



INFORMATIKOS FAKULTETAS

T120B162 Programų sistemų testavimas

3 laboratorinis darbas

Studentas: Gabija Šeškauskaitė, IFF-0/2

Dėstytojas: lekt. Dominykas Barisas

KAUNAS 2023

Turinys

1. Įvadas	3
2. Programos kodo peržiūros eiga.....	3
3. Programos kodo peržiūra	4
4. Priemonės kodo statinei analizei atlikti.....	8
5. Programos kodo statinė analizė.....	8
6. Naujos taisyklės įtraukimas į kodo statinės analizės įrankį	11
7. Išvados	11

1. Įvadas

3 laboratorinio darbo tikslas – išmokti atlikti tikslingą programinio kodo peržiūrą bei pritaikyti statinės kodo analizės įrankius savo projekte.

Darbo uždaviniai:

- 1) Pritaikyti statinę projekto kodo analizę be jokio įrankio – peržiūrėti programos kodą ir rasti potencialias klaidas;
- 2) Susirasti tinkamus statinės kodo analizės įrankius kuriamam projektui;
- 3) Pritaikyti statinės kodo analizės įrankius programos kodui ir identifikuoti programos kodo klaidas;
- 4) Parašyti naują taisyklę, kuri būti pritaikoma statiška nagrinėjant kodą;
- 5) Statinės kodo analizės rezultatus pateikti laboratorinio darbo ataskaitoje.

2. Programos kodo peržiūros eiga

Pirmiausia yra peržiūrimas programos kodas nenaudojant jokių įrankių. Šis procesas vadinamas kodo peržiūra. Tam, kad kodo peržiūra būtų nuosekli, vadovaujamasi šiais kodo peržiūros punktais:

- 1) Kodo formatavimas – ar kodas rašomas nuosekliai, ar tinkami kodo eilučių atitraukimai nuo kairiojo krašto, ar yra taikomos teisingos kodo rašymo taisyklės;
- 2) Kodo architektūra – ar kodas yra išskaidytas į komponentus arba failus (pvz., stiliaus ir scenarijų kalbų failai nėra saugomi viename faile);
- 3) Kodo gerosios praktikos – ar nėra kietai įsiuto kodo, ar naudojamos konstantos vietoj įkoduotų bereikšmių skaičių (pvz., `if (role == 1)`), ar rašomi tikslingi komentarai, ar vengiama didelių `if/else` blokų, ar tinkamai išnaudojamos projekte naudojamų karkasų galimybės ir pan;
- 4) Kodo saugumas – ar kodas yra apsaugotas nuo SQL injekcijų, cross-side scripting, ar tie programos duomenys/rezultatai, kurie neturėtų būti matomi, nėra išvedami į konsolę/naršyklės langą, ar tikrinamos įvedimo laukų reikšmės (ar yra taikoma laukų validacija).

Taip pat tam, kad būtų išlaikomas nuoseklumas, programos kodo peržiūros metu rastos klaidos bus dokumentuojamos tokiomis lentelėmis (1 lentelė):

1 lentelė. Kodo peržiūros metu rastos klaidos dokumentavimo lentelės šablonas

Failo pavadinimas	
Programos kodo fragmento nuotrauka	
Failo kodo eilutė(s)	
Klaidos kategorija (iš sąrašo)	
Klaidos apibūdinimas	
Galimas sprendimas/komentaras	

3. Programos kodo peržiūra

Atlikus kodo peržiūrą, programos kode buvo rasta keletas klaidų/pastebėjimų. Šios klaidos pateiktos žemiau esančiose 2 – 10 lentelėse:

2 lentelė. Kodo peržiūros klaida Nr. 1

Failo pavadinimas	Game.cs
Programos kodo fragmento nuotrauka	
<pre> 3 4 namespace TicTacToe.GameObjects 5 { 6 public class Game : IPrototype<Game> 7 { 8 public bool isFirstPlayersTurn; 9 } </pre>	
Failo kodo eilutė(s)	6
Klaidos kategorija (iš sąrašo)	Kodo geroji praktika/ Kodo saugumas
Klaidos apibūdinimas	Public yra tiesioginė prieiga prie klasės viduje esančios būsenos, kas gali sukelti problemų.
Galimas sprendimas/komentaras	Pakeiskite į private bool isFirstPlayersTurn; arba suteikite kontrolę per savybę.

