

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Jolita Gedminaitė

Kompiuterizuotos darbo vietos administravimo priemonių kūrimas ir tyrimas

Magistro darbas

Darbo vadovas:

prof. E. Bareiša

Kaunas, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Jolita Gedminaitė

Kompiuterizuotos vietos administravimo priemonių kūrimas ir tyrimas

Magistro darbas

Recenzentas:

Doc. E. Karčiauskas
2009-01-13

Vadovas:

prof. E. Bareiša
2009-01-13

Atliko:

IFM-3/4 gr. stud.
Jolita Gedminaitė
2009-01-13

Kaunas, 2009

Turinys

SUMMARY	4
SANTRAUKA	5
1. ĮVADAS	6
2. STATINĖS ANALIZĖS METODŲ APŽVALGA	8
2.1. STATINĖS ANALIZĖS METODO VIETA IR ROLĖ PROGRAMŲ KŪRIMO PROCESĖ	8
2.2. STATINĖ ANALIZĖ.....	8
2.3. STATINĖS ANALIZĖS METODAI.....	10
2.4. FxCOP ĮRANKIS IR JO ANALIZĖ	20
2.4.1. <i>FxCop naudojimas</i>	20
2.4.2. <i>Kodo introspekcijos modelis</i>	21
2.4.3. <i>Taisyklių rinkiniai</i>	23
2.4.4. <i>Taisyklės struktūra</i>	23
2.5. ANALIZĖS IŠVADOS	26
3. DISKSDB REIKALAVIMAI IR REALIZACIJA	27
3.1. NEFUNKCINIAI REIKALAVIMAI IR APRIBOJIMAI	27
3.2. DALYKINĖS SRITIES MODELIS.....	27
3.3. DISKSDB SISTEMOS FUNKCIJOS	28
3.4. DISKSDB SISTEMOS LOGINĖ ARCHITEKTŪRA	30
3.5. DUOMENŲ BAZĖS SCHEMA.....	32
3.6. DISKSDB REALIZACIJOS MODELIS	33
3.7. PAKETŲ PARAMETRAI.....	35
4. EKSPERIMENTINĖ DISKSDB SISTEMOS STATINĖ ANALIZĖ	37
4.1. KLaidų pranešimų paketuose vertinimas pagal taisyklių rinkinius	37
4.2. KLaidų pranešimų paketuose vertinimas pagal pranešimų lygį	39
4.3. KLaidų pranešimų taisyklių rinkiniuose vertinimas pagal pranešimų lygius	43
4.4. Pažeistų taisyklių rinkiniuose tyrimas	43
4.5. Pažeistų taisyklių rinkiniuose vertinimas pagal pažeidimų lygius	46
4.6. Pažeistų taisyklių paketuose vertinimas pagal taisyklių rinkinius	48
4.7. Pažeistų taisyklių tyrimas paketuose	53
4.8. Naujos taisyklės testavimas	55
5. IŠVADOS	56
6. LITERATŪRA	57
7. PRIEDAI	58
1 Priedas. DISKSDB pagrindinė klasių diagrama	58
2 Priedas. FxCOP sugeneruotos ataskaitos fragmentas.....	59

Summary

Computer Supported Workplace Administration Tools Development and Research

This master's degree work consists of three main parts: static analysis methods review, DisksDB system requirement specification and summary of results of static code analysis.

In the first part, introducing to principles of main static analysis methods, static analysis tool FxCop and writing of its rule .

In second part, introducing to analyzed system requirement specification. Also systems main source code data, that was used on static code analysis.

In third part, analysing static analysis tool FxCop output. For more clarity, they were analysed in several ways.

All part results were put in conclusions.

Santrauka

Šis magistrinis darbas susideda iš trijų pagrindinių dalių: statinės analizės metodų apžvalgos, DisksDB sistemos reikalavimų specifikacijos, ir gautų statinės kodo analizės rezultatų apibendrinimo.

Pirmojoje dalyje pateikiami pagrindiniai statinės analizės metodų veikimo principai, supažindinama su statinės analizės įrankiu FxCop bei taisyklių rašymo principais.

Antroje dalyje pateikiama analizuojamos sistemos pagrindiniai reikalavimai ir jų aprašymai. Pateikiami pagrindiniai realizuotos sistemos duomenys, kurie naudojami statinės analizės metu.

Trečioje dalyje analizuojami statinės analizės metu FxCop įrankio pateikti rezultatai. Jie, aiškumo dėlei, pateikiami įvairiais pjūviais.

Visų gautų rezultatų apibendrinimas pateikiamas išvadose.

1. Įvadas

Programinės įrangos kūrimas yra gana sudėtingas procesas kadangi jis susideda iš keleto etapų: reikalavimų analizės, projektavimo, kūrimo, testavimo, įdiegimo, vartotojų mokymais, palaikymo bei tobulinimo. Visi šie etapai yra labai svarbūs, todėl kiekvienas iš jų turi būti atliktas kokybiškai. Tačiau, kaip ir kiekvienoje srityje problemų išvengti yra sunku, todėl turi būti imtasi tam tikrų priemonių joms sumažinti.

Turbūt pagrindinis etapas, kurio metu galima susidurti su esminėmis ne tik šio bet ir ankstesnių etapų klaidomis, yra testavimas. Pagrindinės priežastys dėl kurių testavimo metu gali atsirasti klaidų yra šios:

- nesusipratimai, neapsižiūrėjimai, prasta komunikacija;
- programos sudėtingumas;
- programuotojo klaidos;
- besikeičiantys reikalavimai;
- riboti projektavimo terminai;
- dokumentacijos stoka;
- dokumentacijos kokybės;
- prasti PĮ kūrimo įrankiai;
- žinių ar išsilavinimo stoka;
- ir panašiai...

Dažniausiai pasitaikančios programavimo klaidos yra[10].

Testavimas gali būti atliekamas keletu būdų. Jis yra susijęs su produkto elgsenos sužadinimu ir būsenų stebėjimu. Todėl labai svarbu jį nustatyti pačioje PĮ kūrimo pradžioje. Tačiau yra ir kitų būdų užtikrinančių kokybę. Vienas iš jų – statinės kodo analizės metodas.

Statinė kodo analizė – tai PĮ analizė, kuri atliekama jos nepaleidus. Ji yra susijusi su statiniu sistemos atvaizdavimu, išryškinant potencialias problemas. Kad pasiekti efektyvią naudą testavimas ir statinė analizė turi būti naudojami kartu, jie yra vienas kitą papildantys procesai [5].

Šiame darbe buvo atliktas DisksDB programinės įrangos kūrimo procesas, kuris apėmė reikalavimų analizės, projektavimo, programavimo testavimo bei diegimo etapus. Visi šie etapai yra aprašyti darbe.

Šio darbo **tyrimo sritis** statinės analizės metodas ir jo taikymas. **Tyrimo objektas** statinio kodo metodo taikymas programinės įrangos programinio kodo klaidų paieškai.

Darbo tikslas ištirti statinės analizės metodą. Tam naudojamas FxCop įrankis, kuriame pateiktos statinės analizės taisyklės. Analizė buvo atliekama su jau sukurta programa. Ji buvo sukompiliuota ir pateikta analizei keleto .dll ir .exe failų rinkinių pavidalu. Atlikus statinę

analizę, buvo pateikti įvairūs pranešimai, kurie buvo vėliau buvo išanalizuoti. Šios analizės rezultatai taip pat pateikti šiame darbe.

Atsižvelgus į analizės rezultatus, buvo pakeistas DisksDB kodas. Daugiausia dėmesio buvo skirta kritiniams pranešimams.

Darbo uždaviniai:

1. Apžvelgti statinės analizės metodus;
2. atlikti statinės kodo metodikos analizę;
3. suprojektuoti pasirinktos programinės įrangos architektūrą;
4. atlikti pasirinktos programinės įrangos kokybės tyrimą, išbandant statinio kodo metodiką;
5. suprojektuoti ir išbandyti naują statinio kodo analizės įrankio taisyklę.

Problema

Pagrindinis dažnai pasitaikantis kuriamų sistemų trūkumas – kokybės stoka. Nedidelėms sistemoms ją užtikrinti nėra itin sunku, tačiau jei kalbama apie dideles sistemas, kokybės užtikrinimo klausimas yra itin aktualus ir svarbus, kadangi jos užtikrinimu reikia pradėti rūpintis ankstyvoje projektavimo fazėje.

Sistemos sutrikimų taisymas vėlyvoje kūrimo fazėje, įmonei gali atnešti didžiules išlaidas, todėl svarbu laiku nustatyti galimus sutrikimus. Kodo peržiūra yra vienas iš būdų užtikrinančių programinės įrangos kokybę. Tačiau žmogui tai atlikti gali būti sudėtinga, nes jis dažnai susiduria su tokiomis kliūtimis kaip, laiko apribojimai, ribota aptarnavimo apimtis. Kompiuteriai gali atlikti daugelį užduočių, su kuriom žmogus nesusitvarko.

Kita svarbi problema, kuri dažniausiai pasitaiko didelėse kompanijose - tai parašyto kodo vientisumas. Tai reiškia, kad programuotojams rašant programinį kodą, turi būti laikomasi bendrų taisyklių, kurios nustatytos kompanijos viduje.

Sukurta sistema „DisksDB“ šių problemų tai pat neišvengė. Tačiau ši sistema nėra didelė, o jos klaidų taisymas nesukeltų didelių finansinių nuostolių, tačiau tai pareikalautų papildomų laiko sąnaudų.

2. Statinės analizės metodų apžvalga

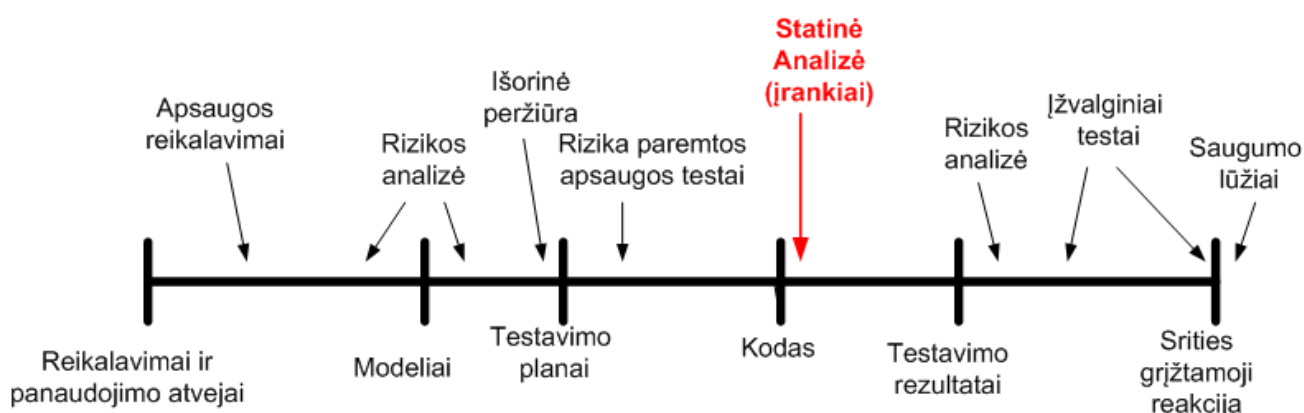
Analizės tikslas išanalizuoti projektavimo proceso ir statinės analizės metodus. Apžvelgti statinės analizės procesą.

2.1. Statinės analizės metodo vieta ir rolė programų kūrimo procese

Pagrindinis dažnai pasitaikantis kuriamų sistemų trūkumas – kokybės stoka. Nedidelėms sistemoms ją užtikrinti nėra itin sunku, tačiau jei kalbama apie dideles sistemas, kokybės užtikrinimo klausimas yra itin aktualus ir svarbus, kadangi jos užtikrinimu reikia pradėti rūpintis ankstyvoje projektavimo fazėje.

Sistemos sutrikimų taisymas vėlyvoje kūrimo fazėje, įmonei gali atnešti didžiules išlaidas, todėl svarbu laiku nustatyti galimus sutrikimus. Vienas iš geriausių būdų, nustatyti sistemos sutrikimus – statinės analizės metodo taikymas.

Žemiau pateiktame paveiksle pateiktas programinės įrangos kūrimo ciklas, kuriame parodyta, kurioje vietoje turi būti vykdoma statinė analizė [5][6].



1 pav. Programinės įrangos kūrimo ciklas.

Kita svarbi problema, kurią gali išspręsti statinės analizės metodas, dažniausiai pasitaiko didelėse kompanijose - tai parašyto kodo vientisumas. Tai reiškia, kad programuotojams rašant programinį kodą, turi būti laikomasi bendrų taisyklių, kurios nustatytos kompanijos viduje.

2.2. Statinė analizė

Bet koks įrankis, kuris analizuoja kodą, jo nevykdant, atlieka statinę analizę. Kad aptiktų saugumo problemas statinės analizės įrankiai turi veikti panašiai kaip ir rašybos tikrintojai. Jie ieško įvairių, gerai suprantamų klaidų. Tačiau klaidų neaptinkamumas dar nereiškia, kad klaidų nėra, tai tik garantuoja, kad programiniame kode nėra nustatytų konkrečių dažniausiai pasitaikančių problemų.

Daugumai patyrusių programuotojų statinė analizė yra naudinga. Tačiau ji nepadaro jų dar didesniais profesionalais. Kai kuriais atvejais statinė analizė yra naudojama žymiai plačiau, negu daugelis žmonių įsivaizduoja.

Statinė analizė yra tinkama identifikuojant apsaugos problemas dėl keleto priežasčių:

- statinė analizė vykdoma nuodugniai, nuolatos ir be nukrypimų. Taip programuotojas yra informuojamas apie iškilusias problemas, ar kurios kodo dalys yra lengvai vykdomos dinaminio testavimu;
- analizuojant patį kodą, gali būti parodomos esminės klaidos priežastys, o ne kuris nors iš jos simptomų. Tai yra ypač svarbu norint įsitikinti, kad pažeidimai yra pataisyti tinkamai;
- statinė analizė gali rasti klaidas ankstyvoje programinės įrangos vystymo fazėje, prieš paleidžiant programą pirmą kartą. Anksti suradus klaidas yra ne tik sumažinami klaidos taisymo kaštai, bet ir greito grįžtamojo ryšio ciklas gali padėti programuotojo darbui: programuotojas turi galimybę ištaisyti klaidas, kurių ankščiau nesitikėjo atsirasiant;
- atsiradus naujai atakų atmainai, pasitelkiant statinės analizės metodą, galima lengvai peržiūrėti didelę teksto dalį, ir pamatyti, kur nauja ataka galėtų įvykti. Kai kurie apsaugos defektai egzistuoja programinėje įrangoje metų metus, kol jie yra surandami. Todėl galimybė apžvelgti patikimą kodą dėl naujai išaiškintų trūkumų tampa neįkainojama.

Klaidų programų kode, statinės analizės įrankiai ieško pagal fiksuotą struktūrą ar taisyklių rinkinį. Labiau pažengusius įrankius laikui bėgant galima papildyti naujomis taisyklėmis, tačiau jeigu taisyklė iki tol nėra parašyta ieškant konkrečios problemos, įrankis niekada neras problemas.

Statinės analizės įrankių išvestis vis dar reikalauja žmogaus įvertinimo. Įrankiui nėra galimybių automatiškai suprasti, kuri problema yra daugiau ar mažiau svarbi vartotojui, todėl nėra galimybių išvengti paieškos išvedamuose pranešimuose ir nuspręsti kurie iš jų turi būti pataisyti, o kurie turi priimtina rizikos lygį [5].

Statinės analizės etapai:

- *Valdymo srauto analizė.* Tikrina ciklus su daug išėjimo ir įėjimo taškų, randa nepasiekiamą kodą ir t.t.
- *Duomenų naudojimo analizė.* Randa neaprašytus kintamuosius, kintamuosius aprašytus dukart be tarpinio priskyrimo, kintamuosius, kurie aprašyti bet niekada nepanaudoti ir t.t.

