

Informatikos fakultetas

**T120B162 Programų sistemų testavimas**

**3 laboratorinis darbas**

|  |  |  |
| --- | --- | --- |
|  | Studentai: | Gabija Skučaitė, IFF-1/7  Vytenis Kriščiūnas, IFF-1/1  Marius Žilinskas, IFF-1/8  Ignas Matulevičius, IFF-1/3 |
| Dėstytojas: | lekt. Kiudys Eligijus |

Kaunas, 2024

TURINYS

1. Įvadas 5

2. Kodo vertinimo sąrašas 6

3. Programos kodo peržiūra be įrankio 7

3.1. Vytenio 7

4. Statinė kodo analizė su įrankiu ReSharper 10

5. Sukurtos statinės kodo analizės taisyklės su ReSharper 11

6. Išvados 15

PAVEIKSLĖLIAI

1 pav. Pasirinkta klasė statinei analizei atlikti 10

2 pav.ReSharper įrankio panaudojimas 10

3 pav. Visos klaidos 11

4 pav. Rastos rašybos klaidos 11

5 pav.Sukurtos naujos statinės analizės taisyklės 12

LENTELĖS

1 lentelė. Kodo klados vertinimo lentelė 6

2 lentelė. 1 kodo peržiūros klaida. 7

3 lentelė. 2 kodo peržiūros klaida. 8

4 lentelė. 3 kodo peržiūros klaida 9

5 lentelė. 1 sukurtos statinės kodo analizės taisyklės klaida 12

6 lentelė. 2 sukurtos statinės kodo analizės taisyklės klaida 13

7 lentelė. 3 sukurtos statinės kodo analizės taisyklės klaida 14

# Įvadas

Šio laboratorinio darbo tikslas – atlikti kodo peržiūrą ir panaudoti statinės kodo analizės įrankius.

Darbo uždaviniai:

1. Peržiūrėti kodą be jokio įrankio ir rasti potencialias klaidas;
2. Atlikti statinę kodo analizę su pasirinktu įrankiu;
3. Sukurti naują taisyklę, kuri būtų panaudojama statinės kodo analizės metu;
4. Pateikti gautus rezultatus ataskaitoje;
5. Įvertinti atlikta darbą – pateikti išvadas.

# Kodo vertinimo sąrašas

1. Skaitomumas – vertinamas kodo suprantamumas, teisingas komentarų panaudojimas;
2. Klaidų valdymas – ar tinkamai tvarkomos išimtys (exceptions);
3. Efektyvumas – ar nėra neefektyvių operacijų, kurios yra nereikalingos ar sulėtintų kodo veikimą;
4. Nėra kodo dubliavimo – ar nėra pasikartojančių kodo segmentų;
5. Logika – ar nėra neteisingos logikos ir galimų klaidų.

|  |  |
| --- | --- |
| Failo pavadinimas |  |
| Klasė |  |
| Metodas |  |
| Eilutė |  |
| Pažeista taisyklė iš sąrašo |  |
| Komentaras |  |
| Kodo fragmento nuotrauka | |
|  | |

1 lentelė. Kodo klados vertinimo lentelė

# Programos kodo peržiūra be įrankio

## Vytenio

|  |  |
| --- | --- |
| Failo pavadinimas | GameClientFacade.cs |
| Klasė | GameClientFacade |
| Metodas | InitializeTank |
| Eilutė | 179 |
| Pažeista taisyklė iš sąrašo | Nėra kodo dubliavimo |
| Komentaras | Abu if‘ai yra panašūs, būtų galima iškelti logiką į atskirus metodus. (fabriko, tanko tipo ir kordinates). |
| Kodo fragmento nuotrauka | |
|  | |

2 lentelė. 1 kodo peržiūros klaida.

