

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

**Laboratorinis darbas Nr. 3**

**Programų sistemų testavimas (T120B162)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | Darbą atliko:  Žilvinas Abromavičius IFF-4/3  Ernestas Venckus IFF-4/3  Darbą priėmė:  Doc. Šarūnas Packevičius  Dėst. Petras Tamošiūnas |

KAUNAS, 2017

Turinys

[1. Užduotis 3](#_Toc499997290)

[2. Darbo eiga 3](#_Toc499997291)

[2.1. Kodo kokybės patikrinimas naudojant Resharper įrankį 3](#_Toc499997292)

[2.2. Taisyklės kūrimas Resharper statinio testavimo įrankiui 5](#_Toc499997293)

[3. Išvados 5](#_Toc499997294)

# Užduotis

Laboratorinio darbo užduotys:

* Patikrinti kodo kokybę, naudojant Resharper statinio testavimo įrankį.
* Sukurti savo taisyklę Resharper statinio testavimo įrankyje ir išbandyti ją.

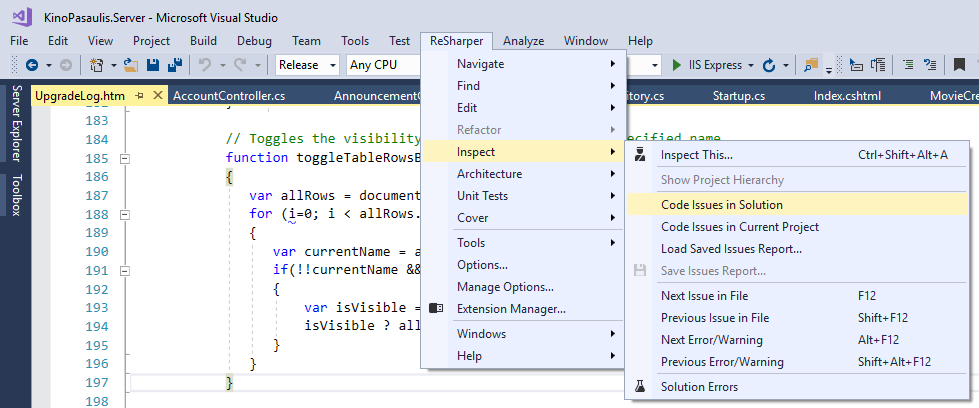
# Darbo eiga

Žemiau pateikta darbo eiga. Darbo pasidalinimas:

* Žilvinas – kodo kokybės analizė, pirmos taisyklės sukūrimas.
* Ernestas – kodo kokybės analizė, antros taisyklės sukūrimas.

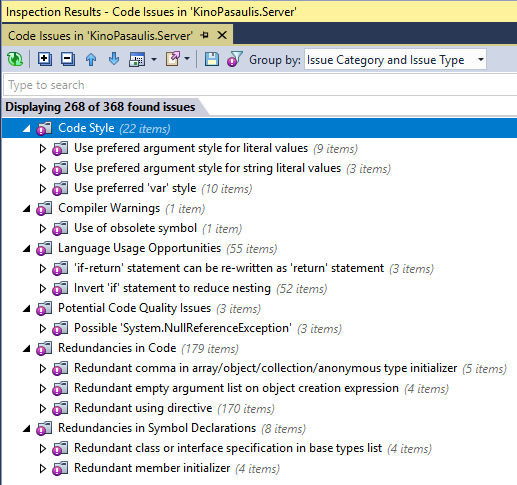
## Kodo kokybės patikrinimas naudojant Resharper įrankį

Statinė analizė paleidžiama paspaudus menu punktą „Resharper“, tada „Inspect“ ir „Code Issues in Solution“.



1 pav. Paleidžiama statinė kodo analizė

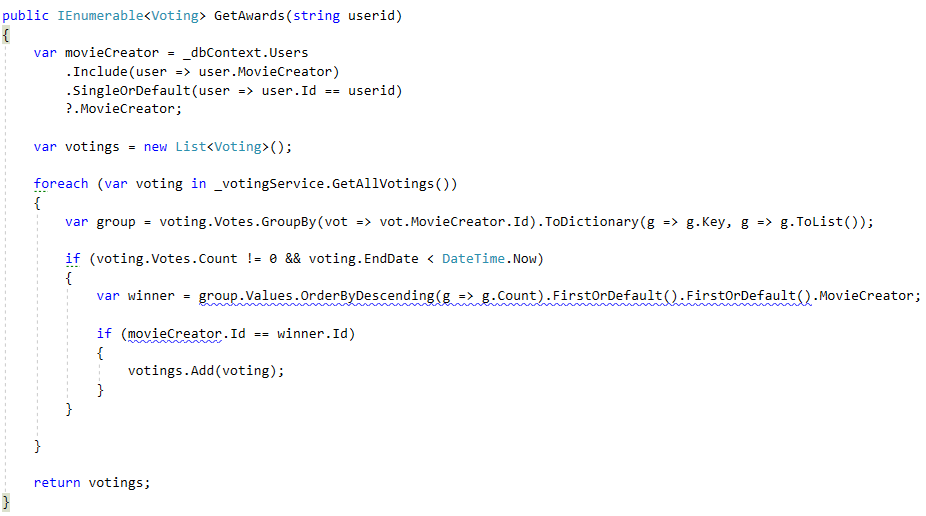
Paleidus statinę kodo analizę, buvo rastos 268 problemos.