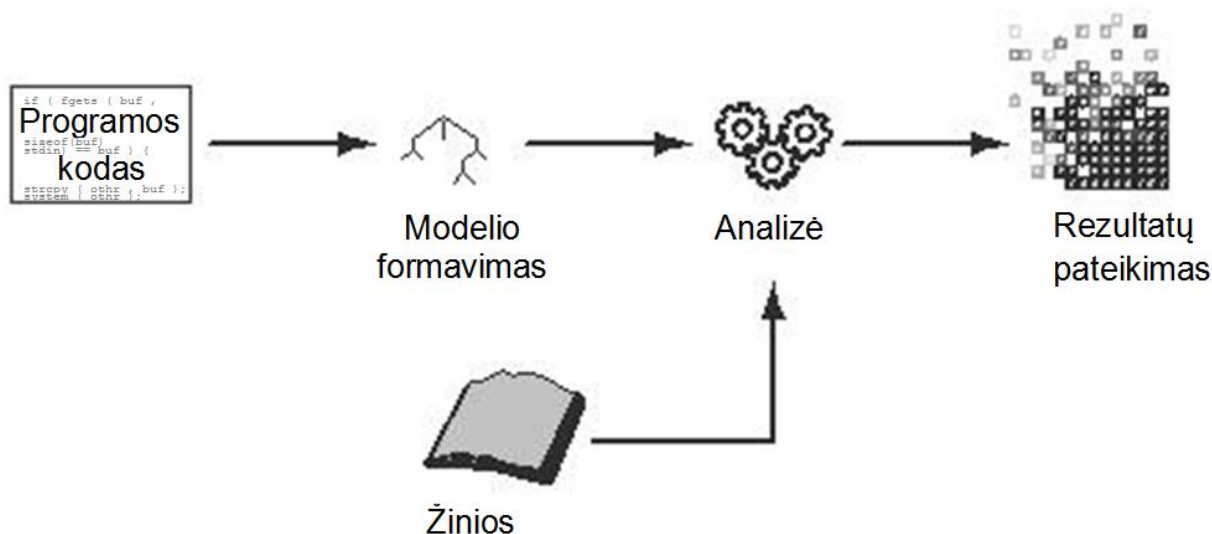
- *Sąsajos analizė*. Tikrina paprogramių ir procedūrų aprašymų neprieštaringumą bei jų panaudojimą
- *Informacijos srauto analizė*. Nustato išeities (angl. *output*) reikšmių priklausomybes. Neieško anomalijų, bet išryškina informaciją *kodo* peržiūrai ar analizei
- *Kelių analizė*. Parenka programos vykdymo kelią ir nurodo operatorius, vykdomus šiuo keliu. Be to, labai naudinga apžvalgos metu [7].



2 pav. Statinės analizės metodo etapai.

2.3. Statinės analizės metodai

Statinės analizės metodai analizės įrankiuose vykdomi daugiaž vienodai (žr. 3 pav.). jie visi priima kodą, suformuoja modelį, kuris jį atvaizduoja, jį analizuoja naudojantis turimomis žiniomis ir baigia pateikdamas rezultatus vartotojui.



3 pav. Bendri statinės analizės įrankio procesai

Modelio formavimas

Statinės analizės įrankis analizuojamą programos kodą paverčia į programos modelį – duomenų struktūrų rinkinį, kuris atspindi kodą. Toks įrankio sukurtas modelis glaudžiai siejasi su analizės rūšimi, kurią jis atlieka. Bendrai paėmus, statinės analizės įrankiai daug ką perima iš kompiliatorių, kadangi daug statinės analizės metodų buvo sukurta žmonių, dirbančių su kompiliatoriais ir kompiliavimo optimizavimo problemomis.

Modelio formavimo metu yra naudojami skirtingi statinės analizės metodai. Toliau aprašyti svarbiausi statinės analizės įrankių metodai ir duomenų struktūros.

1) Leksikos analizė

Įrankiai, kurie operuoja programiniu kodu, darbą pradeda transformuojant kodą į simbolių eilutes, pašalinant nesvarbias programos teksto savybes, tokias kaip tarpai ar komentarai. Toks simbolių srauto sukūrimas vadinamas leksikos analize. Leksikos analizės taisyklės dažnai naudoja įprastas išraiškas, kad būtų galima identifikuoti simbolius. Pateiktame pavyzdyje (žr.

1 pavyzdys.) pavaizduotos paprastos leksikos analizės taisyklės, kurios gali būti naudojamos žemiau nurodyto C kalbos programos fragmento apdorojimui.

C kalbos programos fragmentas:

```
if (ret) // probably true
mat[x][y] = END_VAL;
```

šis kodas pateikia tokias simbolių sekas:

```
IF LPAREN ID(ret) RPAREN ID(mat) LBRACKET ID(x) RBRACKET LBRACKET
ID(y) RBRACKET EQUAL ID(END_VAL) SEMI
```

1 pavyzdys. Paprastos leksikos analizės taisyklės.

```
if          { return IF; }
(          { return LPAREN; }
)          { return RPAREN; }
[          { return LBRACKET; }
]          { return LBRACKET; }
=          { return EQUAL; }
;          { return SEMI; }
/[ \t\n]+/ { /* ignore whitespace */ }
/\/\/*.*/  { /* ignore comments */ }
/[a-zA-Z][a-zA-Z0-9]*/ { return ID; }
```

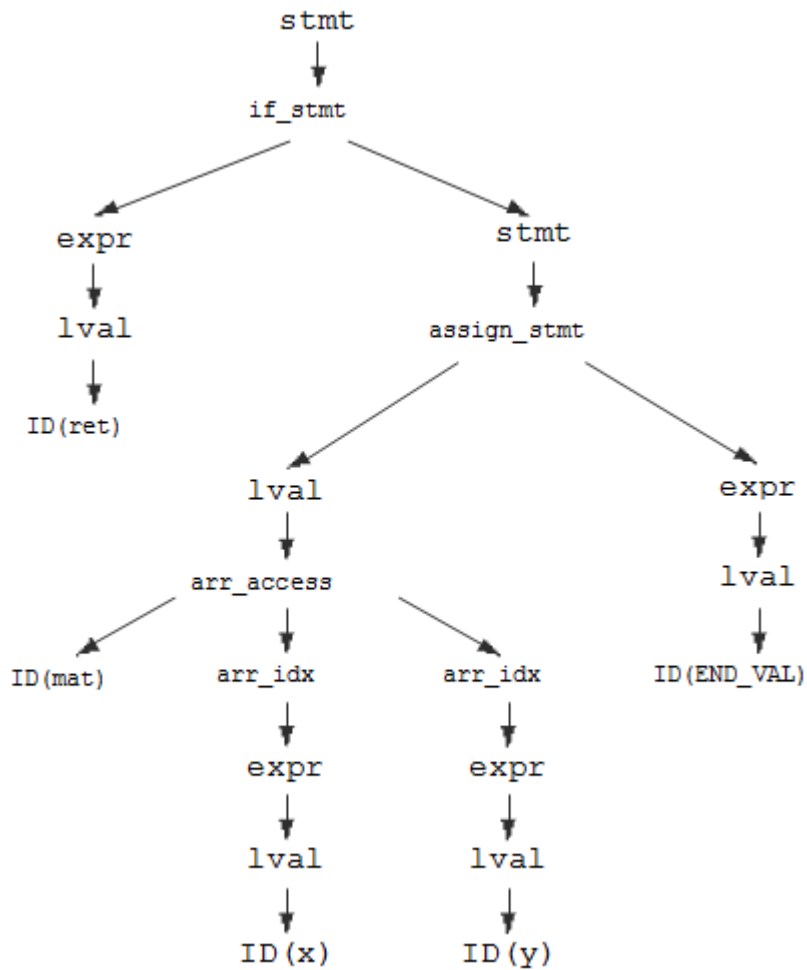
2) Gramatinė analizė (angl. *Parsing*)

Kalbos analizatorius simbolių srauto palyginimui naudoja nuo konteksto nepriklausomą gramatiką (angl. *context-free grammar (CFG)*). Gramatiką sudaro rezultatų rinkiniai, kurie apibūdina kalboje esančius simbolius (elementus). Kalbos analizatorius atlieka kilmės nustatymą palygindamas simbolių srautą su rezultatų rinkiniais. Jeigu kiekvienas simbolis yra sujungtas su simboliu iš kurio jis kilęs, tuomet formuojamas analizės medis. Pateiktame pavyzdyje (žr. 2 pavyzdys.) išvardintas rezultatų rinkinys, kuris gali atlikti paprasto simbolių srauto gramatinę analizę.

2 pavyzdys. Rezultatų taisyklės paprasto simbolių srauto analizei.

```
stmt := if_stmt | assign_stmt
if_stmt := IF LPAREN expr RPAREN stmt
expr := lval
assign_stmt := lval EQUAL expr SEMI
lval = ID | arr_access
arr_access := ID arr_index+
arr_idx := LBRACKET expr RBRACKET
```

Paveiksle pavaizduotas gramatinės analizės medis (žr. 4 pav.), sukurtas naudojantis rezultatų taisyklėmis (žr. 2 pavyzdys.).

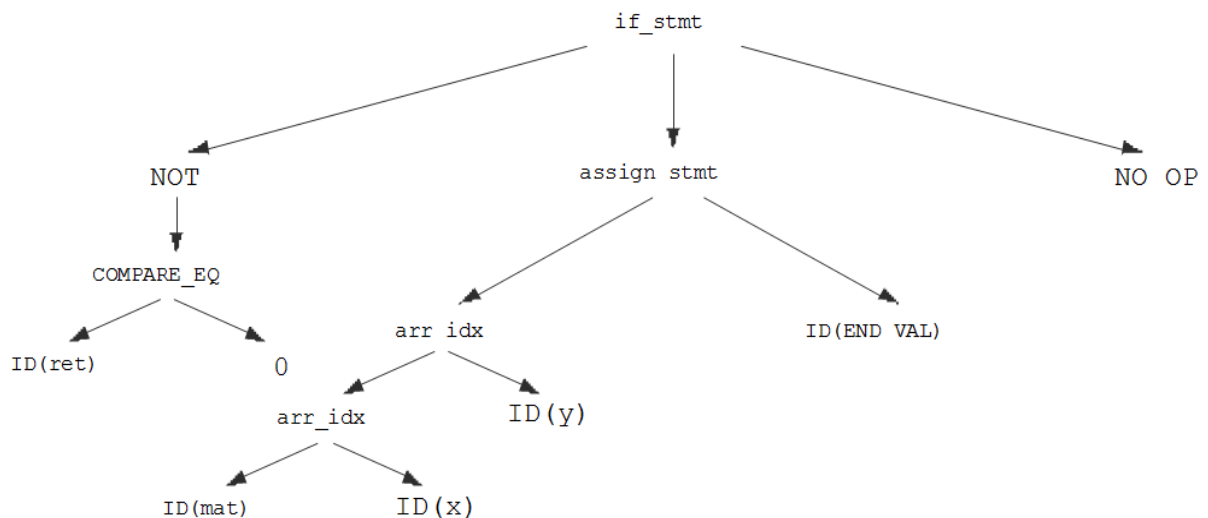


4 pav. Gramatinės analizės medis, išvestas iš simbolių sekos

3) Abstrakti sintaksė

Tam tikri stilistiniai tikrinimai geriausiai vykdomi analizės medžiu, kadangi jis tiksliausiai atspindi programinį kodą. Tačiau atliekant sudėtingą analizę, analizės medis gali būti nepatogus dėl daugelio priežasčių. Medžio taškai išvedami iš gramatinių rezultato taisyklių, kurios gali pateikti negalutinius simbolius. Tie simboliai egzistuoja tik dėl to, kad gramatinė analizė būtų lengva ir nedviprasmiška, o ne dėl to kad būtų sudarytas lengvai suprantamas medis. Todėl jį dažniausia geriau išgauti tiek iš gramatikos detalių, tiek iš sintaksinio cukraus pateikto programos tekste.

Duomenų struktūra, kuri atlieka šiuos dalykus vadinama abstrakčios sintaksės medis (angl. *abstract syntax tree (AST)*). AST tikslas – pateikti standartizuotą programos versiją, tinkamą vėlesnei analizei. AST paprastai sudaromas susiejant medžio konstrukciją su gramatikos rezultato taisyklėmis.



5 pav. Abstrakčios sintaksės medis (AST)

4) Semantinė analizė

Formuojant AST, tuo pat metu yra formuojama simbolių lentelė. Kiekvienam programos identifikatoriui, simbolių lentelė priskiria tam tikro tipo identifikatorių su rodykle į jo aprašymą ar paskelbimą.

Turėdamas AST ir simbolių lentelę, statinės analizės įrankis atlieka tipo tikrinimą. Tipo tikrinimo informacija yra labai svarbi objektiškai orientuotos kalbos analizei, kadangi objekto tipas nustato metodų rinkinį, kurį objektas gali taikyti.

Kompiliatoriuose simbolių suskaidymas ir tipų tikrinimas yra vadinamas semantine analize (angl. *semantic analysis*) kadangi kompiliatorius priskiria reikšmes programoje esantiems simboliams. Statinės analizės įrankiai, kurie naudoja šias duomenų struktūras turi žymų pranašumą lyginant su tais, kurie nenaudoja. Pavyzdžiui, jie gali teisingai suprasti perkrautų operatorių reikšmes C++ kalboje arba nustatyti, kad Java metodas *doPost()* yra *HttpServlet* realizacijos dalis. Šios galimybės leidžia įrankiui atlikti naudingus programos struktūros tikrinimus. Tokiems tikrinimams ypač naudojamas terminas struktūrinė analizė (angl. *structural analysis*).

Atlikus semantinę analizę, kompiliatorių ir labiau pažengusių statinės analizės įrankių keliai išsiskiria. Naujesni kompiliatoriai naudoja AST bei simbolių ir tipų informaciją tarpinio atvaizdavimo (intermediate representation) generavimui. Tai yra bendra mašininio kodo versija, tinkama optimizavimui ir konvertavimui į platformai būdingą objektinį kodą. Statinės analizės kelias nėra tokie aiškūs, priklausomai nuo atliekamos analizės tipo, statinės analizės įrankis AST medyje gali atlikti papildomas transformacijas arba gali sugeneruoti savo tarpinių atvaizdavimų įvairovę, tenkinantį jo poreikius.

Jei statinės analizės įrankis naudoja savo tarpinius atvaizdavimus, paprastai atsižvelgia į priskyrimus, išsišakojimus, ciklus ir funkcijos kvietimus (juos galima traktuoti įvairiais būdais).

Tarpinis atvaizdavimas kurį naudoja statinės analizės įrankis, dažniausia yra aukštesnio lygio programos atvaizdavimas, nei to, kurį naudoja kompiliatorius. Pavyzdžiui, C kalbos kompiliatorius greičiausia savo tarpiniam atvaizdavimui konvertuos visus struktūrinių laukų ryšius į baitų perėjimus struktūroje, o statinės analizės įrankis tęs struktūros laukų susiejimą pagal vardus.

5) Kontrolinio srauto sekimas

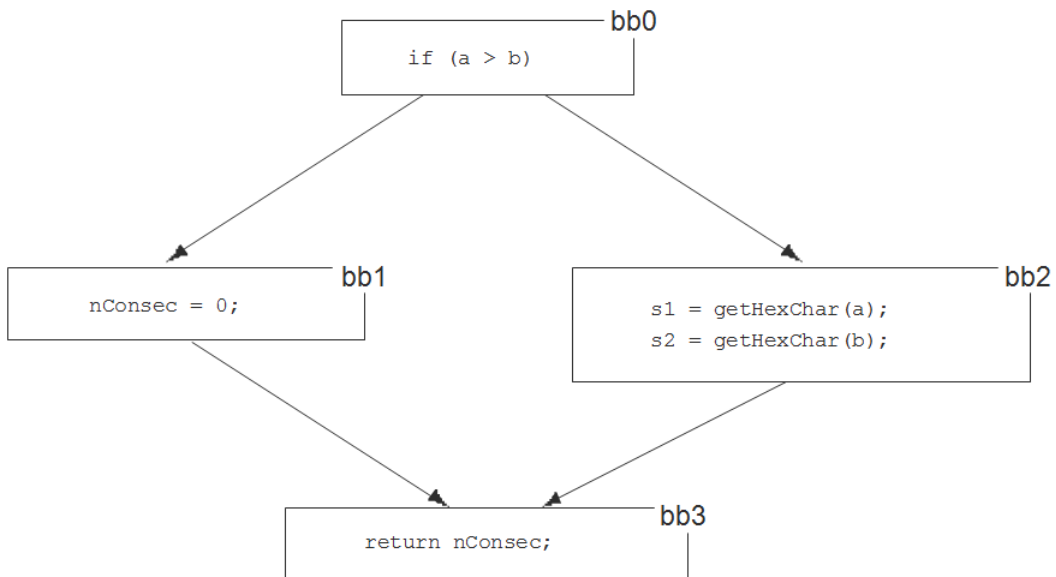
Daugelis statinės analizės algoritmų peržiūri skirtingus vykdymo atvejus, kurie gali įvykti paleidus funkciją. Kad tuos algoritmus padaryti efektyviais, dauguma įrankių vietoj AST arba tarpinio atvaizdavimo suformuoja kontrolinio srauto grafą (control flow graph). Kontrolinio srauto grafo mazguose yra baziniai blokai (basic blocks): komandų sekos, kurios visada bus įvykdytos pradedant pirma ir baigiant paskutine, be galimybės praleisti nors vieną. Kontrolinio srauto grafe yra nurodoma ribos, o atitinkamas kontrolinis sruvas pereina tarp blokų. Galinės ribos atitinka galimus ciklus.