3 lentelė. Kodo peržiūros klaida Nr. 2

Failo pavadinimas	Game.cs
Programos kodo fragmento nuotrauka	

<pre> 34 public Game(GameFactory gameFactory1, GameFactory gameFactory2, Player player1, Player player2, string roomName, int boardSize, bool obstacles) 35 { 36 Player1 = player1; 37 Player2 = player2; 38 GameRoomName = roomName; 39 ToggleObstacles = obstacles; 40 Board = BoardCreator.factoryMethod(boardSize); 41 42 isFirstPlayersTurn = true; 43 44 // Link the players to the game as well 45 Player1.PlayingRoomName = GameRoomName; 46 Player1.Piece = gameFactory1.CreatePiece(player1).ToString(); 47 48 if (Player2 != null) 49 { 50 Player2.PlayingRoomName = GameRoomName; 51 Player2.Piece = gameFactory2.CreatePiece(player2).ToString(); 52 } 53 } </pre>	
Failo kodo eilutė(s)	36
Klaidos kategorija (iš sąrašo)	Kodo saugumas
Klaidos apibūdinimas	Neatliekamas tikrinimas, ar Player2 yra null prieš bandant jį naudoti.
Galimas sprendimas/komentaras	Pridėkite patikrinimą, kad išvengtumėte null reference exception.

4 lentelė. Kodo peržiūros klaida Nr. 3

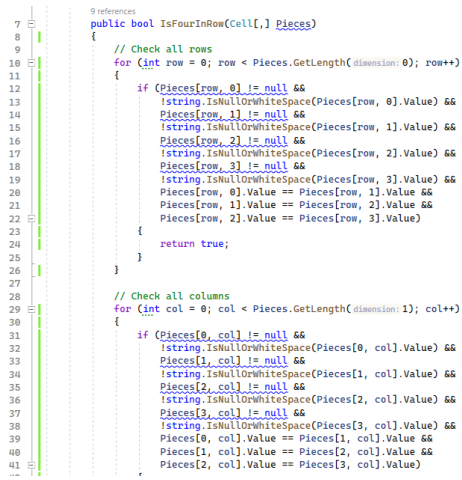
Failo pavadinimas	Board3.cs
Programos kodo fragmento nuotrauka	
<pre> 4 namespace TicTacToe.GameObjects; 5 6 public class Board3 : Board 7 { 8 public IWinningStrategy winningStrategy; 9 } </pre>	
Failo kodo eilutė(s)	6
Klaidos kategorija (iš sąrašo)	Kodo geroji praktika/ Kodo saugumas
Klaidos apibūdinimas	Public yra tiesioginė prieiga prie vidinės būsenos, kuri gali sukelti problemų.
Galimas sprendimas/komentaras	Pakeiskite į privačią savybę ir suteikite prieigą per savybę.

5 lentelė. Kodo peržiūros klaida Nr. 4

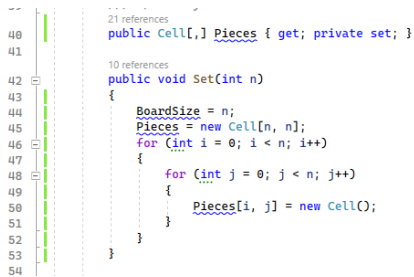
Failo pavadinimas	Cell.cs
Programos kodo fragmento nuotrauka	
<pre> 1 namespace TicTacToe.GameObjects 2 { 3 public class Cell 4 { 5 public string Value { get; private set; } 6 } 7 } </pre>	
Failo kodo eilutė(s)	3

Klaidos kategorija (iš sąrašo)	Kodo geroji praktika/Kodo saugumas
Klaidos apibūdinimas	Nesvarbu, ar Value yra viešas ar privatus, nes Cell klasė yra immutable (nekeičiama), tačiau galite padaryti tik get'e ir panaikinti Set metodą.
Galimas sprendimas/komentaras	Pakeiskite į public string Value { get; }.