2 pav. Rastos 268 kodo problemos

Iš 2 paveiksliuko matome, kad didžioji dauguma kodo problemų sudaro „Redundancies in Code“, konkrečiau – „Redundant using directive“, netgi 170 tokių atvejų. Ši problema nėra rimta – tiesiog faile nurodyta, kad yra naudojamas tam tikras „namespace“, kai tuo tarpu jis nėra naudojamas. Taip pat buvo rasta kitų problemų: kodo stiliaus, kompiliatoriaus pranešimų, kalbos naudojimo patobulinimo galimybių, potencialių kodo kokybės problemų, simbolių deklaracijos dubliavimo bei kodo dubliavimo.

Rastos trys potencialiai rimtos problemos: visos jos „Possible ‚System.NullReferenceException‘“. Ši problema nurodo, kad kode įmanomi scenarijai, kada gali įvykti null objekto klaida. Žemiau pateiktas vienas iš tų atvejų:



3 pav. Mėlynai pabraukta, kuriose vietose įmanoma System.NullReferenceException klaida

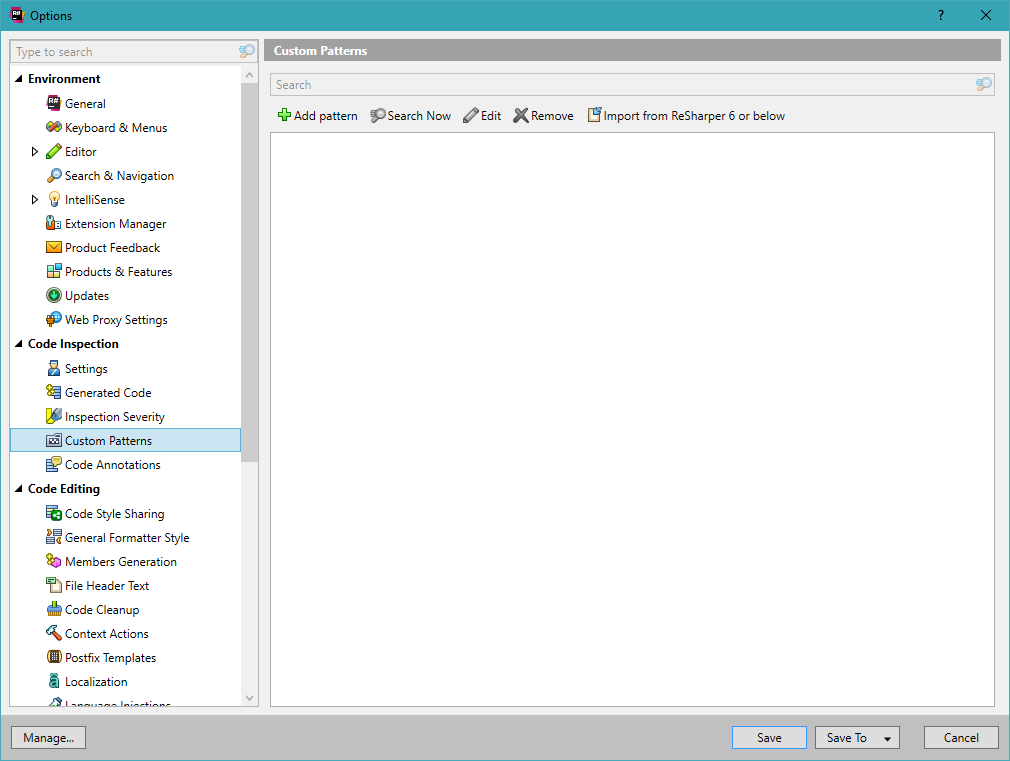
## Taisyklių kūrimas Resharper statinio testavimo įrankiui

Buvo nuspręsta sukurti dvi taisykles su statinio testavimo įrankiu Resharper.