3 pavyzdys. C kalbos programos fragmentas sudarantis keturis pagrindinius blokus.

```
if (a > b) {  
  nConsec = 0;  
} else {  
  s1 = getHexChar(1);  
  s2 = getHexChar(2);  
}  
return nConsec;
```

Nurodyto programos fragmento (žr. 3 pavyzdys.) kontrolinio srauto grafe (žr. 6 pav.) keturi baziniai blokai sužymėti nuo bb0 iki bb3. Kiekvieno iš šių blokų komandos yra pateiktos programinio kodo forma, tačiau bazinė blokinė struktūra statinės analizės įrankyje dažniausia turi nuorodas į AST mazgus arba įrankio tarpinio atvaizdavimo mazgus.

Kai paleidžiama programa, jos kontrolinis sruvas gali būti apibrėžtas bazinių blokų rinkiniu, kuriuos ji vykdo. Pėdsakas (trace) tai bazinių blokų seka, kuri charakterizuoja perėjimą per kodą. Aukščiau pateiktam kodui (žr. 3 pavyzdys.) egzistuoja du galimi veikimo keliai. Tie du keliai paženklinėti dviem unikaliais pėdsakais [bb0, bb1, bb3] ir [bb0, bb2, bb3] kontrolinio srauto grafe (žr. 6 pav.)



6 pav. Kontrolinio srauto grafas su keturiais baziniais blokais

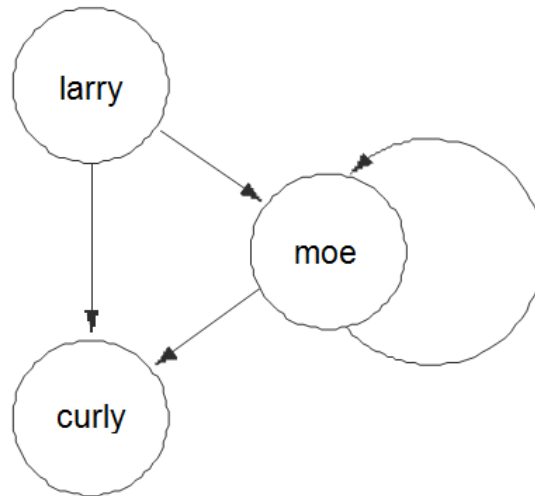
Kreipinių grafas (call graph) vaizduoja galimus kontrolinius srautus tarp funkcijų ir metodų. Nesant funkcijos nuorodų ar virtualių metodų, kreipinių grafo konstravimas yra funkcijos identifikatorių, nurodytų funkcijoje, analizė. Grafo mazgai atitinka funkcijas, o rodyklės galimus funkcijų kreipinius.

Programos fragmentas su trimis funkcijomis (žr. 4 pavyzdys.) ir šios programos kreipinių grafas (žr. 7 pav.) patekti žemiau.

4 pavyzdys. Programos fragmentas su trimis funkcijomis.

```

int larry(int fish) {
if (fish) {
moe(1);
} else {
curly();
}
}
int moe(int scissors) {
if (scissors) {
curly();
moe(0);
} else {
curly();
}
}
int curly() {
/* empty */
}
  
```



7 pav. Programos fragmento (žr. 4 pavyzdys.) kreipinių grafas

6) Duomenų srauto sekimas

Duomenų srauto analizės algoritmai programoje analizuoja duomenų judėjimą. Kompiliatoriai atlieka duomenų srauto analizę kad paskirstyti registrus, pašalinti neveikiantį kodą, ir atlikti daug kitų optimizavimų. Duomenų srauto analizė paprastai apima funkcijos kontrolinio srauto grafo kirtimą ir stebėjimą kur yra generuojamos duomenų reikšmės ir kur jos yra naudojamos. Funkcijos konvertavimas į vienetinę statinio priskyrimo (Static Single Assignment (SSA)) formą išsprendžia daug duomenų srauto problemų. SSA formoje funkcijos kintamajam reikšmę leidžiama priskirti tik vieną kartą. Pritaikant šį apribojimą, nauji kintamieji turi būti pateikiami programoje. Kompiliatoriaus literatūroje, nauji kintamieji dažniausia yra pateikti papildant tikrąjį kintamojo vardą apatiniu skaitiniu indeksu, kad kintamajam x priskyrus reikšmę tris kartus, perrašyta programa juos perduos kintamiesiems x_1 , x_2 , ir x_3 .

SSA forma yra naudinga, nes lengva nustatyti kiekvieno kintamojo, pateikto programoje, reikšmės kilmę. Šią savybę turi daugelis taikomųjų programų. Pavyzdžiui, jei SSA kintamajam kada nors yra priskiriama pastovi reikšmė, šia reikšme SSA kintamasis gali būti pakeistas visuose panaudojimuose. Šis metodas vadinamas pastovios reikšmės platinimu (constant propagation). Jis yra naudingas ieškant apsaugos spragų, tokių kaip įkoduotų (hard-coded) slaptažodžių ar šifravimo raktų.

5 pavyzdys. TEA šifravimo algoritmo fragmentas, įprasta programinio kodo forma

```

sum = sum + delta ;
sum = sum & top;
y = y + (z<<4)+k[0] ^ z+sum ^ (z>>5)+k[1];
y = y & top;
z = z + (y<<4)+k[2] ^ y+sum ^ (y>>5)+k[3];
z = z & top;

```

6 pavyzdys. TEA šifravimo algoritmo fragmentas, vienetinė statinio priskyrimo forma (SSA)

```
sum2 = sum1 + delta1 ;  
sum3 = sum2 & top1;  
Y2 = Y1 + (z1<<4)+k[0]1 ^ z1+sum3 ^ (z1>>5)+k[1]1;  
Y3 = Y2 & top1;  
Z2 = z1 + (y3<<4)+k[2]1 ^ y3+sum3 ^ (y3>>5)+k[3]1;  
z3 = z2 & top1;
```

7) Žymės skleidimas

Apsaugos įrankiams būtina žinoti kurias reikšmes programoje įsilaužėlis gali turėti galimybę kontroliuoti. Duomenų srautų naudojimas siekiant nustatyti ką įsilaužėlis gali kontroliuoti, vadinamas žymės skleidimu (taint propagation). Jis reikalauja duomenų pro kur informacija patenka į programą ir kaip ji joje keliauja. Žymės skleidimas yra vienas iš būdų identifikuoti daug tvirtinimo ir atvaizdavimo trūkumų. Pavyzdžiui, programa, kurioje yra buferio perpildymo pažeidimas, beveik visada turi duomenų srauto kelią iš įvesties funkcijos į pažeidžiamą operaciją.

8) Rodyklės pavadinimo keitimas (Pointer Aliasing)

Rodyklės pavadinimo keitimo analizė – tai dar viena duomenų srauto problema. Pavadinimo pakeitimo analizės tikslas yra suprasti kurios rodyklės gali rodyti į tą pačią atminties vietą. Pavadinimo keitimo algoritmai apibrėžia rodyklės ryšius tokiais terminais kaip „privalo keisti pavadinimą“, „gali keisti pavadinimą“, „negali keisti pavadinimą“. Daug kompiliatoriaus optimizavimų reikalauja tam tikros pavadinimo keitimo analizės. Pavyzdžiui, kompiliatorius

Pavyzdžiui, kompiliatorius galėtų iš pertvarkyti kitus du sakinius, jei rodyklės p1 ir p2 nerodo į pačią atminties vietą:

```
*p1 = 1;  
*p2 = 2;
```

Apsaugos įrankiams, pavadinimo keitimo analizė yra svarbi atliekant žymės skleidimą. Srautui jautrus žymės sekimo algoritmas turi atlikti pavadinimo keitimo analizę, kad išsiaiškintų, kad duomenys perduodami iš *getUserInput()* į *processInput()*:

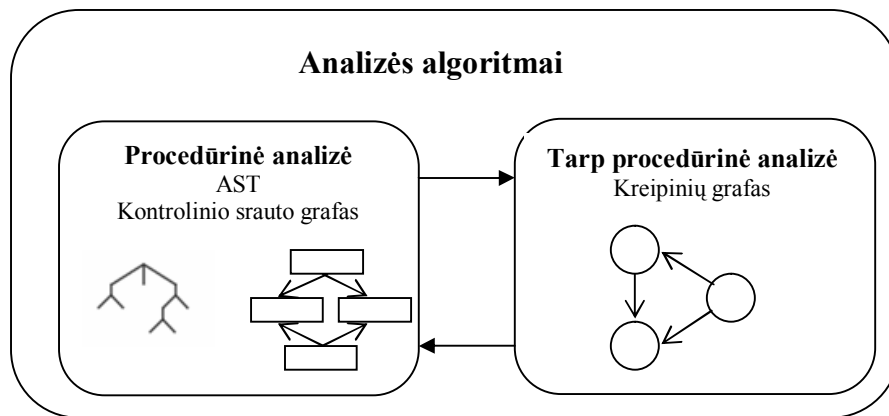
```
p1 = p2;  
*p1 = getUserInput ();  
processInput (*p2);
```

Statinės analizės įrankiai daro prielaidą, kad rodyklės, bent jau tos, kurios perduodamos kaip funkcijos argumentai, nekeičia pavadinimo. Ši prielaida daroma daugelyje įrankių, kad būtų pateikti naudingi rezultatai, tačiau tai gali įtakoti svarbių rezultatų nepastebimumą.

Analizės algoritmai

Bet kokia labiau pažengusi analizės strategija susideda bent jau iš dviejų pagrindinių dalių: procedūrinės analizės komponento, kuris vykdo individualios funkcijos analizę ir tarp

procedūrinės analizės komponento, kuris skirta funkcijų sąveikos analizei. Šie du komponentai turi tarpusavio ryšius, kurie sieja mažesnes duomenų struktūras. (žr. 8 pav.)



8 pav. Analizės algoritmų komponentai

Statinės analizės įrankiai

Statinės analizės metu yra naudojama daug skirtingų statinės analizės įrankių rūšių. Statinės analizės įrankių rūšys [5]:

- Tipų tikrinimo (angl. *Type checking*)
- Stiliaus tikrinimo (angl. *Style checking*)
- Programos supratimo (angl. *Program understanding*)
- Programos kontrolės (angl. *Program verification*)
- Savybių tikrinimo (angl. *Property checking*)
- Klaidų ieškojimo (angl. *Bug finding*)
- Apsaugos apžvalga (angl. *Security review*)

2.4. FxCop įrankis ir jo analizė

2.4.1. FxCop naudojimas

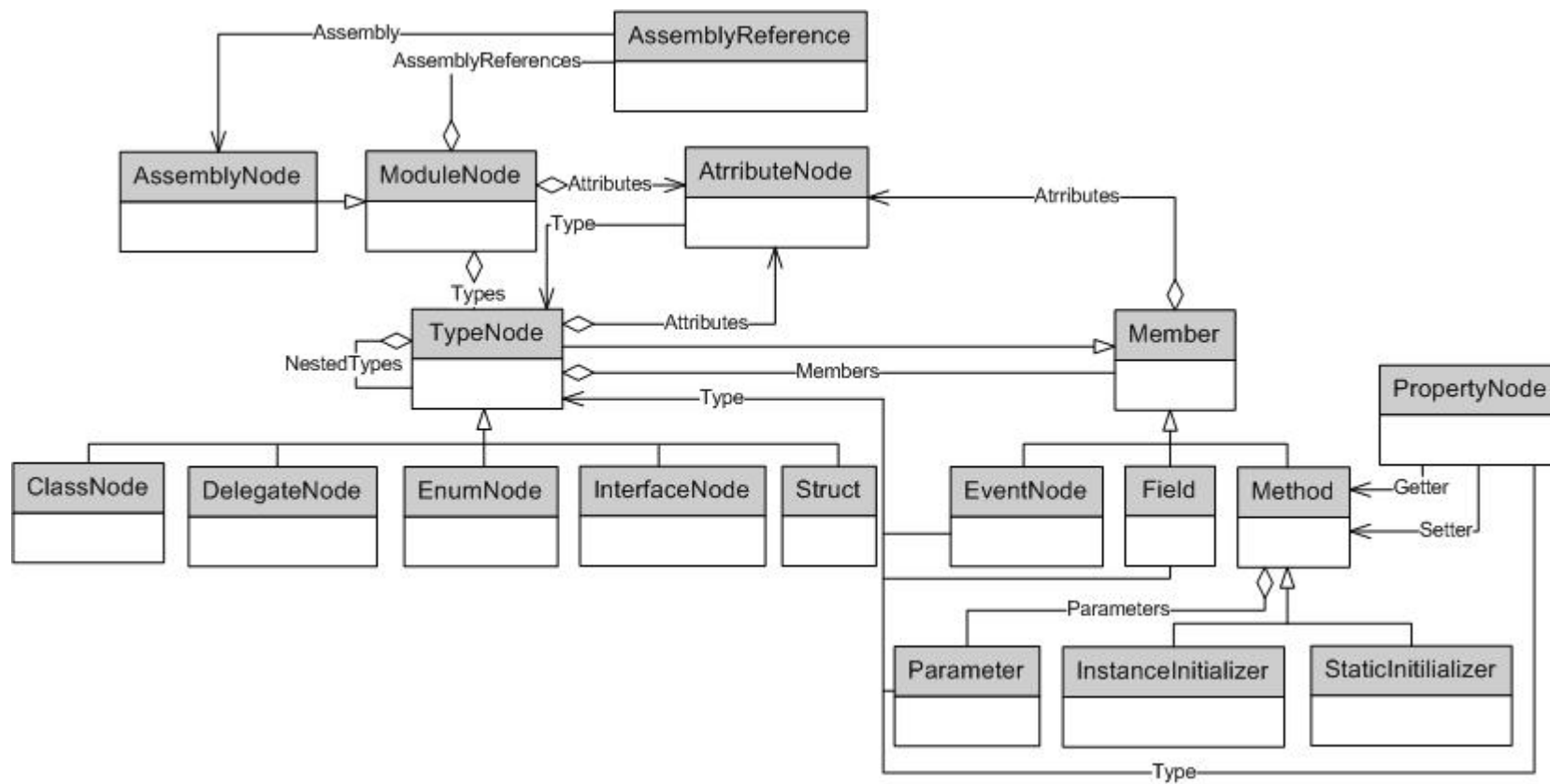
Priešingai nei senesni kodo analizės įrankiai (pvz. Lint, kuris skirtas C kalba parašyto kodo analizei), FxCop neanalizuoja source kodo. Jis analizuoja dvejetainę bendrą tarpinę kalbą (angl. *Common Intermediate Language (CIL)*) sugeneruotą .NET kompiliatorių ir pateiktą .NET paketuose (EXE ir DLL failuose). Analizė tampa įmanoma dėl metaduomenų esančių CIL gausos. Tiesiogiai analizuojant paketus, FxCop išvengia prisirišimo prie konkrečios .NET programavimo kalbos: jis analizuoja programas, parašytas C#, VB.NET, C++/CLI ar kitomis galimomis .NET kalbomis, jų nemodifikuojant. Todėl prieš analizuojant pasirinktą programą prieš tai ją reikia sukompiliuoti.

Sukompiliuota programa paketais įkeliami į FxCop projektą. Tų paketų yra tiek, kiek Visual Studio solution turi projektų. Po to pasirenkamos taisyklės, kurios bus taikomos atliekant analizę ar kiti analizės parametrai, ir tik tada galima pradėti statinę analizę. FxCop patikrins

pateiktus paketus pagal pasirinktas taisykles ir pateiks rezultatų suvestinę, apie aptiktus pažeidimus. Taip pat parametruose galima nurodyti, kad šis įrankis netikrintų automatiškai sugeneruoto programinio kodo.

2.4.2. Kodo introspekcijos modelis

FxCop stainės kodo analizės įrankis kiekvienam paketui sukuria introspekcijos modelį kurį vėliau analizuoja (žr. 9 pav.)



9 pav. FxCop įrankio objektų modelis

2.4.3. Taisyklių rinkiniai

FxCop turi keletą taisyklių rinkinių, paremtų .NET Framework kūrimo nurodymais, kurie yra jau įkelti sukūrus naują FxCop projektą. Žemiau yra pateikiami taisyklių rinkinių aprašymai.

Dizaino taisyklių rinkinys (angl. *Design ruleset*). Susitelkiama ties sąsaja su vartotoju ir programinio kodo struktūra. Šios grupės taisyklės užtikrina tinkamą klasių, įvykių, rinkinių, vardų sričių ir parametrų realizavimą. Šio rinkinio taisyklių pažeidimai paprastai neturi įtakos kodo veikimui.

Globalizacijos taisyklių rinkinys (angl. *Globalization ruleset*). Aptinkami trūkstami arba neteisingi informacijos, susijusios su globalizacija ir lokalizacija, panaudojimai.

Funkcinio suderinamumo taisyklių rinkinys (angl. *Interoperability*). Skirtingų kompiuterių ir sistemų savybė, užtikrinanti, kad jie galėtų veikti drauge, keistis duomenimis ir jais naudotis. Pasiekama laikantis standartų, taip pat naudojant suderinančias priemones.