|  |  |
| --- | --- |
| Failo pavadinimas | GameClientFacade.cs |
| Klasė | GameClientFacade |
| Metodas | OnGameLoop |
| Eilutė | 300 |
| Pažeista taisyklė iš sąrašo | Klaidų valdymas |
| Komentaras | Šie metodai yra asinchroniniai, jei jie išmestų klaidas kodas sulūžtų. Galima būtų kodą įdėti į try ir catch blokus. |
| Kodo fragmento nuotrauka | |
|  | |

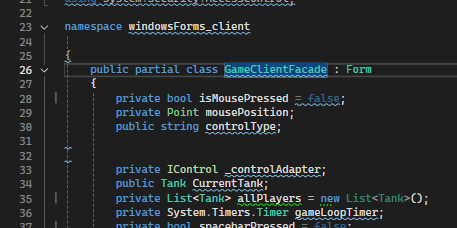
3 lentelė. 2 kodo peržiūros klaida.

|  |  |
| --- | --- |
| Failo pavadinimas | GameClientFacade.cs |
| Klasė | GameClientFacade |
| Metodas | OnKeyDown |
| Eilutė | 238 |
| Pažeista taisyklė iš sąrašo | Skaitomumas |
| Komentaras | Judėjimas į skirtingas puses yra aprašomas su keliais if‘ais. Jei jų atsirastų daugiau ar if‘ai imtų plėstis būtų sunku skaityti kodą, todėl galima sukurti atskirą metodą, kuris valdytų judėjimą. |
| Kodo fragmento nuotrauka | |
|  | |

4 lentelė. 3 kodo peržiūros klaida

# Statinė kodo analizė su įrankiu ReSharper

Galima pasirinkti norimą klasę statinei analizei atlikti.



1 pav. Pasirinkta klasė statinei analizei atlikti

A screenshot of a computer

Description automatically generated

2 pav.ReSharper įrankio panaudojimas

A screenshot of a computer

Description automatically generated

3 pav. Visos klaidos

A screenshot of a computer program

Description automatically generated

4 pav. Rastos rašybos klaidos

# Sukurtos statinės kodo analizės taisyklės su ReSharper

A screenshot of a computer

Description automatically generated

5 pav.Sukurtos naujos statinės analizės taisyklės

|  |  |
| --- | --- |
| Failo pavadinimas | Tank.cs |
| Klasė | Tank |
| Metodas | UpdateShield1 |
| Eilutė | 124 |
| Pažeista taisyklė iš sąrašo | Skaitomumas |
| Komentaras | Metodo pavadinime yra skaičius. |
| Kodo fragmento nuotrauka | |
|  | |

5 lentelė. 1 sukurtos statinės kodo analizės taisyklės klaida

|  |  |
| --- | --- |
| Failo pavadinimas | Bullet.cs |
| Klasė | Bullet |
| Metodas | Move |
| Eilutė | 56 |
| Pažeista taisyklė iš sąrašo | Skaitomumas |
| Komentaras | Metodo pavadinimas tūrėtų būti bent 5 simbolių ilgio. |
| Kodo fragmento nuotrauka | |
|  | |

6 lentelė. 2 sukurtos statinės kodo analizės taisyklės klaida

|  |  |
| --- | --- |
| Failo pavadinimas | GameClientFacade.cs |
| Klasė | GameClientFacade |
| Metodas | InitializeTank |
| Eilutė | 176 |
| Pažeista taisyklė iš sąrašo | Skaitomumas |
| Komentaras | Panaudota per daug argumentų. Reikėtų naudoti mažiau nei 3. |
| Kodo fragmento nuotrauka | |
|  | |

7 lentelė. 3 sukurtos statinės kodo analizės taisyklės klaida

# Išvados

1. Rastos potencialios klaidos kodo peržiūrėjimo metu;
2. Galima surasti daug daugiau potencialių klaidų kode naudojant statinės analizės įrankius.
3. Pavyko parašyti kelias statinės analizės taisykles ir integruoti jas į statinės kodo analizės įrankį;
4. Pavyko dokumentuoti visą atliktą veiklą ir paruošti ataskaitą.