6 lentelė. Kodo peržiūros klaida Nr. 5

Failo pavadinimas	FourByFourWinningStrategy.cs
Programos kodo fragmento nuotrauka	
 <pre> 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 -- 9 references public bool IsFourInRow(Cell[,] Pieces) { // Check all rows for (int row = 0; row < Pieces.GetLength(dimension: 0); row++) { if (Pieces[row, 0] != null && !string.IsNullOrEmpty(Pieces[row, 0].Value) && Pieces[row, 1] != null && !string.IsNullOrEmpty(Pieces[row, 1].Value) && Pieces[row, 2] != null && !string.IsNullOrEmpty(Pieces[row, 2].Value) && Pieces[row, 3] != null && !string.IsNullOrEmpty(Pieces[row, 3].Value) && Pieces[row, 0].Value == Pieces[row, 1].Value && Pieces[row, 1].Value == Pieces[row, 2].Value && Pieces[row, 2].Value == Pieces[row, 3].Value) { return true; } } // Check all columns for (int col = 0; col < Pieces.GetLength(dimension: 1); col++) { if (Pieces[0, col] != null && !string.IsNullOrEmpty(Pieces[0, col].Value) && Pieces[1, col] != null && !string.IsNullOrEmpty(Pieces[1, col].Value) && Pieces[2, col] != null && !string.IsNullOrEmpty(Pieces[2, col].Value) && Pieces[3, col] != null && !string.IsNullOrEmpty(Pieces[3, col].Value) && Pieces[0, col].Value == Pieces[1, col].Value && Pieces[1, col].Value == Pieces[2, col].Value && Pieces[2, col].Value == Pieces[3, col].Value) { return true; } } return false; } </pre>	
Failo kodo eilutė(s)	7-80
Klaidos kategorija (iš sąrašo)	Kodo architektūra
Klaidos apibūdinimas	IsFourInRow metodas per ilgas ir sudėtingas.
Galimas sprendimas/komentaras	Padalinkite metodą į smulkesnius, lengviau suprantamus metodus.

7 lentelė. Kodo peržiūros klaida Nr. 6

Failo pavadinimas	Board.cs
Programos kodo fragmento nuotrauka	
 <pre> 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 -- 21 references public Cell[,] Pieces { get; private set; } 10 references public void Set(int n) { BoardSize = n; Pieces = new Cell[n, n]; for (int i = 0; i < n; i++) { for (int j = 0; j < n; j++) { Pieces[i, j] = new Cell(); } } } </pre>	
Failo kodo eilutė(s)	40

Klaidos kategorija (iš sąrašo)	Kodo gerosios praktikos
Klaidos apibūdinimas	Nereikalingas masyvo inicializavimas konstruktoriuje, nes vėliau jis yra perrašomas Set metode.
Galimas sprendimas/komentaras	Ištrinkite šią eilutę konstruktoriuje.

8 lentelė. Kodo peržiūros klaida Nr. 8

Failo pavadinimas	Player.cs
Programos kodo fragmento nuotrauka	
Failo kodo eilutė(s)	49
Klaidos kategorija (iš sąrašo)	Kodo geroji praktika
Klaidos apibūdinimas	Nėra patikrinimo dėl null palyginimo Equals metode.
Galimas sprendimas/komentaras	Ištraukite patikrinimą dėl null palyginimo, kad išvengtumėte NullReferenceException. Pvz., return Id?.Equals(other.Id) ?? false && Name?.Equals(other.Name) ?? false;.

8 lentelė. Kodo peržiūros klaida Nr. 9

Failo pavadinimas	Subject.cs
Programos kodo fragmento nuotrauka	
Failo kodo eilutė(s)	40
Klaidos kategorija (iš sąrašo)	Kodo architektūra
Klaidos apibūdinimas	Subject klasė neturi jokios veikiančios Update metodo implementacijos, todėl Notify metodas neveiks kaip tikimasi.
Galimas sprendimas/komentaras	Implementuokite veikiantį Update metodą.

4. Priemonės kodo statinei analizei atlikti

Statinės kodo analizės įrankiu yra testuojamas tic tac toe žaidimas, todėl statinei kodo analizei pasirinktas tinkamas įrankis – StyleCop. StyleCop yra atvirojo kodo statinio kodo analizės įrankis iš Microsoft, kuris tikrina C# kodą, ar jis atitinka StyleCop rekomenduojamus kodavimo stilius ir Microsoft .NET Framework projektavimo gairių pogrupį. Jį galima atsisiųsti iš NuGet packages.

5. Programos kodo statinė analizė

Panaudojus statinės kodo analizės įrankį StyleCop buvo rasti 99+ įspėjimai.

