметризованог понашања које се изражава кроз ток извршења преко секвенцирања и конкурисања подактивности. Дијаграм активности садржи чворове и гране.

*Слика 13. Дијаграм активности*

## Дијаграм стања

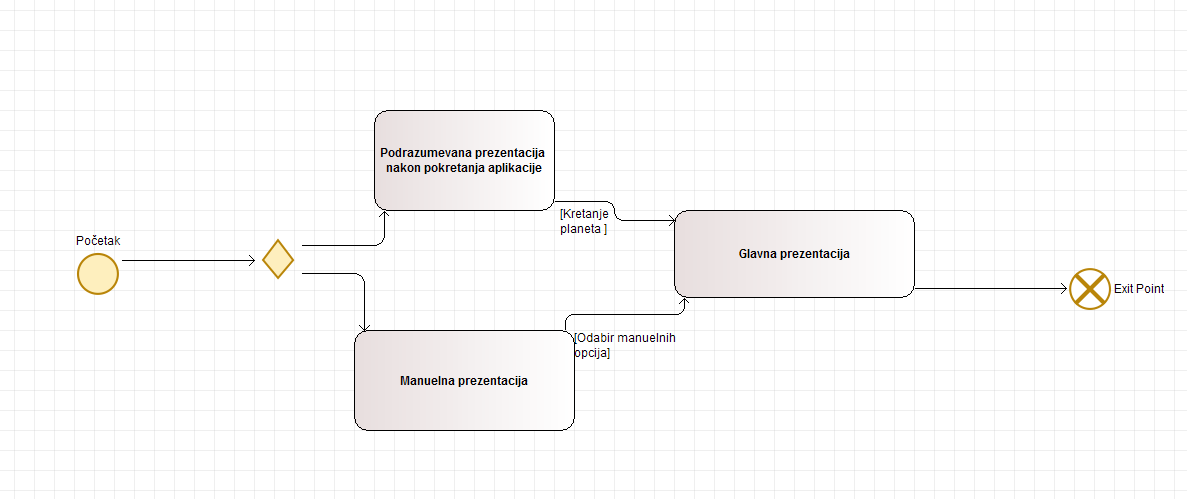
Аутомат стања:

* Понашање које специфицира секвенце стања кроз која пролази.
* Моделира понашање неког ентитета или протокол интеракције.

Дијаграм стања је граф који приказује аутомат стања:

* Чворови су стања.
* Гране су прелази.

Дијаграм стања, за дати пројектни задатак, приказан је на слици 14.

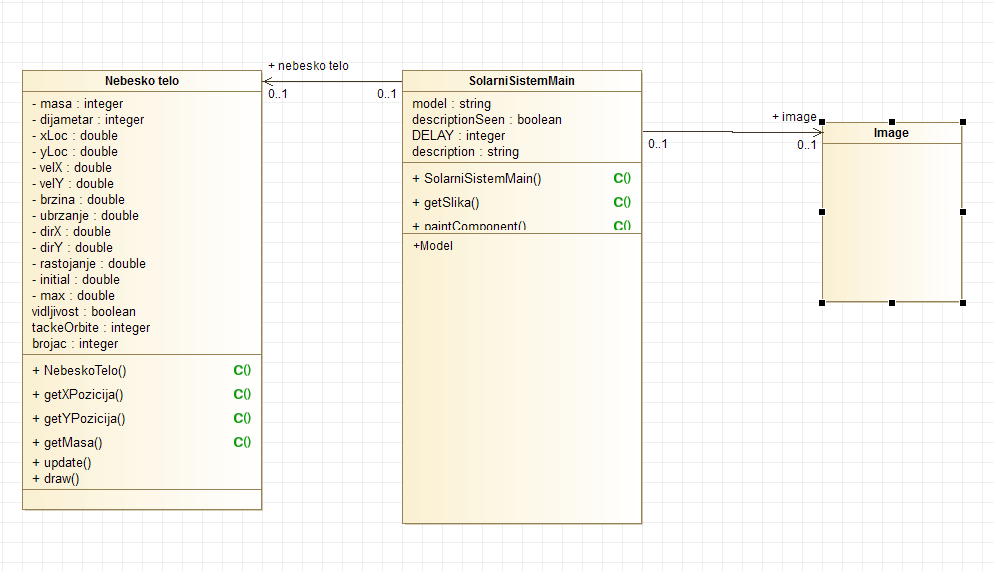
*Слика 14. Дијаграм стања*

## Дијаграм класа

Дијаграм класа показује скуп класа, интерфејса, сарадњи и других ствари структуре, повезаним релацијама.

Елементи дијаграма класа:

* Ствари: класа, интерфејс, типови, изузеци, шаблони, сарадње, пакети
* Релације: зависност, генерализације, асоцијације, реализације

*Слика 15. Дијаграм класа*

# Литература

1. <http://moodle.fink.rs/>
2. <https://docs.oracle.com/javase/tutorial/uiswing/>
3. <https://www.modelio.org/documentation-menu/tutorials.html>
4. <https://www.eclipse.org/>
5. <https://code.visualstudio.com/>
6. <https://sr.wikipedia.org/sr-ec/%D0%A1%D1%83%D0%BD%D1%87%D0%B5%D0%B2_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC>