



Kauno technologijos universitetas

Informatikos Fakultetas

Programų Sistemų Testavimas (T120B162)

3-iojo laboratorinio darbo ataskaita

Atliko IFF-2/1 grupės studentai:

Rokas Gudžiūnas

Gedmantas Šilinskas

Rugilė Jovaišaitė

Simona Gerikaitė

Priėmė:

doc. prakt. Guogis Evaldas

KAUNAS, 2024

Turiny

Laboratorinio darbo tikslas.....	3
Laboratorinio darbo uždutys	3
Aprašymas	3
Naudojamos priemonės	3
Analizės rezultatai	3
Gautų problemų aiškinimasis	4
Išvados.....	7

Laboratorinio darbo tikslas

Šio laboratorinio darbo tikslas yra įvertinti mūsų testuojamos programos kokybę, naudojant statinio analizavimo įrankį.

Laboratorinio darbo užduotys

1. Pasirinkti tinkamą įrankį statinei kodo analizei.
2. Paleisti kodo analizės įrankį ant mūsų programinio projekto.
3. Įvertinti analizės rezultatus ir identifikuoti galimas kodo vietas, kurios gali sukelti problemų.

Aprašymas

Programinės įrangos statinė analizė – tai procesas, kurio tikslas yra identifikuoti galimas klaidas kodo nagrinėjimo metu, o ne vykdymo. Ši analizė leidžia atpažinti klaidas, kurios gali būti susijusios su kodo rašymo praktika, potencialiomis saugumo spragomis ar netinkamu kodo struktūrizavimu. Statinė analizė atliekama pagal ilgą kontrolinį sąrašą, kuris gali būti automatizuotas, pasitelkus analizės įrankį.

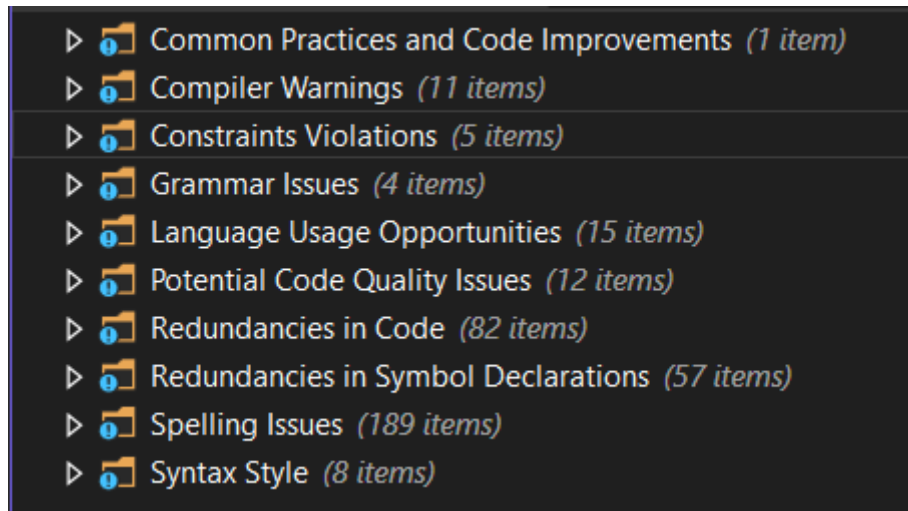
Naudojamos priemonės

Mūsų projekto statinei kodo analizei pasirinkome JetBrains sukurtą „ReSharper“ įrankį, kuris sklandžiai integruojasi su Visual Studio ir yra vienas geriausių rinkoje, nes jame yra didelis iš anksto nustatytų kodo tikrinimo taisyklių rinkinys, kuris gali aptikti įvairiausias problemas, tad nėra būtinybės kurti papildomas savas taisykles. Taip pat šis įrankis savyje turi įdiegtą dirbitinio intelekto asistentą, kuris dar labiau palengvina kodo analizę ir leidžia nustatyti aibę galimų netikslumų kode, kurios gali pakenkti jo kokybei ir funkcionalumui.