	Code	Description	Project	File	Line	Suppression State
▶	SA1633	The file header is missing or not located at the top of the file.	TicTacToe	BoardCreator.cs	1	Active
▶	SA1200	Using directive should appear within a namespace declaration	TicTacToe	BoardCreator.cs	1	Active
▶	SA1507	Code should not contain multiple blank lines in a row	TicTacToe	BoardCreator.cs	4	Active
▶	SA1600	Elements should be documented	TicTacToe	BoardCreator.cs	6	Active
▶	SA1505	An opening brace should not be followed by a blank line	TicTacToe	BoardCreator.cs	7	Active
▶	SA1629	Documentation text should end with a period	TicTacToe	BoardCreator.cs	10	Active
▶	SA1629	Documentation text should end with a period	TicTacToe	BoardCreator.cs	12	Active
▶	SA1616	Element return value documentation should have text	TicTacToe	BoardCreator.cs	13	Active
▶	SA1300	Element 'factoryMethod' should begin with an uppercase letter	TicTacToe	BoardCreator.cs	14	Active
▶	SA1000	The keyword 'if' should be followed by a space	TicTacToe	BoardCreator.cs	16	Active
▶	CS8603	Possible null reference return.	TicTacToe	BoardCreator.cs	26	Active

1 pav. Failo BoardCreator klaidos

▶	SA1633	The file header is missing or not located at the top of the file.	TicTacToe	FourByFourWinningStrate...	1	Active
▶	SA1507	Code should not contain multiple blank lines in a row	TicTacToe	FourByFourWinningStrate...	81	Active
▶	SA1513	Closing brace should be followed by blank line	TicTacToe	FourByFourWinningStrate...	94	Active
▶	SA1101	Prefix local calls with this	TicTacToe	FourByFourWinningStrate...	100	Active
▶	SA1101	Prefix local calls with this	TicTacToe	FourByFourWinningStrate...	100	Active
▶	SA1513	Closing brace should be followed by blank line	TicTacToe	FourByFourWinningStrate...	101	Active

2 pav. FourByFourWinningStrategy klaidos

▶	SA1507	Code should not contain multiple blank lines in a row	TicTacToe	Board3.cs	4	Active
▶	SA1600	Elements should be documented	TicTacToe	Board3.cs	6	Active
▶	SA1600	Elements should be documented	TicTacToe	Board3.cs	8	Active
▶	SA1401	Field should be private	TicTacToe	Board3.cs	8	Active
▶	SA1307	Field 'winningStrategy' should begin with upper-case letter	TicTacToe	Board3.cs	8	Active
▶	SA1600	Elements should be documented	TicTacToe	Board3.cs	10	Active
▶	SA1101	Prefix local calls with this	TicTacToe	Board3.cs	12	Active
▶	SA1101	Prefix local calls with this	TicTacToe	Board3.cs	13	Active
▶	SA1600	Elements should be documented	TicTacToe	Board3.cs	16	Active
▶	SA1101	Prefix local calls with this	TicTacToe	Board3.cs	20	Active
▶	SA1101	Prefix local calls with this	TicTacToe	Board3.cs	20	Active
▶	SA1600	Elements should be documented	TicTacToe	Board3.cs	24	Active

3 pav. Failo Board3 klaidos

▷	SA1633	The file header is missing or not located at the top of the file.	TicTacToe	Game.cs	1	Active
▷	SA1200	Using directive should appear within a namespace declaration	TicTacToe	Game.cs	1	Active
▷	SA1200	Using directive should appear within a namespace declaration	TicTacToe	Game.cs	2	Active
▷	SA1600	Elements should be documented	TicTacToe	Game.cs	6	Active
▷	SA1600	Elements should be documented	TicTacToe	Game.cs	8	Active
▷	SA1307	Field 'isFirstPlayersTurn' should begin with upper-case letter	TicTacToe	Game.cs	8	Active
▷	SA1401	Field should be private	TicTacToe	Game.cs	8	Active
▷	SA1611	The documentation for parameter 'player1Factory' is missing	TicTacToe	Game.cs	15	Active
▷	SA1611	The documentation for parameter 'roomName' is missing	TicTacToe	Game.cs	15	Active
▷	SA1611	The documentation for parameter 'boardSize' is missing	TicTacToe	Game.cs	15	Active
▷	SA1611	The documentation for parameter 'obstacles' is missing	TicTacToe	Game.cs	15	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	17	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	18	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	19	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	20	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	22	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	25	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	25	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	26	Active
▷	SA1611	The documentation for parameter 'gameFactory1' is missing	TicTacToe	Game.cs	34	Active
▷	SA1611	The documentation for parameter 'gameFactory2' is missing	TicTacToe	Game.cs	34	Active

