Contents

[Uvod 2](#_Toc521676028)

[Logika 2](#_Toc521676029)

[Predstavitev šahovnice in figur na njej 2](#_Toc521676030)

[Iskanje dovoljenih polj 2](#_Toc521676031)

[Postavljanje nove kraljice 2](#_Toc521676032)

[Premikanje prejšnjih kraljic 2](#_Toc521676033)

[Grafični vmesnik 2](#_Toc521676034)

[Risanje šahovnice 2](#_Toc521676035)

[Risanje figur 4](#_Toc521676036)

[Časovnika 4](#_Toc521676037)

[Časovnik vseh rešitev 4](#_Toc521676038)

[Časovnik za posamezno rešitev 5](#_Toc521676039)

[Gumbi 5](#_Toc521676040)

# Uvod

## Problem

Na polje velikosti n x n želimo postaviti n kraljic, tako da se med seboj ne bodo napadale. V našem primeru se bomo osredotočili na polje velikost 8 x 8.

## Način iskanja rešitev

Rešitve iščemo s pomočjo sestopanja ali vračanja (ang. backtracking). To je iterativni način, ki sistematično pregleduje vse možnosti. Algoritem se drži preprostih pravil. Dokler ne postaviš 8 kraljic na šahovnico:

1. Kraljico v novi vrstici postavi v prvi prost stolpec.
2. Ko kraljice v določeni vrstici ne moreš postaviti, ker nobeno polje ni prosto, začni premikati kraljico v zgornji vrstici do prvega prostega polja. Če v koraku 2 spet ne dobiš prostega polja, ga ponovi (torej prestavi kraljico v višji vrstici). Ko korak 2 uspe, nadaljuješ s korakom 1.

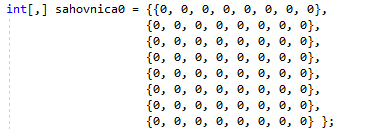
# Logika

## Predstavitev šahovnice in figur na njej

Šahovnico predstavimo z dvodimenzijskim seznamom števil, torej int[,]. Prva kompomenta predstavlja vrstice, druga pa stolpce. Obojih je 8, potekajo pa od števila 0 do 7. Šahovnica ima torej 64 polj. Vasako polje ima bodisi vrednost 0, 1, 2 ali 3. Ta števila izberemo zaradi grafičnega prikaza kraljic.

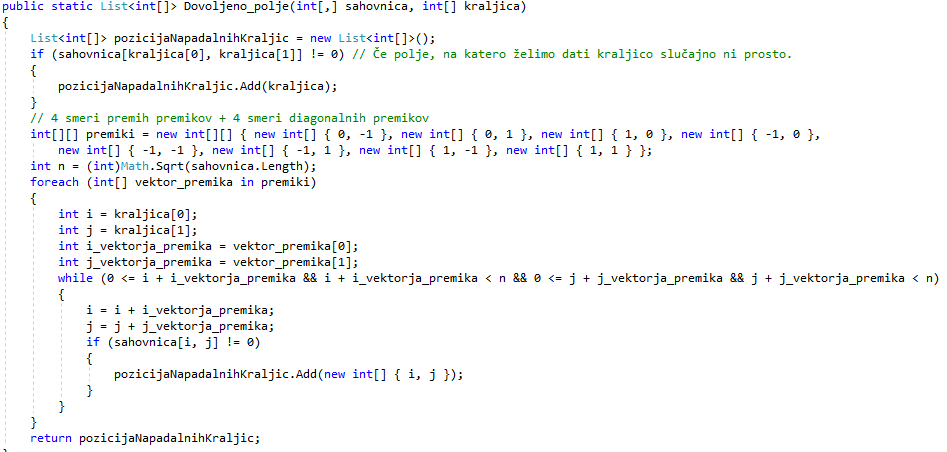
* Vrednost 0 pomeni, da na polju ni kraljice.
* Vrednost 1 pomeni, da je na polju kraljica. Prikažemo jo z belo kraljico.
* Vrednost 2 pomeni, da na polje ne smemo dati kraljice, ker je polje napadeno od ene druge že obstoječe kraljice. Na tem polju se bo prikazala rdeča kraljica.
* Vrednost 3 označuje kraljico, ki napada, torej preprečuje novi kraljici, da bi jo lahko postavili na ploščo. Takih kraljic je lahko na plošči več.

Osnovna šahovnica, na katero začnemo postavljati kraljice, je prazna.



## Iskanje dovoljenih polj

Vsako polje, na katero želimo postaviti novo kraljico, moramo preveriti. Polje je prosto, če v vrstici polja in stolpcu polja ni nobene druge kraljice. Prav tako ne sme biti kraljice na nobeni diagonali, ki ji pripada polje. Funkcija, ki išče dovoljena polja, deluje tako, da ji najprej podamo vektorje premikov kraljic. Z danega polja se pomikamo v smeri vseh vektorjev premikov tako dolgo, dokler ne zadenemo ob drugo kraljico ali ne pademo iz šahovnice. Če smo zadeli ob kraljico vemo, da je polje napadeno. Hkrati smo izvedeli za kooridnati napadajoče kraljice.