Analizės rezultatai

Išanalizavus projektą, ReSharper pateikė galimų problemų sąrašą, matomą 1 paveiksle, ir suklasifikavo pagal tipus:

- Bendrosios praktikos ir kodo gerinimo galimybės
- Kompiliatoriaus įspėjimai
- Apribojimų pažeidimai
- Gramatinės klaidos
- Kalbos naudojimo galimybės
- Potencialios kodo kokybės problemos
- Kodo pasikartojimai
- Simbolių deklaracijų pasikartojimai
- Rašybos klaidos
- Sintaksės stilius



1 pav. Analizės rezultatai

Gautų problemų aiškinimasis

Bendrosios praktikos ir kodo gerinimo galimybės – ReSharper įrankis pažymėjo eilutę „public async static“, nes norėtų, kad vietoje „async“ naudotume „static“, tačiau šis metodas buvo automatiškai sugeneruotas ABP, tad jo nekeisime.

```
0 references
public async static Task<int> Main(string[] args)
{
    Log.Logger = new LoggerConfiguration()
    #if DEBUG
```

Kompiliatoriaus įspėjimai – dėl šio įspėjimo nieko nekeisime, nes niekados nebus taip, jog gali atsirasti „null reference exceptions“.

```
8 namespace Gymzi.Exercises,
27 references
9 public class Exercise : AuditedAggregateRoot<Guid>
10 {
11     11 references
    public string Name { get; set; }
12     5 references
    string Exercise.Name { get; set; }
    CS8618: Non-nullable property 'Name' must contain a non-null value when exiting constructor. Consider declaring the property as nullable.
    Non-nullable property 'Name' is uninitialized. Consider declaring the property as nullable.
    Show potential fixes (Alt+Enter or Ctrl+.)
```

Gramatinės klaidos – nors rastos klaidos nedaro įtakos mūsų kodo funkcionalumui, o komentarai yra suprantami, jas pataisėme.

```

2 references
3 public static class MultiTenancyConsts
4 {
5     /* Enable/disable multi-tenancy easily in a single point.
6      * If you will never need to multi-tenancy, you can remove
7      * related modules and
8      */
9     public const bool IsEn
10 }
11

```

Consider removing 'will'. (Usually, 'will' does not occur in a conditional clause, unless in the sense 'want to' or 'be willing to'.)

Incorrect: If this **will not go** right, we will have to do it again.
 Correct: If this **does not go** right, we will have to do it again.

Powered by LanguageTool

```

//auto configure for the base class props
me).IsRequ

```

This word is normally spelled as one.

Incorrect: I now have a **part time** job.
 Correct: I now have a **part-time** job.

Powered by LanguageTool

Kalbos naudojimo galimybės – šie ReSharper įrankio teikiami pasiūlymai nesuteikia jokios papildomos naudos, tad nieko nekeisime.

```

using (var scope = context.ServiceProvider.CreateScope())
{
    void System.IDisposable.Dispose()
    Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
}

```

IDE0063: 'using' statement can be simplified

IDE0063: 'using' statement can be simplified

Convert into 'using' declaration

Show potential fixes (Alt+Enter or Ctrl+.)

Potencialios kodo kokybės problemos – šios klasės buvo automatiškai sugeneruotos ABP, dėl ko jų nekeisime, nes paryškintos vietos nebuvo rašytos mūsų.

```

2 references
14 public abstract class SampleDomainTests<TStartupModule> : GymziiDomainTestBase<TStartupModule>
15     where TStartupModule : IAbpModule
16 {
17     private readonly IIdentityUserRepository _identityUserRepository;
18     private readonly IdentityUserManager _identityUserManager;
19
20     0 references
21     protected SampleDomainTests()
22     {
23         _identityUserRepository = GetRequiredService<IIdentityUserRepository>();
24         _identityUserManager = GetRequired
25

```

Virtual member call in constructor

```

18 public class SampleRepositoryTests : GymziiEntityFrameworkCoreTestBase
19 {
20     private readonly IRepository<IdentityUser, Guid> _appUserRepository;
21
22     0 references
23     public SampleRepositoryTests()
24     {
25         _appUserRepository = GetRequiredService<IRepository<IdentityUser, Guid>>();
26     }