▷	SA1611	The documentation for parameter 'roomName' is missing	TicTacToe	Game.cs	34	Active
▷	SA1611	The documentation for parameter 'boardSize' is missing	TicTacToe	Game.cs	34	Active
▷	SA1611	The documentation for parameter 'obstacles' is missing	TicTacToe	Game.cs	34	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	36	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	37	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	38	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	39	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	40	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	42	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	45	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	45	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	46	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	48	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	50	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	50	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	51	Active
▷	SA1106	Code should not contain empty statements	TicTacToe	Game.cs	51	Active
▷	SA1107	Code should not contain multiple statements on one line	TicTacToe	Game.cs	51	Active
▷	SA1005	Single line comment should begin with a space	TicTacToe	Game.cs	58	Active
▷	SA1623	The property's documentation summary text should begin with: 'Gets or sets'	TicTacToe	Game.cs	63	Active
▷	SA1623	The property's documentation summary text should begin with: 'Gets or sets'	TicTacToe	Game.cs	68	Active

▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	130	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	144	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	145	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	146	Active
▷	SA1507	Code should not contain multiple blank lines in a row	TicTacToe	Game.cs	147	Active
▷	SA1600	Elements should be documented	TicTacToe	Game.cs	155	Active
▷	SA1116	The parameters should begin on the line after the declaration, whenever the parameter span across multiple lines	TicTacToe	Game.cs	157	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	158	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	158	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	158	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	158	Active
▷	SA1600	Elements should be documented	TicTacToe	Game.cs	161	Active
▷	SA1600	Elements should be documented	TicTacToe	Game.cs	166	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	170	Active
▷	CS8602	Dereference of a possibly null reference.	TicTacToe	Game.cs	170	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	170	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	170	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	172	Active
▷	CS8602	Dereference of a possibly null reference.	TicTacToe	Game.cs	172	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	172	Active
▷	SA1101	Prefix local calls with this	TicTacToe	Game.cs	172	Active

4 pav. Game klaidos

▷	▲	SA1633	The file header is missing or not located at the top of the file.	TicTacToe	Cell.cs	1	Active
▷	▲	SA1600	Elements should be documented	TicTacToe	Cell.cs	3	Active
▷	▲	SA1600	Elements should be documented	TicTacToe	Cell.cs	5	Active
▷	▲	SA1642	Constructor summary documentation should begin with standard text	TicTacToe	Cell.cs	7	Active
▷	▲	SA1629	Documentation text should end with a period	TicTacToe	Cell.cs	8	Active
▷	▲	SA1201	A constructor should not follow a property	TicTacToe	Cell.cs	10	Active
▷	▲	SA1101	Prefix local calls with this	TicTacToe	Cell.cs	12	Active
▷	▲	SA1122	Use string.Empty for empty strings	TicTacToe	Cell.cs	12	Active
▷	▲	SA1642	Constructor summary documentation should begin with standard text	TicTacToe	Cell.cs	15	Active
▷	▲	SA1629	Documentation text should end with a period	TicTacToe	Cell.cs	16	Active
▷	▲	SA1614	Element parameter documentation should have text	TicTacToe	Cell.cs	18	Active
▷	▲	SA1101	Prefix local calls with this	TicTacToe	Cell.cs	21	Active
▷	▲	SA1629	Documentation text should end with a period	TicTacToe	Cell.cs	25	Active
▷	▲	SA1629	Documentation text should end with a period	TicTacToe	Cell.cs	27	Active
▷	▲	SA1101	Prefix local calls with this	TicTacToe	Cell.cs	30	Active
▷	▲	SA1507	Code should not contain multiple blank lines in a row	TicTacToe	Cell.cs	32	Active
▷	▲	SA1629	Documentation text should end with a period	TicTacToe	Cell.cs	35	Active
▷	▲	SA1629	Documentation text should end with a period	TicTacToe	Cell.cs	37	Active
▷	▲	SA1300	Element 'getStatus' should begin with an uppercase letter	TicTacToe	Cell.cs	38	Active
▷	▲	SA1629	Documentation text should end with a period	TicTacToe	Cell.cs	44	Active