Mobilumo taisyklių rinkinys (angl. *Mobility*) kodo paieška, kuris efektyviai neveiks mobilioje ar nesurištoje aplinkoje.

Pavadinimų taisyklių rinkinys (angl. *Naming*) vykdo vardų standartų, aprašytų projektavimo nurodymuose, pažeidimų paiešką. Tikrinama, ar elementų (pvz.: klasių, atributų ar kintamųjų) pavadinimai atitinka nustatytus standartus. Kai kurios taisyklės gali aptikti netgi rašybos klaidas priskirtuose varduose.

Našumo taisyklių rinkinys (angl. *Performance*) aptinkamos vietos, kuriose kodas galėtų būti optimizuotas našumui užtikrinti. Ieškoma įvairaus nesusijusio ar perteklinio programos kodo.

Perkeliamumo taisyklių rinkinys (angl. *Portability*) vykdoma kodo, kuris gali sunkiai būti perkeliamas tarp funkcionuojančių aplinkų, paieška.

Apsaugos taisyklių rinkinys (angl. *Security*) padeda nustatyti nepakankamus ar neteisingus apsaugos taikymus. Ieškoma trūkstamų atributų, netinkamų teisių panaudojimų ir galimybių įvykti SQL įterpinių atakoms.

Panaudojimo taisyklių rinkinys (angl. *Usage*) aptinkami galimi trūkumai paketuose, kurie gali įtakoti sistemos veikimą

2.4.4. Taisyklės struktūra

Taisyklės metaduomenys – XML

Kiekvienas taisyklių rinkinys turi XML resursus aprašančius metaduomenis, kiekvienai taisyklei.

```

<?xml version="1.0" encoding="utf-8"?>
<Rules FriendlyName="Naming Rules">
  <Rule TypeName="DoNotNameEnumValuesReserved" Category="Microsoft.Naming"
  CheckId="CA1700">
    <Name>Do not name enum values 'Reserved'</Name>
    <Description>Avoid naming an enum value 'Reserved'. Adding 'Reserved'
    values will lead to a breaking change on removal when the value is actually
    used in a later version of the product.</Description>
    <Url>@/Naming/DoNotNameEnumValuesReserved.html</Url>
    <Resolution>If '{0}' is not used in the current implementation, remove
    it. Otherwise give it a meaningful name.</Resolution>
    <Email>
    </Email>
    <MessageLevel Certainty="95">Error</MessageLevel>
    <FixCategories>Breaking</FixCategories>
    <Owner />
  </Rule>
  <Rule TypeName="ResourceStringCompoundWordsShouldBeCasedCorrectly"
  Category="Microsoft.Naming" CheckId="CA1701">
    <Name>Resource string compound words should be cased correctly</Name>
    <Description>Avoid creating compound words from terms which exist in the
    dictionary as discrete terms. Do not create a compound word such as
    'StopWatch' or 'PopUp'. These terms are recognized in the dictionary and
    should be cased as 'Stopwatch' and 'Popup'.</Description>
    <Url>@/Naming/ResourceStringCompoundWordsShouldBeCasedCorrectly.html</Url>
    <Resolution Name="ShouldBeDiscreteTerm">The compound word '{0}' in
    resource '{2}' exists as a discrete term. If your usage is intended to be
    single word, case it as '{3}'.</Resolution>
    <Resolution Name="ShouldBeCompoundWord">The discrete term '{0}' in
    resource '{2}' should be expressed as a compound term. If '{0}' refers back
    to an API component, case it as '{3}', otherwise split it into two words
    separated by a space.</Resolution>
    <Localized Name="ErrorReadingXmlResource">Error reading Xml resource:
    {0}</Localized>
    <Email>
    </Email>
    <MessageLevel Certainty="80">Error</MessageLevel>
    <FixCategories>NonBreaking</FixCategories>
    <Owner />
  </Rule>
</Rules>

```

Gairių aprašymai

Rule gali būti naudojama tiek kartų, kiek reikia. Nė viena iš šių gairių neturėtų būti suprantama kai nebūtina. Įvykyje, kuriame reikšmė nereikalaujama, gairė vis tiek turėtų būti, tačiau jos turinys gali likti tuščias [8].

Gairė, atributas	Gairės, atributo aprašymas
FriendlyName	Taisyklių rinkinio vardas, kuris parodomas FxCop GUI taisyklių medyje. FriendlyName yra Rules atributas.
TypeName	.NET klasės vardas, kuri realizuoja taisyklę. Nereikia maišyti su vardų sritimi. TypeName yra Rule atributas.
Category	Grupė, kuriai priklauso taisyklė. Tai programinė FriendlyName versija. Viename rinkinyje jis dažniausia yra vienodas visoms taisyklėms.

	Category yra svarbus galutiniam vartotojui, kai jis įtraukia išimtis į taisyklę Category yra Rule atributas.
CheckId	Identifikatorius, atpažįstantis taisyklę tarp kitų, tame pačiame taisyklių rinkinyje. Dažniausia jį sudaro dvi raidės ir keturi skaitmenys. kaip ir Category, jis yra svarbus, kai vartotojas paskelbia taisyklei išimtis (exceptions). CheckId yra Rule atributas.
Name	Taisyklės vardas, koks jis bus atvaizduojamas FxCop GUI taisyklių medyje.
Description	Taisyklės aprašymas. Dažniausia tai yra detalus paaiškinimas kodėl taisyklė egzistuoja ir kaip vartotojas turėtų ištaisyti pažeidimą. FxCop šį aprašymą pateikia apatiniame GUI lange, kai pažymimas pažeistos taisyklės pranešimas.
Url	Su taisykle susijęs URL, kuris gali pateikti daugiau informacijos apie taisyklę negu paprastai įdedama prie Description. Url gali būti paliktas tuščias.
Resolution	Trumpas pasiūlymas kaip pataisyti pažeidimą. Jis gali turėti String.Format rezervuotas vietas (placeholders), kurių reikšmės programiškai pateikia taisyklė, kai ji aptinka pažeidimą. FxCop parodo sprendimą vartotojui, kai taisyklė yra pažeidžiama. Daug sprendimų vienai taisyklei gali būti aprašyti jei Name atributas yra įtraukiamas į Resolution gairę. Taisyklė parinks vieną iš šių sprendimų kai ji aptiks pažeidimą.
MessageLevel	Klaidos lygis. Galimos reikšmės: CriticalError, Error, CriticalWarning, or Warning.
Certainty	Tai sveikasis skaičius nuo 0 iki 100, kuris parodo kaip dažnai taisyklė aptinka tikrą problemą programos kode. Ši gairė yra tik informacinė.
Email	Elektroninio pašto adresas susijęs su taisykle skirtas parodyti vartotojui. Dažnai paliekamas tuščias.
FixCategories	Charakterizuoja pažeidimo taisymo poveikį. Galimos reikšmės: Breaking, NonBreaking, or DependsOnFix. Ši gairė yra tik informacinė.
Owner	Asmuo, kuriam „priklauso“ taisyklė. Skirta parodyti vartotojui. Dažnai paliekama tuščia.

Taisyklėms yra priskirtas vienas iš penkių svarbumo lygių:

Kritinė klaida (angl. *Critical error*). Trūkumai, kurie gali būti matomi, arba kurie neleidžia kodui tinkamai atlikti įprastus scenarijus. Į kritinių klaidų pranešimus turėtų būti reaguojama pirmiausiai.

Klaida (angl. *Error*). Šie trūkumai turi mažesnės įtakos kodo veikimui negu kritinės klaidos, tačiau jie neturėtų būti praleisti jų atidžiai neįvertinus

Kritinis perspėjimas (angl. *Critical Warning*). Trūkumai kurie dažniausia neturi arba turi mažai neigiamos įtakos kodo elgsenai. Jie pirmiausia svarstomi norint užtikrinti kodo palaikymą

ir taisant neoptimalius sprendimus matomuose elementuose. Tačiau nedaugelis tokių pranešimų yra svarstybinos klaidos, todėl jie turėtų būti atidžiai peržvelgti.

Perspėjimas (angl. *Warning*). Trūkumai, kurie dažniausia apsvarstomi norint užtikrinti bazinio kodo stabilumą, išplečiamumą ir palaikomumą.

Informacija (angl. *Informational*). Pranešimai, kuriuos pateikia taisyklės kurios pateikia informaciją apie objektą, o ne aptinka klaidas.[9]

2.5. Analizės išvados

Šioje dalyje buvo pasirinktas kodo analizės įrankis geriausia atitinkantis kuriamos sistemos reikalavimus, kadangi sistema DisksDB bus realizuota C# kalba, kurią palaiko .NET. kadangi šią kalbą analizuoja statinės analizės įrankis FxCop, jis ir buvo pasirinktas. Kita svarbi priežastis buvo kad šis įrankis pateikiamas nemokamai ir jį galima integruoti į sistemos kūrimo aplinką Visual Studio.

3. DisksDB reikalavimai ir realizacija

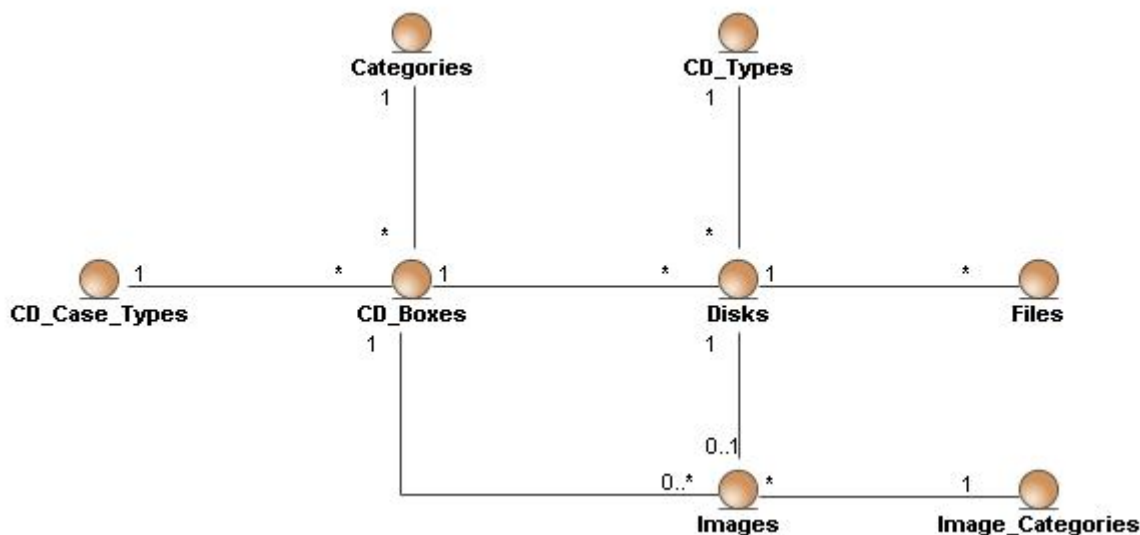
3.1. Nefunkciniai reikalavimai ir apribojimai

Pagrindiniai sistemos reikalavimai yra šie:

- Nepriklausomumas nuo platformos – kuriama sistema turi būti kuo mažiau priklausoma nuo konkrečios platformos. Kliento dalis turi būti nepriklausoma nuo operacinės sistemos: personalinio kompiuterio programa turi veikti Windows XP, 2003, 2000, Linux ir BSD aplinkose, o delninio kompiuterio programa – Windows CE 4.2 ir aukštesnėse aplinkose
- Išplečiamumas – pirminė projekto versija turi būti suprojektuota lanksčiai kadangi ateityje gali būti planuojama šį projektą plėsti.
- Patogus naudojimas – sistema turi būti draugiška vartotojui ir ją diegiančiam bei palaikančiam administratoriui.
- Lengvas integravimas – įrankis turi lengvai integruotis su organizacijoje naudojamomis ir naujai diegiamomis sistemomis.
- Sistema turi būti sukurta .NET palaikančia kalba.

3.2. Dalykinės srities modelis

Kompiuterizuojama dalykinė sritis suskirstoma į keletą objektų apjungtų ryšiais (žr. 10 pav.).



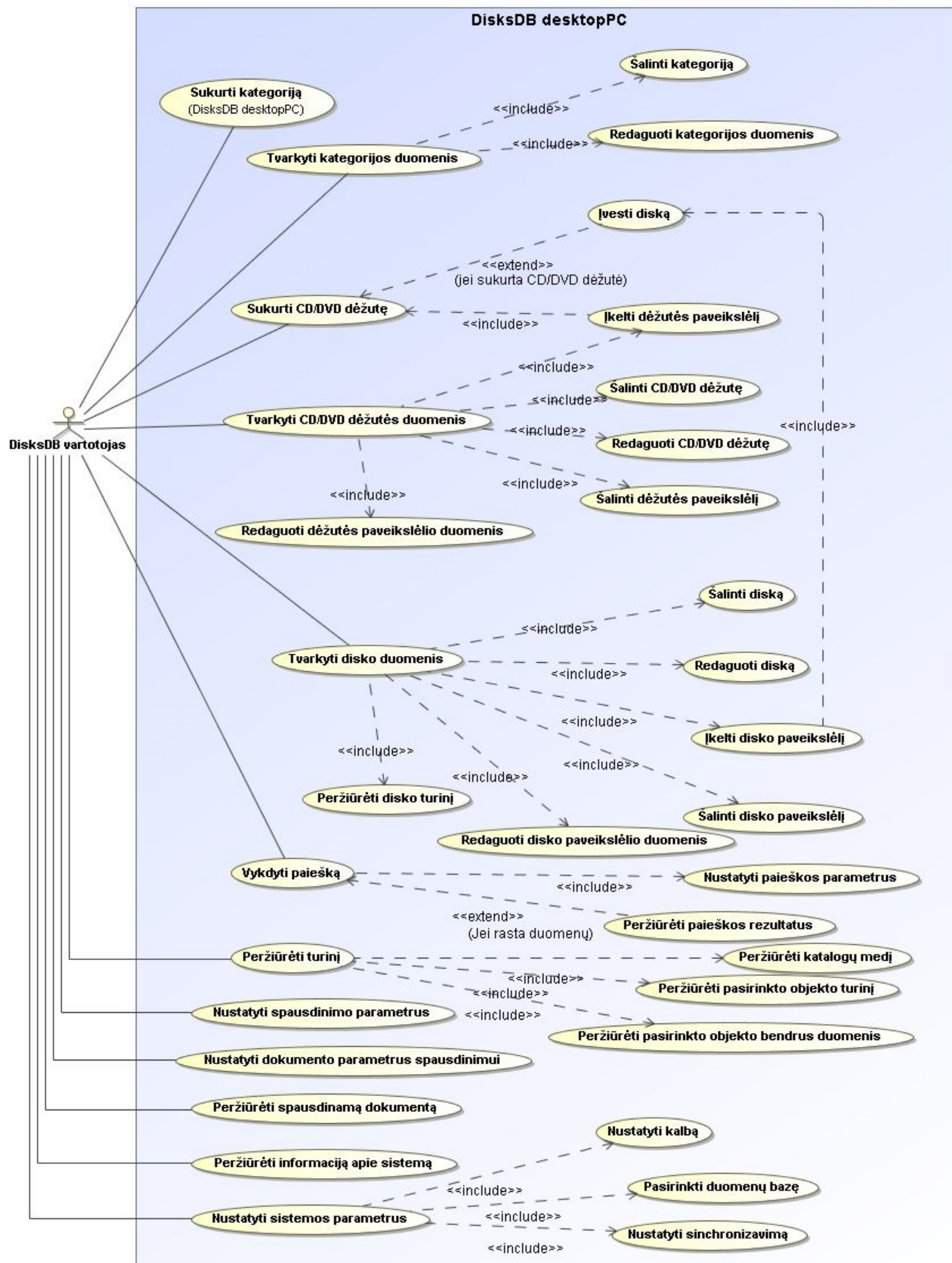
10 pav. Diskų tvarkymų sistemos objektų diagrama

Šie objektai išreiškia pagrindinę kompiuterizuojamą struktūrą, kuri bus naudojama duomenims saugoti.