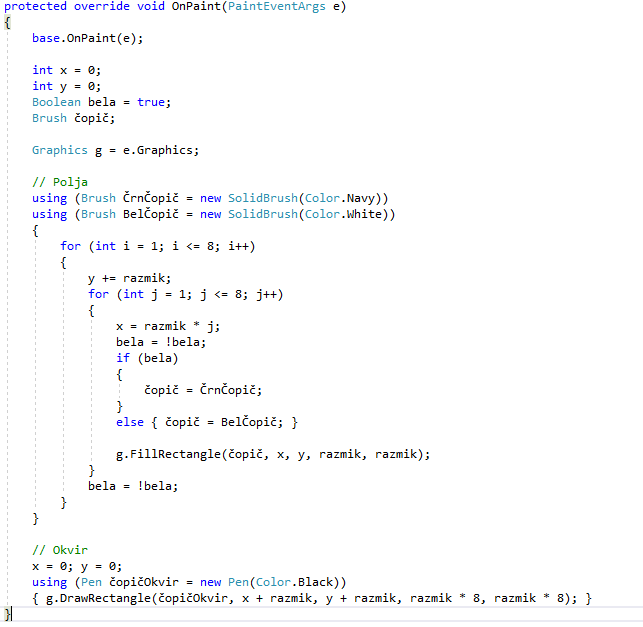
## Postavljanje nove kraljice

## Premikanje prejšnje kraljice

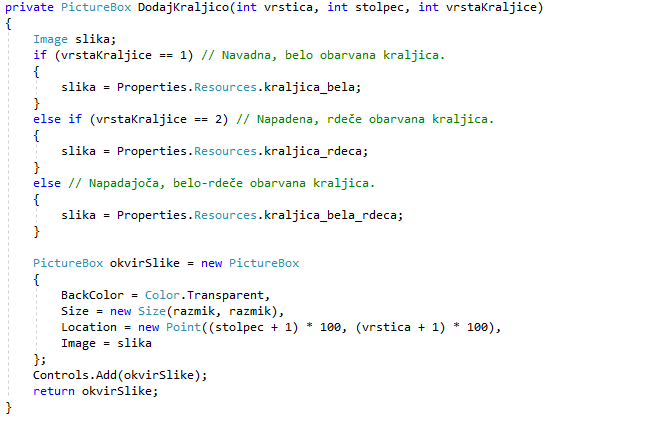
# Grafični vmesnik

## Risanje šahovnice

Šahovnico narišemo le enkrat. Rišemo jo z dvema zankama. V vrsticah izmenjujemo barve, prav tako v stolpcih.



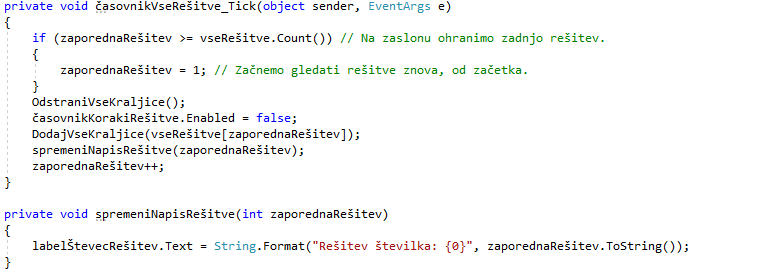
## Risanje figur



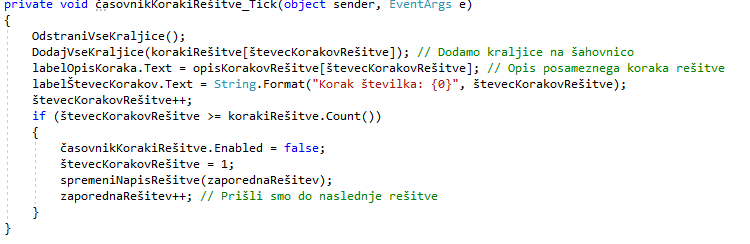
## Časovnika

Koda deluje

### Časovnik vseh rešitev



### Časovnik za posamezno rešitev



## Gumbi

# Zaključek

## Možne izboljšave

* Odpravili bi lahko utripanje, torej osveževanje vseh slik kraljic na šahovnici. Slike tistih kraljic, katerim se ne spremeni barva in pozicija, bi lahko pustili nedotaknjene, namesto da jih pobrišemo in na novo narišemo.
* Poskrbeli bi lahko, da bi se velikost polj urejala dinamično glede na velikost okna aplikacije. V manjšem oknu bi bila polja manjša.
* Problem bi lahko reševali na poljubno veliki (n x n) šahovnici.

# Viri:

* Idejo za sestopanje in algoritem sem dobil na
* Sliko bele kraljice sem pridobil na spletni strani <https://commons.wikimedia.org/wiki/File:Chess_qlt45.svg>
* Ikono za aplikacijo sem iz slike pretvoril na spletni strani <http://icoconvert.com/>