## 

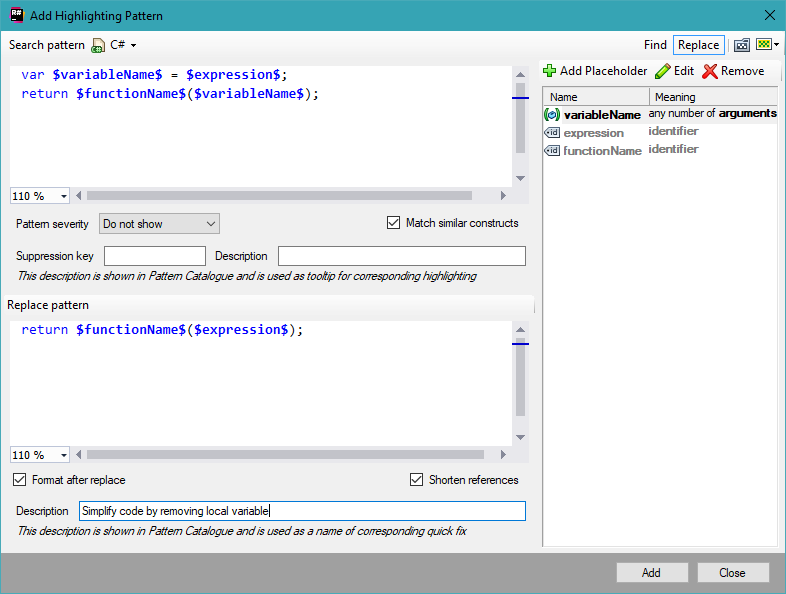
## Pirmos taisyklės kūrimas

Visual Studio 2017 programos menu juostoje nuspaudžiamas „Resharper“ ir tada „Options“. Atsidariusiame menu lange atidaromas menu punktas „Code Inspection“ ir tada „Custom Patterns“.



4 pav. Pateiktame lange bus kuriamos taisyklės

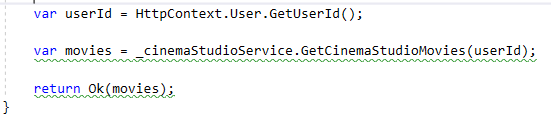
Nuspaudžiamas „Add Pattern“. Atsivėrusiame lange parašomas „Search Pattern“, kurio bus ieškoma ir „Replace Pattern“, kuriuo bus siūloma pakeisti rastą kodą. Įvedami taisyklės duomenys ir taisyklė išsaugoma.



5 pav. Pridedama taisyklė

Paspaudžiama „Add“, taisyklė tada būna pridėta. Vėliau nurodoma, taisyklės svarba – „Suggestion“.

Sukurta taisyklė sutrumpina kodą, kai yra sukuriamas lokalus kintamasis vardu $variableName$, kuris yra grąžinamas kaip funkcijos argumentas. Taisyklės veikimas pavaizduotas žemiau esančiuose paveikslėliuose: 6 paveiksliuke pavaizduota, kaip paryškinamas rastas kodas, kuriam gali būti pritaikyta taisyklė, 7 paveiksliuke matomas įvykdytas pakeitimas.

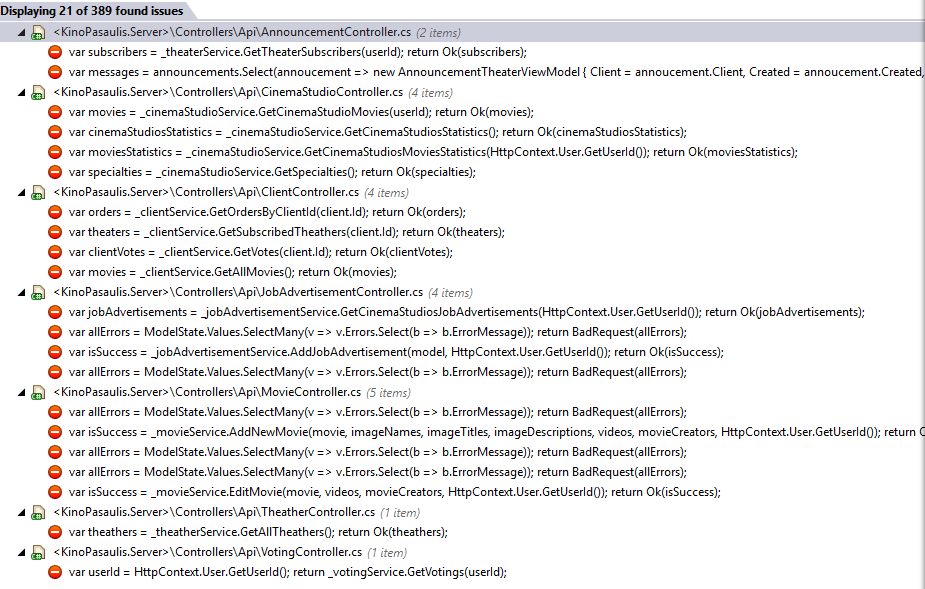


6 pav. Rasta kodo vieta, kuriai gali būti pritaikyta sukurta taisyklė



7 pav. Įvykdytas pakeitimas

Atlikus statinę kodo analizę su nauja sukurta taisykle, buvo aptikta 21 atvejai, kuriems būtų galima pritaikyti šią taisyklę.



8 pav. Sukurtos taisyklės statinės kodo analizės rezultatai

## Antros taisyklės kūrimas

Antros taisyklės kūrimas dar neįgyvendintas.

# Išvados

Laboratorinis darbas buvo atliktas sėkmingai. Išanalizavus kodą buvo aptikta įvairių smulkių problemų, tarp kurių trys potencialiai pavojingos. Buvo sukurtos kelios kodo analizės taisyklės Resharper statinio testavimo įrankiu. Jas išbandžius pastebėta, kad sėkmingai veikia.