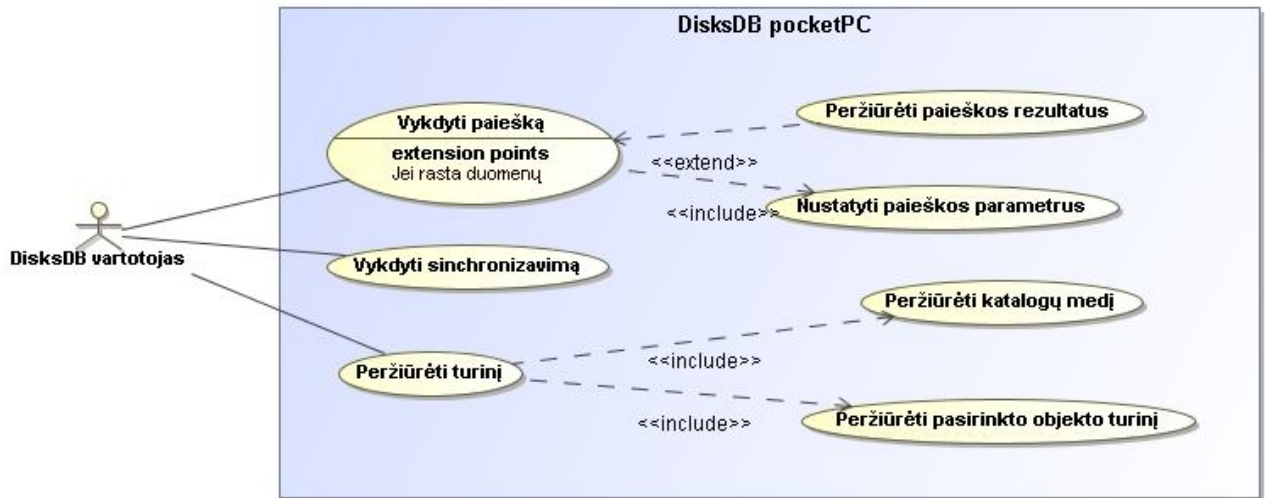
3.3. DisksDB sistemos funkcijos

DisksDB sistema bus skirta dviem aplinkom: personalinio kompiuterio aplinkai ir delininio kompiuterio aplinkai. Jos bus atskirtos ir galės veikti nepriklausomai viena nuo kitos, tačiau tarpusavyje jos galės bendrauti t.y. sinchronizuoti duomenis. Žemiau pateiktos šių aplinkų panaudojimo atvejų diagramos (žr. *11 pav.* ir *12 pav.*).

Šiose schemose DisksDB vartotojas yra asmuo turintis kompaktinių ar DVD diskų organizavimo problemą. Sistema turi padėti ją išspręsti, realizuotomis funkcijomis.



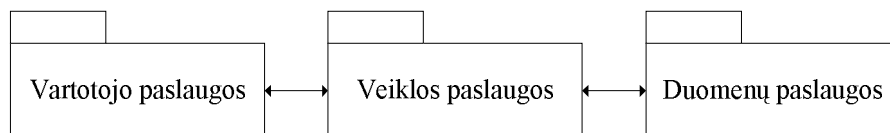
11 pav. DisksDB desktopPC sistemos panaudojimų atvejų diagrama



12 pav. DisksDB pocketPC sistemos panaudojimų atvejų diagrama

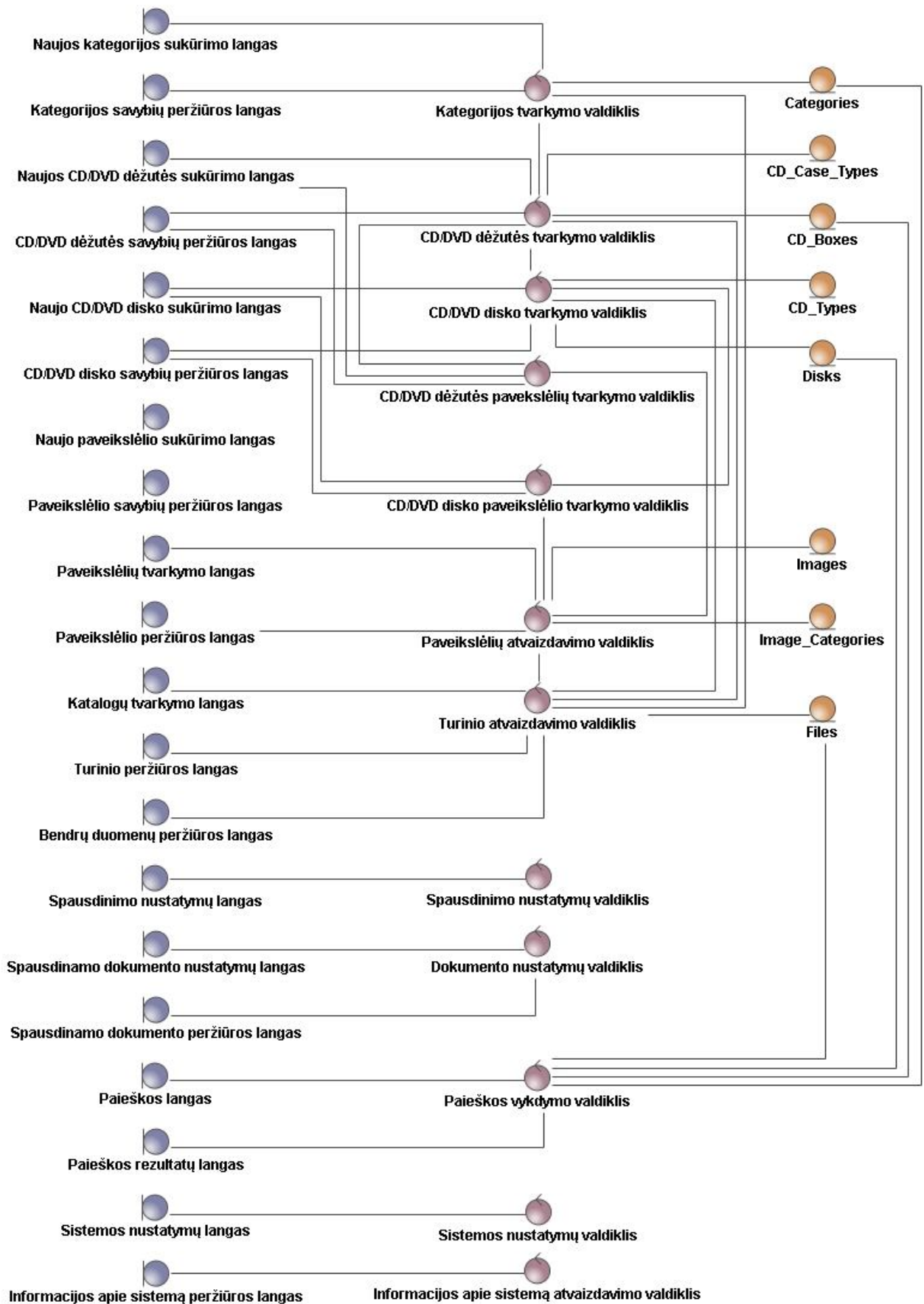
3.4. DisksDB sistemos loginė architektūra

Pagrindinės kuriamos sistemos loginės dalys: vartotojo paslaugos, veiklos paslaugos ir duomenų paslaugos (žr. 13 pav.). Vartotojo paslaugos apibūdina sistemos grafinę aplinką, kurioje vartotojas į sistemą įveda duomenis bei gauna rezultatus. Veiklos paslaugos atvaizduoja sistemos vidinę struktūrą, kuri parodo duomenų perėjimus tarp metodų ar klasių. Duomenų paslaugos parodo kaip duomenys yra saugomi, iš jos sistema paima duomenis kai vartotojas to pareikalauja.

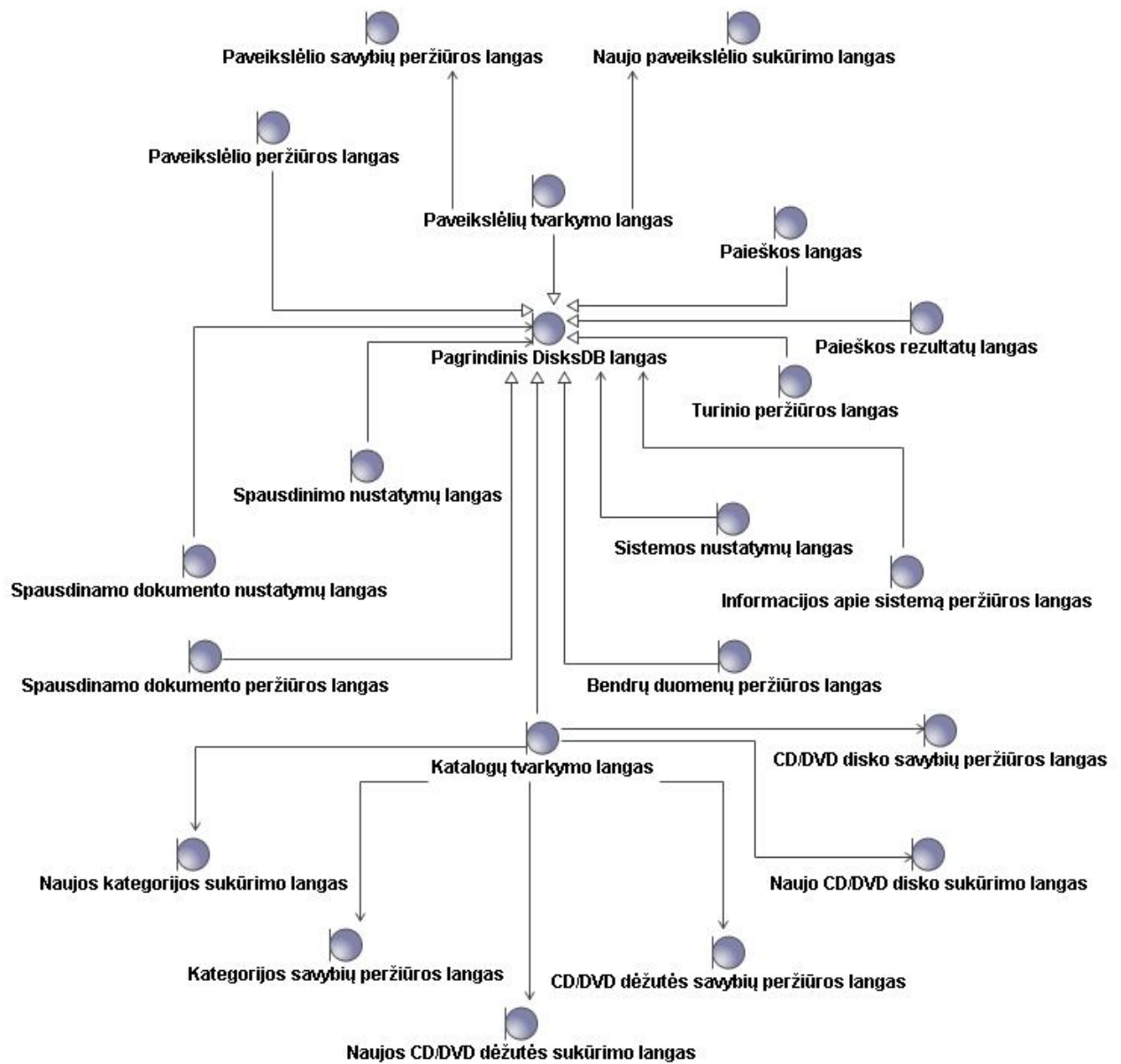


13 pav. Trijų lygių loginės architektūros principinė schema

Detalioj DisksDB (žr. 14 pav.) sistemos informacijoj pateikiami ryšiai tarp šių trijų loginės architektūros dalių. Jie ir parodo pagrindinius informacinius srautus.



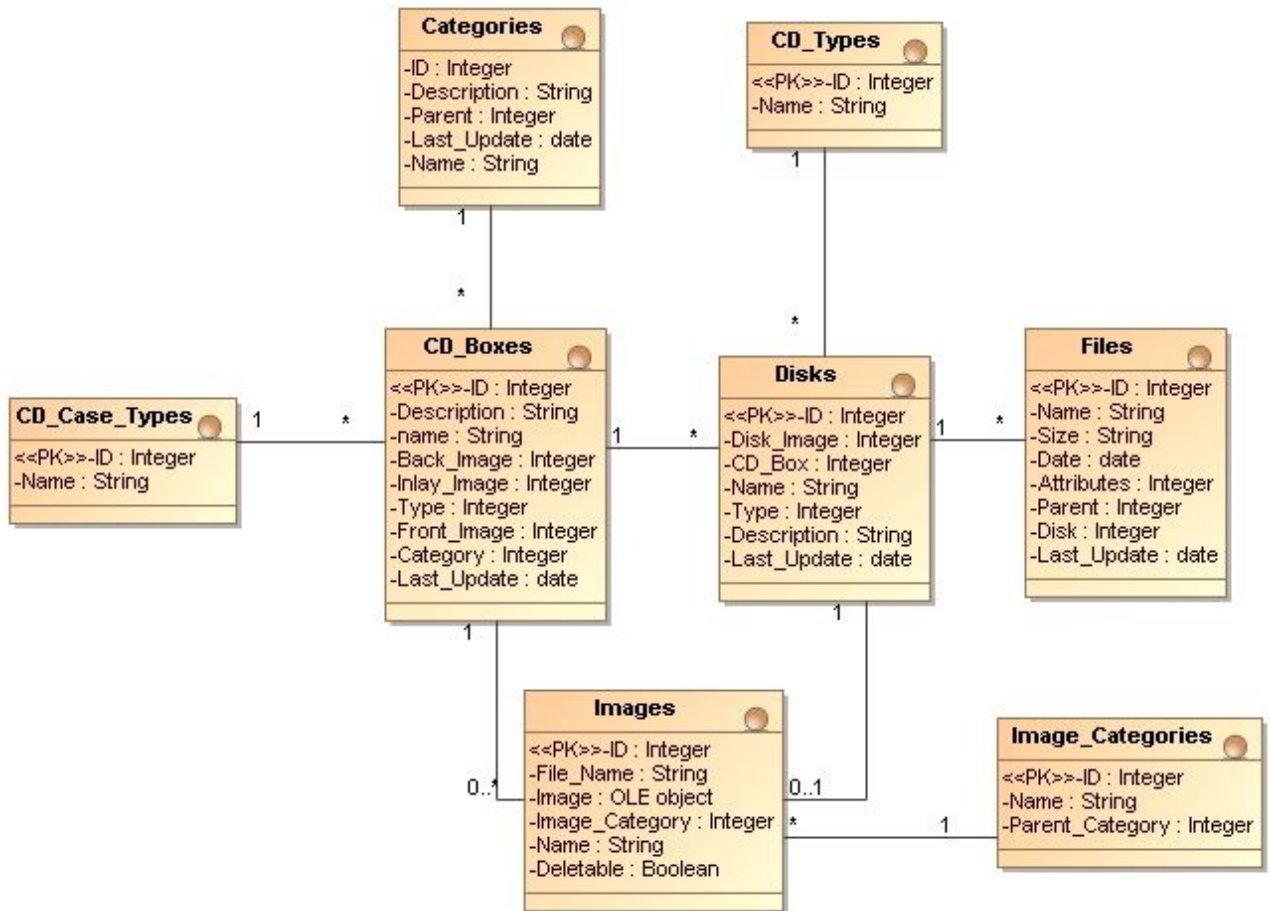
14 pav. Detali DisksDB sistemos objektų diagrama



15 pav. Vartotojo paslaugų objektų tarpusavio ryšiai

3.5. Duomenų bazės schema

Sistemos duomenų bazę sudaro dalykinės srities objektai sujungti tarpusavio ryšiais. Joje yra saugomi vartotojo įvesti duomenys, kurie vėliau gali būti panaudojami. Naudojant įvairias užklausas galima tuos duomenis

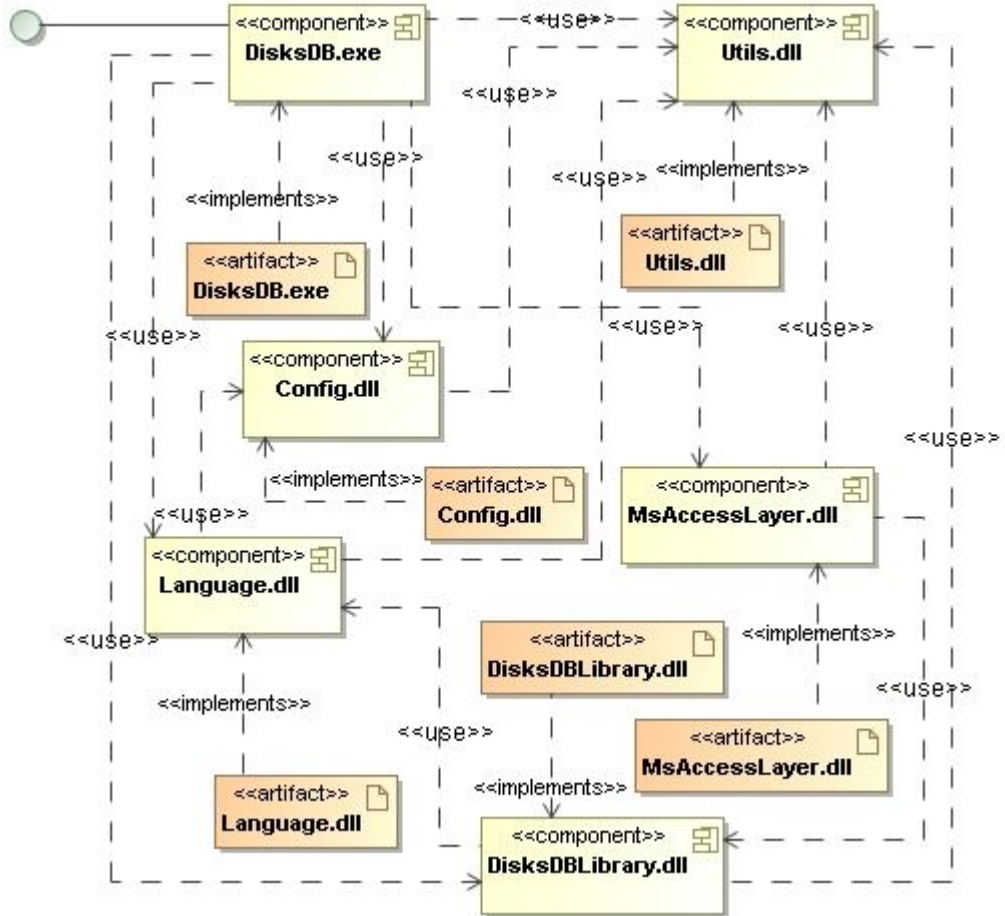


16 pav. Duomenų klasių diagrama

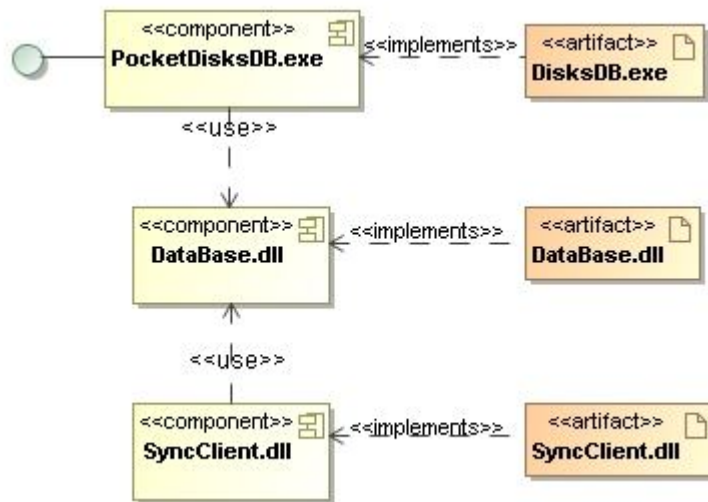
3.6. DisksDB realizacijos modelis

Sistemos komponentų galima pamatyti kaip susiję sistemoje esantys projektai tarpusavyje taip pat galima pamatyti kokiais artefaktais jie bus realizuoti.