5 pav. Failo Cell klaidos

▷	▲	CS0108	'Piece.getStatus()' hides inherited member 'Cell.getStatus()'. Use the new keyword if hiding was intended.	TicTacToe	Piece.cs	16	Active
▷	▲	SA1633	The file header is missing or not located at the top of the file.	TicTacToe	Piece.cs	1	Active
▷	▲	SA1600	Elements should be documented	TicTacToe	Piece.cs	3	Active
▷	▲	SA1505	An opening brace should not be followed by a blank line	TicTacToe	Piece.cs	4	Active
▷	▲	SA1600	Elements should be documented	TicTacToe	Piece.cs	6	Active
▷	▲	SA1128	Put constructor initializers on their own line	TicTacToe	Piece.cs	6	Active
▷	▲	SA1101	Prefix local calls with this	TicTacToe	Piece.cs	8	Active
▷	▲	SA1600	Elements should be documented	TicTacToe	Piece.cs	11	Active
▷	▲	SA1101	Prefix local calls with this	TicTacToe	Piece.cs	13	Active
▷	▲	SA1600	Elements should be documented	TicTacToe	Piece.cs	16	Active
▷	▲	SA1300	Element 'getStatus' should begin with an uppercase letter	TicTacToe	Piece.cs	16	Active

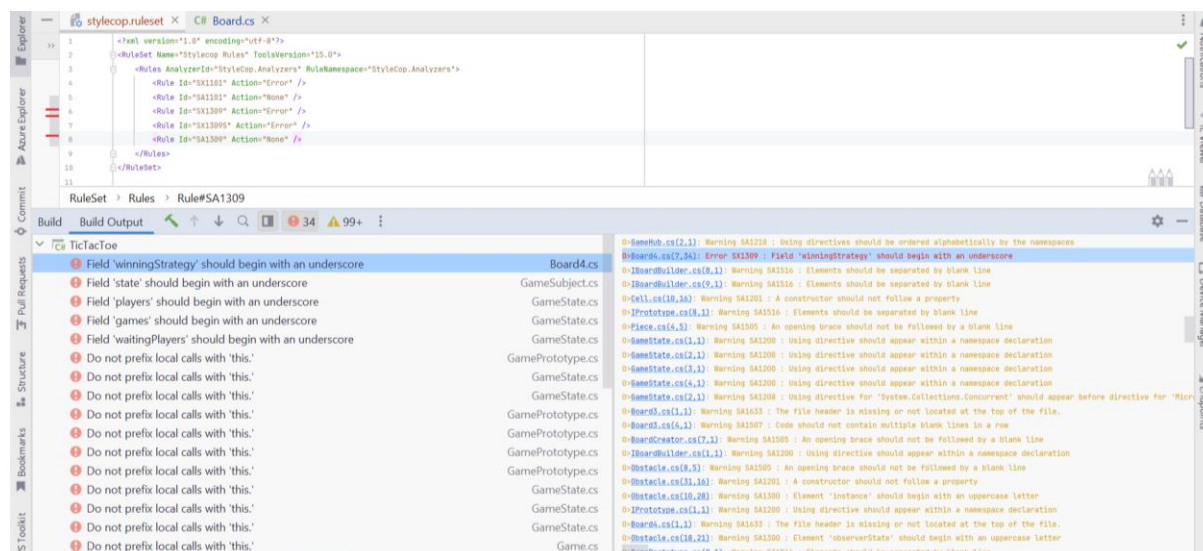
6 pav. Failo Piece klaidos

6. Naujos taisyklės įtraukimas į kodo statinės analizės įrankį

Naujos taisyklės įtraukimui buvo naudotas StyleCop.

Buvo sukurtas stylecop.ruleset ir jis taip pat buvo įrašytas į .csproj failą.

Įtraukta taisyklė buvo nustatyta kaip klaida.



7 pav. Naujos taisyklės įtraukimas

7. Išvados

- 1) Laboratorinio darbo metu buvo peržiūrėtas programos kodas bei surastos potencialios klaidos bei rastos taisytinos kodo vietos, pritaikant gerąsias programavimo praktikas;
- 2) Surasti statinės kodo analizės įrankiai, kurie, kaip parodė analizės procesas, buvo tinkami kuriamam projektui;
- 3) Surastas statinės kodo analizės įrankis buvo pritaikytas programos kodui, buvo identifikuotos, o vėliau ir ištaisytos surastos programos kodo klaidos;
- 4) Parašyta nauja taisyklė, kuri vėliau buvo integruota į statinės kodo analizės įrankį;
- 5) Paruošta laboratorinio darbo ataskaita.