```

Simbolių deklaracijų pasikartojimai – šitoje skiltyje paprastai nurodoma, kad kode yra kažkas, ką būtų galima patobulinti, siekiant aiškumo kode. Tačiau tai neturi įtakos mūsų kodo funkcionalumui, tad palikome ankstesnį mūsų variantą.

```

5 namespace Gymzii.Permissions;
6
7 0 references
8 public class GymziiPermissionDefinitionProvider : PermissionDefinitionProvider
9 {
10     0 references
11     public override void Define(IPermissionDefinitionContext context)
12     {
13         var myGroup = context.AddGroup(name: GymziiPermissions.GroupName, displayName: L(name: "Permission:Gymzii"));
14         //Define your own permissions here. Example:
15         //myGroup.AddPermission(GymziiPermissions.MyPermission1, L("Permission:MyPermission1"));
16         var booksPermission = myGroup.AddPermission(name: GymziiPermissions.Exercises.Default, displayName: L(name: "Permission:Exercises"));
17         booksPermission.AddChild(name: GymziiPermissions.Exercises.Create, displayName: L(name: "Permission:Exercises.Create"));
18         booksPermission.AddChild(name: GymziiPermissions.Exercises.Edit, displayName: L(name: "Permission:Exercises.Edit"));

```

Rašybos klaidos – ReSharper įrankio žodyne nėra žodžio „Gymzii“, tad jis paryškina vien tik šituos žodžius, tačiau tai nėra jokia klaida ir nieko nekeisime, nes tai yra projekto pavadinimas.

```

15
16 namespace Gymzii;
17
18 [DependsOn(
19     params dependedTypes: typeof(AbpAuditLoggingDomainSharedModule),
20     typeof(AbpBackgroundJobsDomainSharedModule),
21     typeof(AbpFeatureManagementDomainSharedModule),
22     typeof(AbpIdentityDomainSharedModule),
23     typeof(AbpOpenIddictDomainSharedModule),
24     typeof(AbpPermissionManagementDomainSharedModule),
25     typeof(AbpSettingManagementDomainSharedModule),
26     typeof(AbpTenantManagementDomainSharedModule)
27 )]
28 4 references
29 public class GymziiDomainSharedModule : AbpModule
30 {

```

Sintaksės stilius – ReSharper duoda pasiūlymus koreguoti kodą taip, jog jis taptų kuo švaresnis, tvarkingesnis ir besivadovaujantis geriausiomis programuotojų praktikomis. Galimos stiliaus problemos, kurias pabrėžia statinės analizės įrankiai:

- Skliaustų stilius: ar skliaustai yra naujose eilutėse, ar toje pačioje eilutėje kaip ir metodo pavadinimas, if sakiniai ir pan.
- Įtrauka: ar įtraukos yra naudojamos visame kode.
- Tarpai: ar yra tarpų prieš operatorius, po kablelių ir pan.

Kadangi mūsų programos kodas buvo nuo pat pradžių rašomas, vadovaujantis tvarkingo kodo principais, jokių grubių nebuvo rasta ir taisyti nereikėjo.

Sukurta taisyklė

Šioje taisyklėje yra tikrinama ar kintamojo pradžia atitinka tam tikrą „policy“.

Search Pattern:

```
public $returnType$ $methodName$($parameters$)
{
    $anyCode$;
    $policyName$ = GymziiPermissions.Exercises.$expectedPolicyName$;
}
```

```
public ExerciseAppService(IRepository<Exercise, Guid> repository) :base(repository)
{
    _exerciseRepository = repository;
    GetPolicyName = GymziiPermissions.Exercises.Default;
    GetListPolicyName = GymziiPermissions.Exercises.Default;
    CreatePolicyName = GymziiPermissions.Exercises.Create;
    UpdatePolicyName = GymziiPermissions.Exercises.Edit;
    DeletePolicyName = GymziiPermissions.Exercises.Delete;
}
```

Išvados

Buvo atlikta projekto kodo analizė ir sukurta nauja taisyklė pasitelkiant JetBrains įrankį „ReSharper“, kuris puikiai veikia Visual Studio aplinkoje ir nereikalauja papildomų taisyklių kūrimo. Pasinaudoję šio įrankio pagalba, išsiaiškinome, kad projekte funkcinų klaidų nėra. Buvo rastos tik kelios smulkios klaidos, tokios kaip gramatinės ir panašios, kurios mūsų programos funkcionalumui įtakos neturi.