Žemiau yra pateikti DisksDB sistemos komponentų diagramos atskirai pocketPC ir atskirai desktoPC aplinkoms (žr. 17 pav. ir 18 pav.).

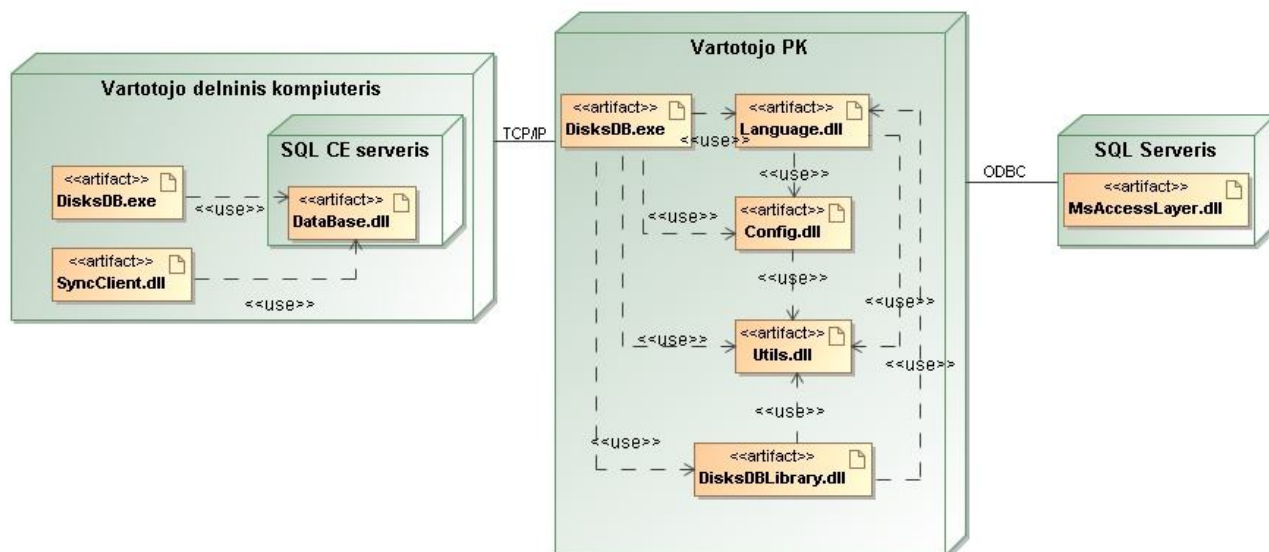


17 pav. DesktopPC platformos komponentų diagrama



18 pav. PocketPC platformos komponentų diagrama

Sistemos diegimo diagramoje (žr. 19 pav.) pateiktas bendras sukurtos sistemos modulių diegimas atskirose aplinkose bei jų tarpusavio ryšiai. Jų viduje parodyta kokie artefaktai sudaro sistemą bei atskurus jos modulius.



19 pav. DisksDB sistemos diegimo diagrama

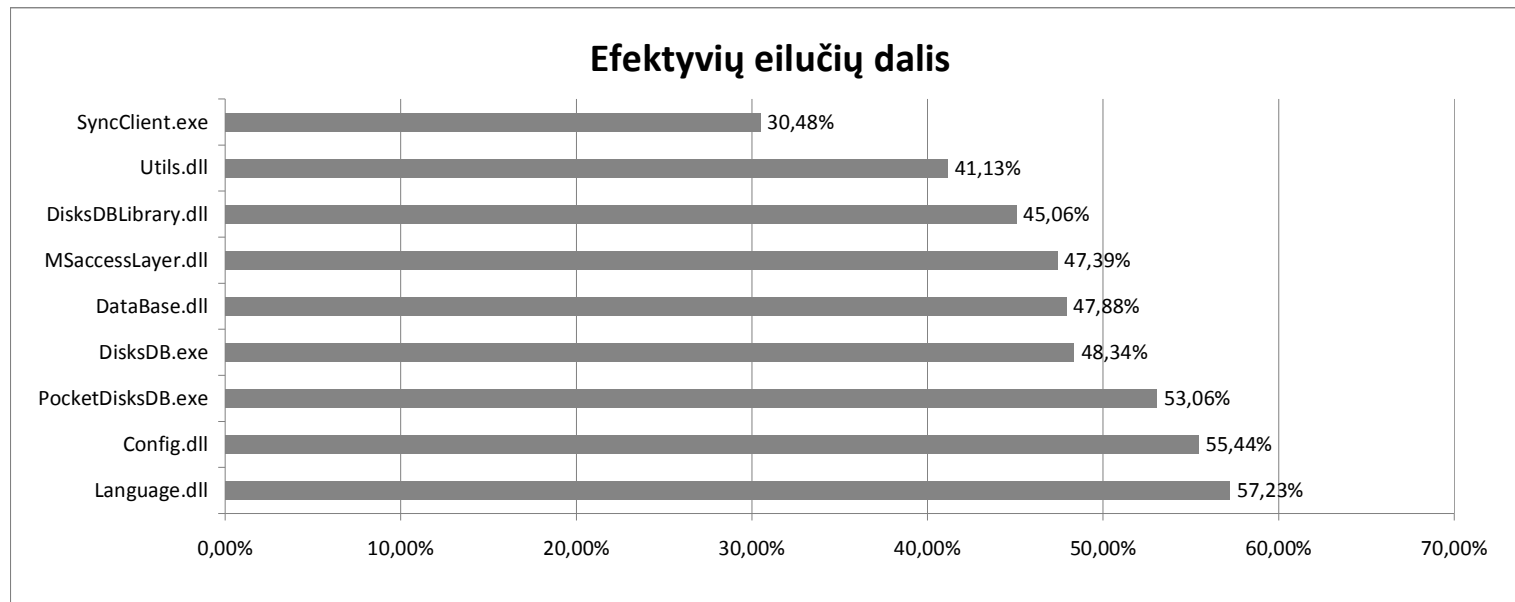
3.7. Paketų parametrai

Sukurtą sistemą sudaro du moduliai PocketPC ir DesktopPC. Šiuos du modelius sudaro paketai, kurie buvo gauti kompiliuojant programinį kodą. Sistemos kūrimo įrankio pagalba, buvo gauti pagrindiniai šių paketų duomenys (žr. 1 lentelė.). šioje lentelėje pateikiami kiekvieno paketo failų, klasių vardų sričių ir pan. Kiekiai. Taip pat pateikiamas kodo eilučių ir efektyvių eilučių kiekiai, bei efektyvių eilučių dalis.

Efektyvios eilutės tai tos eilutės, kurios nėra tuščios, ar kuriose nėra vien tik metodų skliaustų ar komentarų.

1 lentelė. Sistemos paketų pagrindinė informacija

Platforma	Projektas	Failų skaičius	Automatinis kodas	Automatinio kodo dalis	Vardų sritys	.NET tipai					Eilučių kiekis	Efektyvių eilučių kiekis	Efektyvių eilučių dalis
						Klasių	Struktūrų	Sąrašų	Interfeisų	Delegatų			
DesktopPC	Config.dll	3	0	0%	1	5	0	0	0	1	772	428	55.44%
	DisksDB.exe	63	27	43%	4	65	2	0	4	17	13846	6693	48.34%
	DisksDBLibrary.dll	35	0	0%	1	41	2	1	4	13	7058	3180	45.06%
	Language.dll	11	3	27%	1	22	0	0	0	4	3175	1817	57.23%
	MSaccessLayer.dll	6	0	0%	1	8	0	0	0	1	1973	935	47.39%
	Utils.dll	7	0	0%	1	6	1	1	0	0	693	285	41.13%
PocketPC	DataBase.dll	6	0	0%	1	4	0	1	1	0	1299	622	47.88%
	SyncClient.exe	2	0	0%	1	1	0	0	0	0	105	32	30.48%
	PocketDisksDB.exe	9	4	44%	4	8	0	0	0	4	1208	641	53.06%
Bendra:		142	34	24%	15	160	5	3	9	40	30129	14633	48.57%



20 pav. Efektyvių eilučių dalis nuo bendros eilučių sumos paketuose

4. Eksperimentinė DisksDB sistemos statinė analizė

4.1. Klaidų pranešimų paketuose vertinimas pagal taisyklių rinkinius

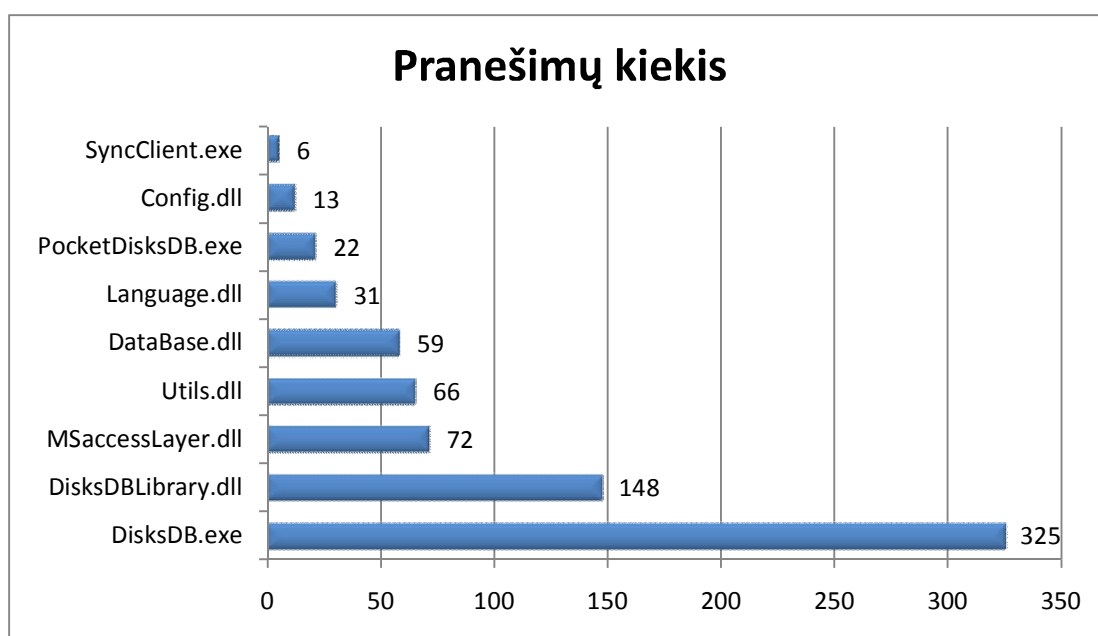
Atlikus pasirinktiems paketams statinę analizę FxCop įrankiu, buvo gauti pranešimai, kurie nurodo pažeidimų vietas ir pačius pažeidimus. Šie pranešimai – tai taisyklių reakcija į paketuose esančius pažeidimus t.y. kiek taisyklė aptiko pažeidimų, tiek pranešimų ji parodys.

Žemiau pateiktoje lentelėje parodyta kiek pranešimų pateikė tam tikra taisyklių grupė kiekvienam paketui. Taip pat parodytas bendras pranešimų kiekis rinkiniuose bei paketuose.

Taisyklių grupių aprašymai pateikti 2.4.3 skyriuje.

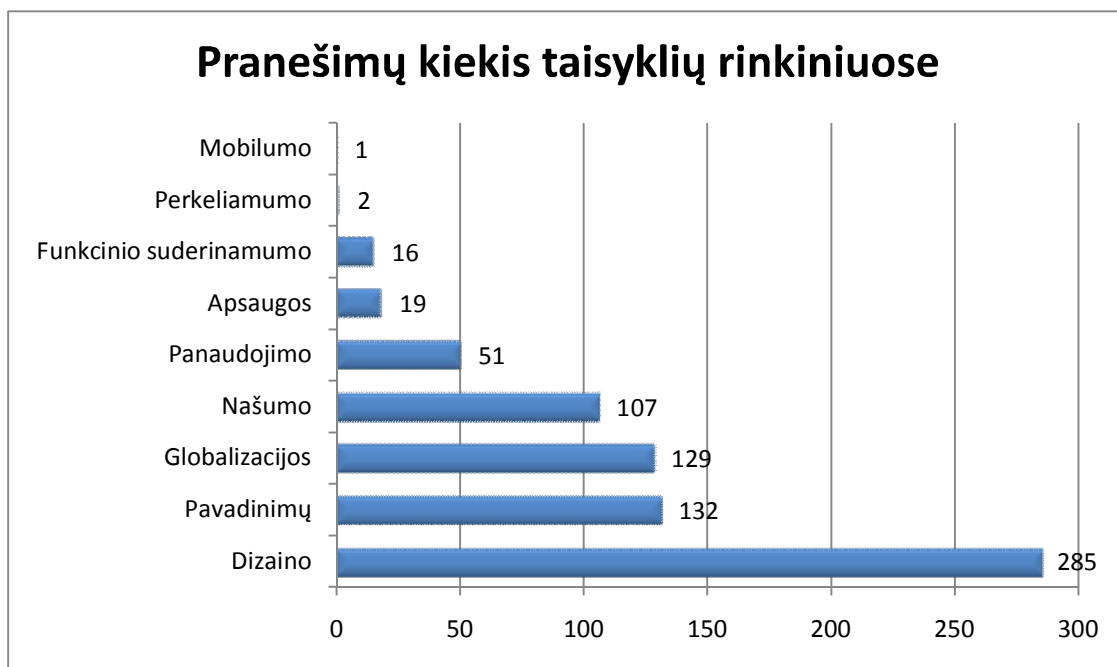
2 lentelė. Pranešimų kiekiai paketuose ir taisyklių rinkiniuose

Taisyklių rinkiniai	Config.dll	DisksDB.exe	DisksDBLibrary.dll	Language.dll	MSaccessLayer.dll	Utils.dll	DataBase.dll	SyncClient.exe	PocketDisksDB.exe	Pranešimų kiekis
Dizaino	6	132	51	6	39	28	16	2	5	285
Globalizacijos	2	73	11	5	12	1	16	0	9	129
Funkcinio suderinamumo	0	9	1	0	1	5	0	0	0	16
Mobilumo	0	1	0	0	0	0	0	0	0	1
Pavadinimų	2	49	57	2	7	8	4	1	2	132
Našumo	2	38	17	11	10	17	10	0	2	107
Perkeliamumo	0	0	0	0	0	2	0	0	0	2
Apsaugos	0	6	0	2	0	2	7	1	1	19
Panaudojimo	1	17	11	5	3	3	6	2	3	51
Bendra:	13	325	148	31	72	66	59	6	22	742



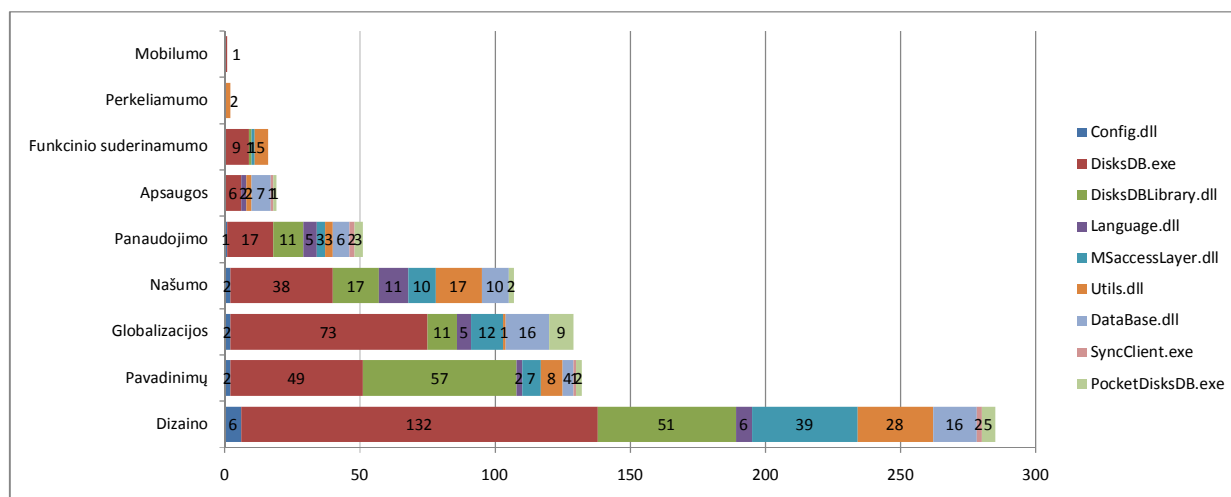
21 pav. Pranešimų kiekiai paketuose

Iš 21 pav. pavaizduotoje diagramoje pateiktų duomenų galima padaryti išvadą, kad daugiausia pranešimų teko DisksDB.exe paketui. Tačiau tai nepasako kokio lygio ar kokio tipo pranešimai buvo pateikti. Todėl nustatymui, kad į šiuos pranešimus reikia reaguoti, reikia atlikti pačių klaidų analizę. Taip pat nėra atsižvelgta į kiekvieno paketo eilučių kiekį, todėl negalima teigti kad daugiausia klaidų tenka DisksDB.exe paketui.



22 pav. Pranešimų kiekiai taisyklių rinkiniuose

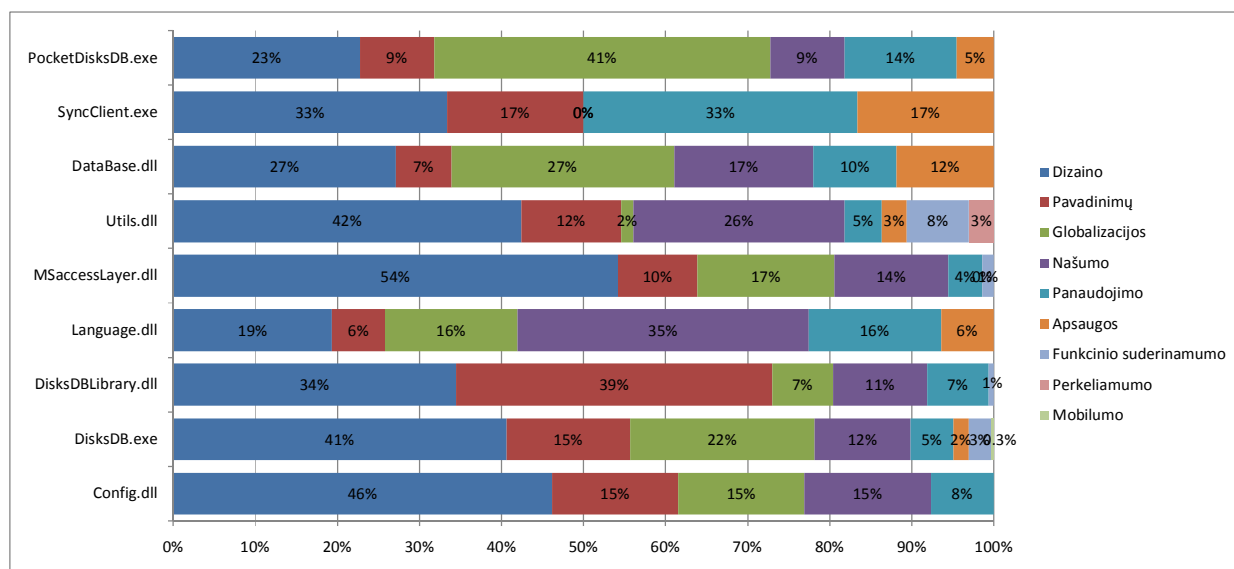
Iš 22 pav. pavaizduotoje diagramoje pateiktų duomenų paaiškėjo, kad daugiausia pranešimų pateikė dizaino taisyklių rinkinio taisyklės. O detalizavus visus pranešimus taisyklių rinkiniuose (žr. 23 pav.) nustatyta, kad didžioji dalis dizaino taisyklių pranešimų teko DisksDB.exe paketui.



23 pav. Pranešimų kiekių paketuose taisyklių rinkiniuose

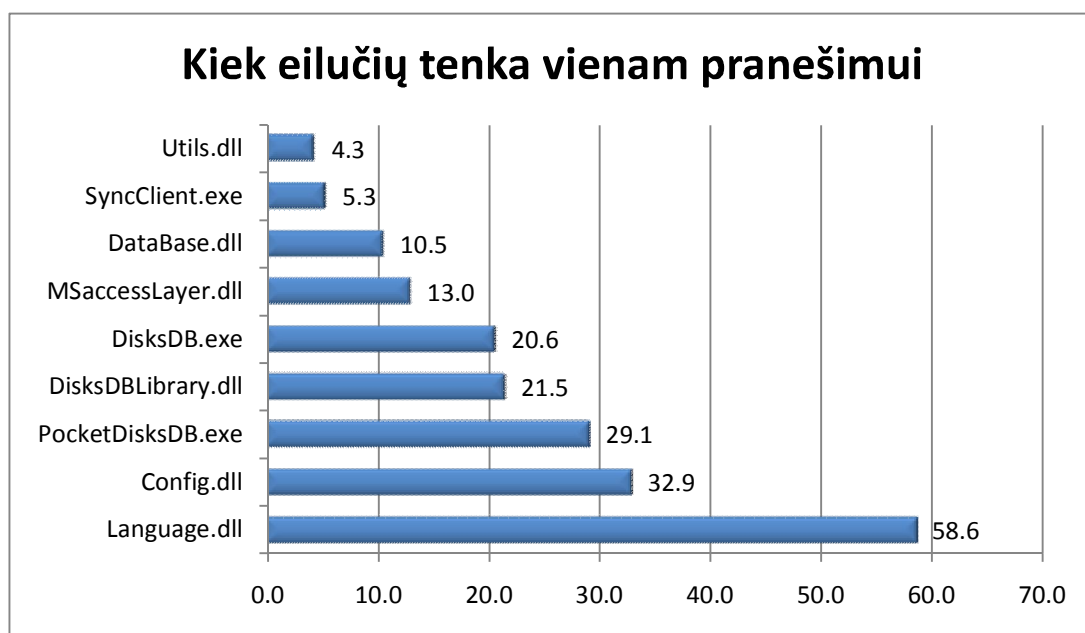
Palyginus kiekviename pakete esančius santykinus klaidų tipų dydžius (žr. 24 pav.) galima nesunkiai pastebėti, kad beveik kiekviename pakete vyrauja dizaino klaidos, išskyrus

DisksDBLibrary.dll, kuriame daugmaž vienodai vyrauja tiek dizaino tiek pavadinimų klaidos, ir Language.dll, kuriame daugiausia yra našumo klaidų o tik po to seka dizaino klaidos.



24 pav. Santykiniai klaidų tipų dydžiai paketuose

Apskaičiavus kiekvienam paketui tenkančių klaidų pranešimų ir aktyvių eilučių (žr. 1 lentelė.), kurioms jie pateikti, santykį (žr. 25 pav.), buvo nustatyta, kad daugiausia klaidų atitenka paketams Utils.dll ir SyncClient.exe. Šis santykis parodo, kas kiek eilučių pateikiamas pranešimas.



25 pav. Eilučių kiekiai vienam klaidos pranešimui kiekviename pakete

4.2. Klaidų pranešimų paketuose vertinimas pagal pranešimų lygį

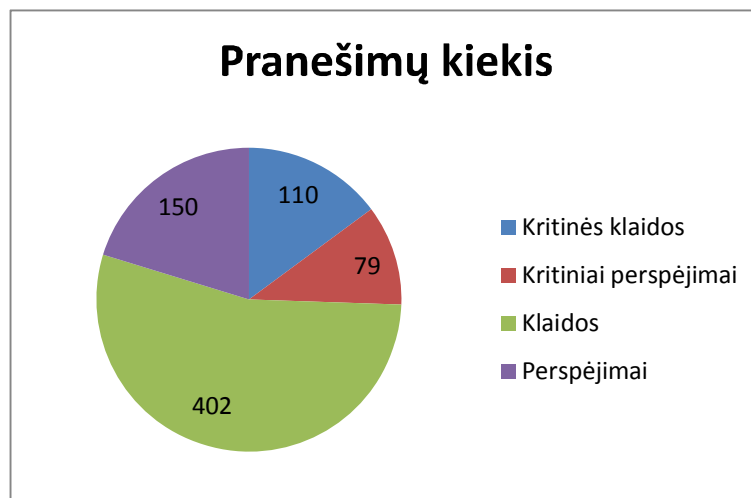
Nustačius kokiam pakete kokio tipo klaidų daugiausia, reikia nustatyti į kuriuos iš jų reikia reaguoti pirmiausia. Tam reikia išanalizuoti pateiktų klaidų pranešimų lygius.

Žemiau patiektoje lentelėje atvaizduojami pranešimų lygių kiekiai paketuose.

3 lentelė. Pranešimų lygiai paketuose

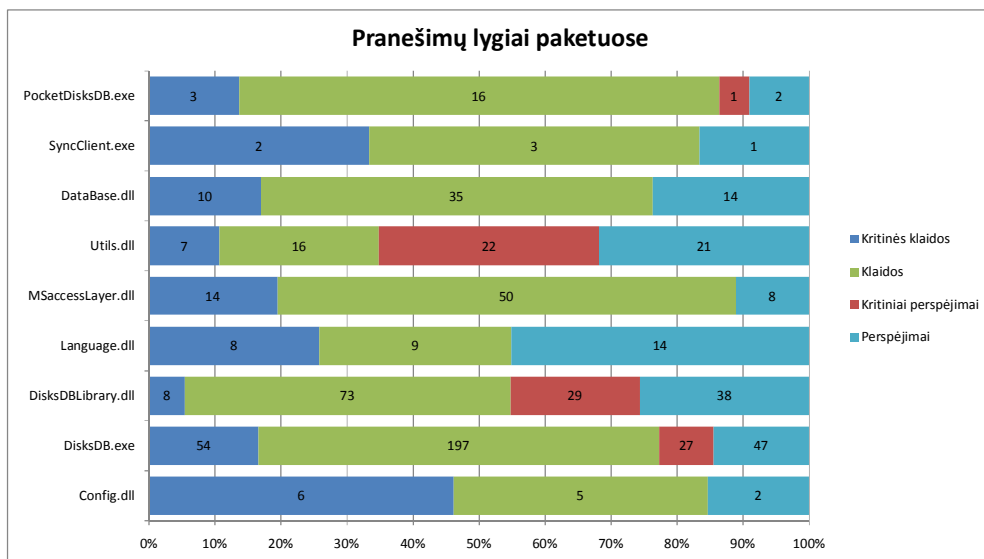
Pranešimų lygiai	Config.dll	DisksDB.exe	DisksDBLibrary.dll	Language.dll	MSaccessLayer.dll	Utils.dll	DataBase.dll	SyncClient.exe	PocketDisksDB.exe	Pranešimų kiekis
Kritinės klaidos	6	54	8	8	14	7	10	2	3	112
Klaidos	5	197	73	9	50	16	35	3	16	404
Kritiniai perspėjimai	0	27	29	0	0	22	0	0	1	79
Perspėjimai	2	47	38	14	8	21	14	1	2	147
Bendra:	13	325	148	31	72	66	59	6	22	742

Apibendrinus pranešimų lygių paketuose duomenis, nustatyta, kiek ir kokių pranešimų buvo pateikta iš viso (žr. 26 pav.). Daugiausia pateikta klaidų lygio pranešimai, kurie užima daugiau nei pusę visų pateiktų pranešimų. Klaidų lygių aprašymai pateikti 2.4.3 skyriuje.



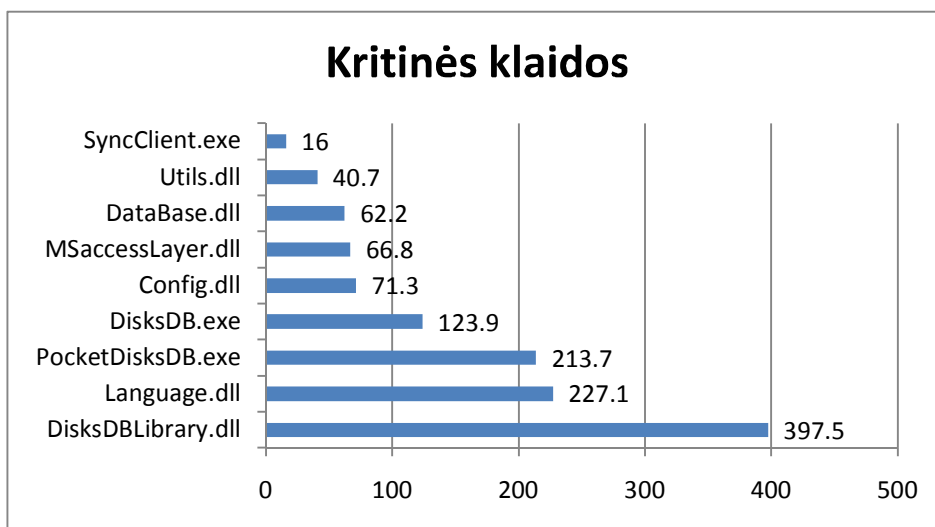
26 pav. Pranešimų lygių kiekiai

Šie pranešimai pasiskirtę paketuose (žr. 27 pav.). Iš šių pasiskirstymų galima nustatyti, kad santykinai didžiausius pranešimų kiekius, į kuriuos reikia pirmiausia reaguoti, turi paketai Config.dll, SyncClient.exe ir Language.dll, tačiau nustatyti kuriame pakete jų yra daugiausia galima apskaičiavus kiekvienam paketui tenkančių klaidų pranešimų (žr. 3 lentelė.) ir aktyvių eilučių (žr. 1 lentelė.), kurioms jie pateikti, santykį (žr. 28 pav., 29 pav., 30 pav. ir 31 pav.).

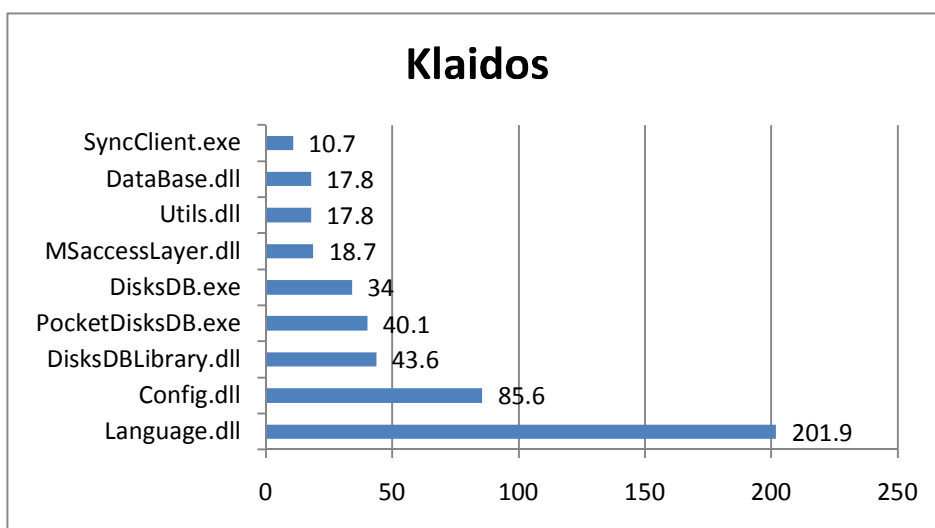


27 pav. Pranešimų lygiai paketuose

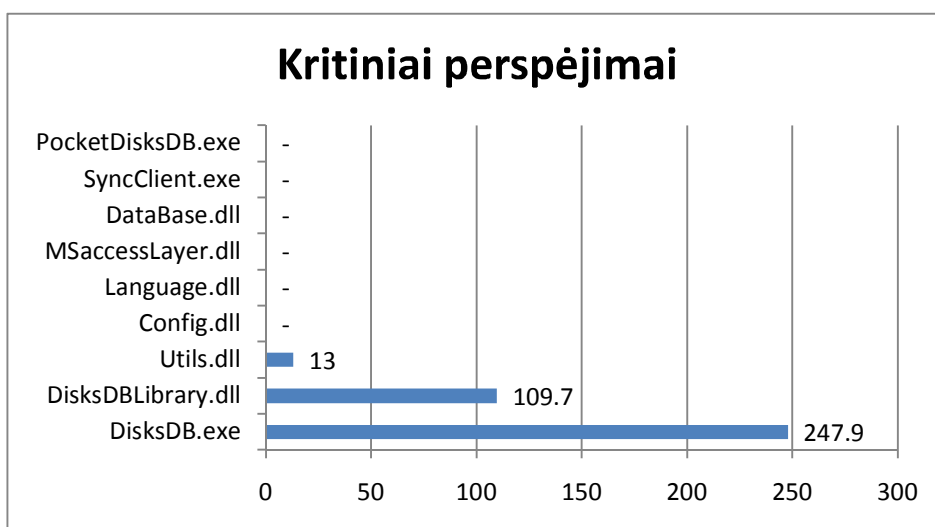
Visa tai atlikus, galima pamatyti, kokiuose paketuose tam tikro lygio klaidų yra daugiausia. Žemiau pateiktose diagramose matome, kad pagrindiniai paketai kuriems tenka daugiausia visų lygių pranešimai yra SyncClient.exe, Utils.dll ir DataBase.dll, nes vienas pranešimas tenka palyginti mažam kiekiui eilučių. O mažiausiai kritinių klaidų turi paketas DisksDBLibrary.dll, klaidų – Language.dll, kritinių perspėjimų – DisksDB.exe, perspėjimų – PocketDisksDB.exe. Reikia pastebėti, kad kritinių perspėjimų turi tik paketai Utils.dll, DisksDBLibrary.dll ir DisksDB.exe.



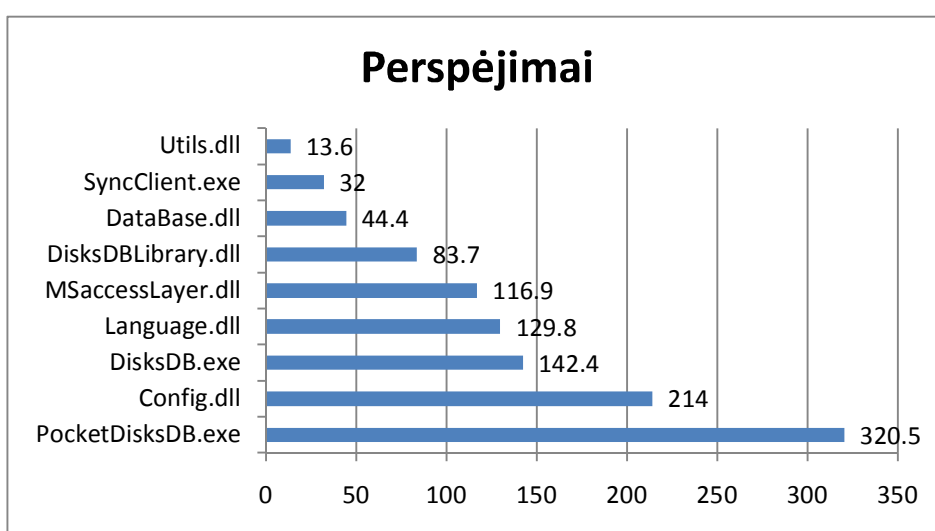
28 pav. Eilučių skaičius paketuose, kurioms tenka vienas kritinės klaidos lygio pranešimas



29 pav. Eilučių skaičius paketuose, kurioms tenka vienas klaidos lygio pranešimas



30 pav. Eilučių skaičius paketuose, kurioms tenka vienas kritinio perspėjimo lygio pranešimas



31 pav. Eilučių skaičius paketuose, kurioms tenka vienas perspėjimo lygio pranešimas

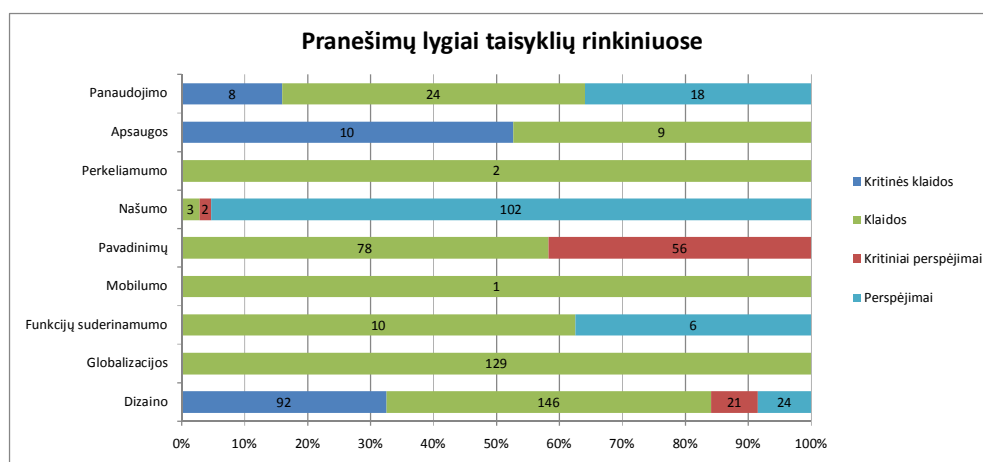
4.3. Klaidų pranešimų taisyklių rinkiniuose vertinimas pagal pranešimų lygius

Kad nustatyti į kurio taisyklių rinkinio taisyklių pranešimus reikia reaguoti pirmiausia, reikia nustatyti, kiek kuriame rinkinyje yra atitinkamo lygio pranešimų.

4 lentelė. Pranešimų lygiai kiekviename taisyklių rinkinyje

Taisyklių rinkiniai	Kritinės klaidos	Kritiniai perspėjimai	Klaidos	Perspėjimai	Pranešimų kiekis
Dizaino	92	21	146	24	283
Globalizacijos	0	0	129	0	129
Funkcinio suderinamumo	0	0	10	6	16
Mobilumo	0	0	1	0	1
Pavadinimų	0	56	78	0	134
Našumo	0	2	3	102	107
Perkeliamumo	0	0	2	0	2
Apsaugos	10	0	9	0	19
Panaudojimo	8	0	24	18	50
Bendra:	110	79	402	150	741

Kad būtų aiškiau šie duomenys pateikiami grafiškai (žr. 32 pav.).



32 pav. Pranešimų lygiai taisyklių rinkiniuose

4.4. Pažeistų taisyklių rinkiniuose tyrimas

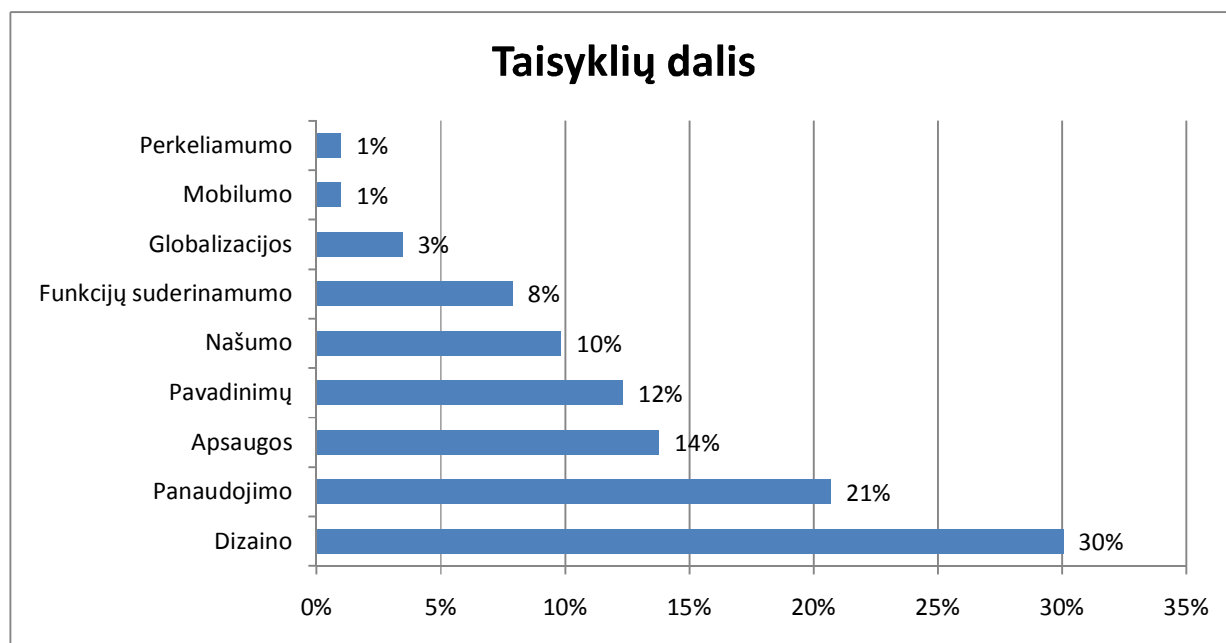
Kiekvienas taisyklių rinkinys turi tam tikrą taisyklių kiekį. Kiekviena taisyklė gali pateikti keletą pranešimų. Tačiau pranešimai parodo kokioj vietoj buvo padarytas pažeidimas, o jų kiekis – kiek kartų buvo pažeista taisyklė.

Žemiau pateiktoje lentelėje pateikti taisyklių rinkiniuose esančių taisyklių kiekis, jų dalis rinkiniuose nuo bendro taisyklių kiekio, pažeistų taisyklių kiekis, kokia rinkinio taisyklių dalis buvo pažeista, bei pažeistų taisyklių dalis nuo bendro pažeidimų kiekio. Buvo nustatyta, kad bendras visuose taisyklių rinkiniuose esančių taisyklių kiekis yra 203, o iš jų pažeista statinės analizės metu – 63. Tai sudaro 31% visų taisyklių.

5 lentelė. Taisyklių pažeidimai

Taisyklių rinkiniai	Taisyklių kiekis	Taisyklių dalis	Pažeistų taisyklių kiekis	Pažeistų taisyklių dalis	Pažeistų taisyklių dalis nuo bendro pažeidimų kiekio
Dizaino	61	30%	19	31%	30%
Globalizacijos	7	3%	4	57%	6%
Funkcijų suderinamumo	16	8%	2	13%	3%
Mobilumo	2	1%	1	50%	2%
Pavadinimų	25	12%	11	44%	17%
Našumo	20	10%	12	60%	19%
Perkeliamumo	2	1%	1	50%	2%
Apsaugos	28	14%	4	14%	6%
Panaudojimo	42	21%	9	21%	14%
Bendra:	203	100%	63	31%	100%

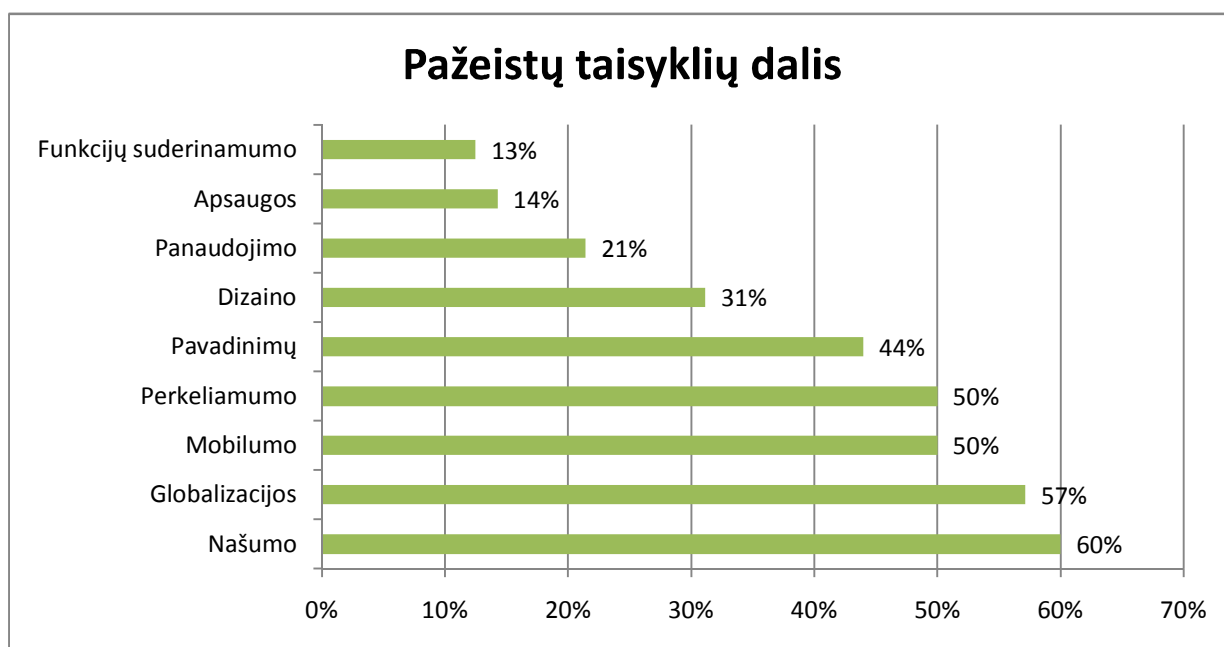
Pateiktoje diagramoje (žr. 33 pav.) matoma, kad didžiausia visų taisyklių dalis tenka dizaino taisyklių rinkiniui t.y. beveik trečdalis. Mažiausia dalis tenka perkeliamumo ir mobilumo taisyklių rinkiniams.



33 pav. Taisyklių dalys taisyklių rinkiniuose nuo bendro taisyklių kiekio

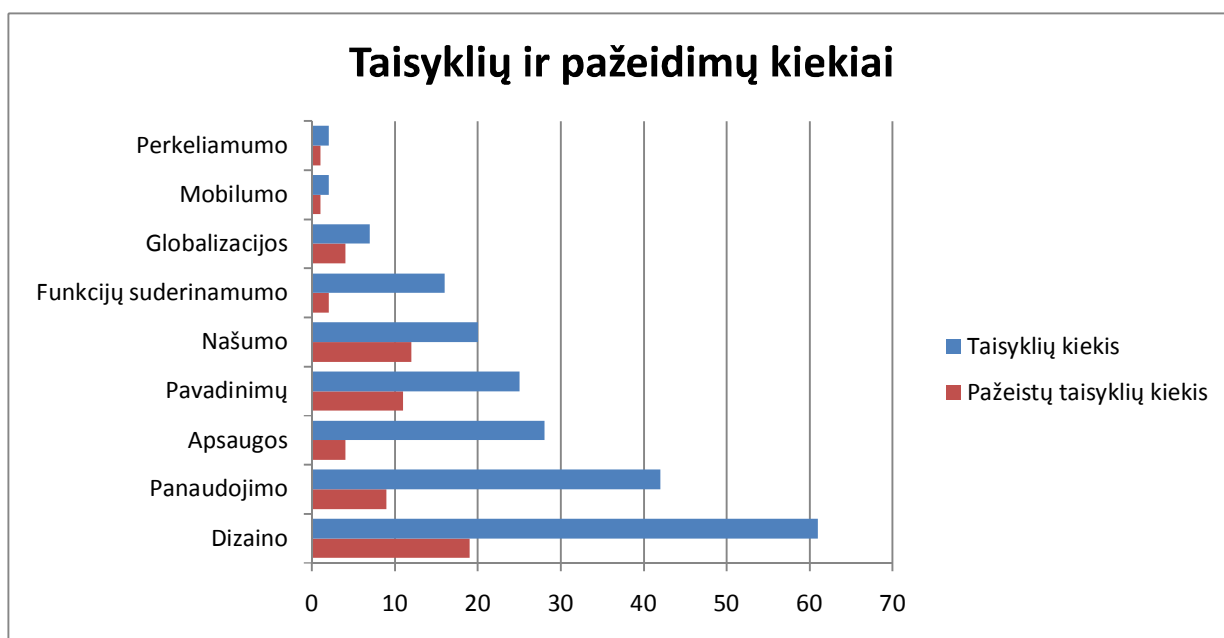
Išanalizavus pažeidimus, buvo nustatyta, kad daugiausia buvo pažeistos našumo ir globalizacijos taisyklių rinkinių taisyklės (žr. 34 pav.), nors, kaip anksčiau buvo išsiaiškinta,

daugiausia taisyklių turi dizaino taisyklių rinkinys. Todėl galima daryti išvadą, kad pažeidimų skaičius nepriklauso nuo taisyklių kiekio.



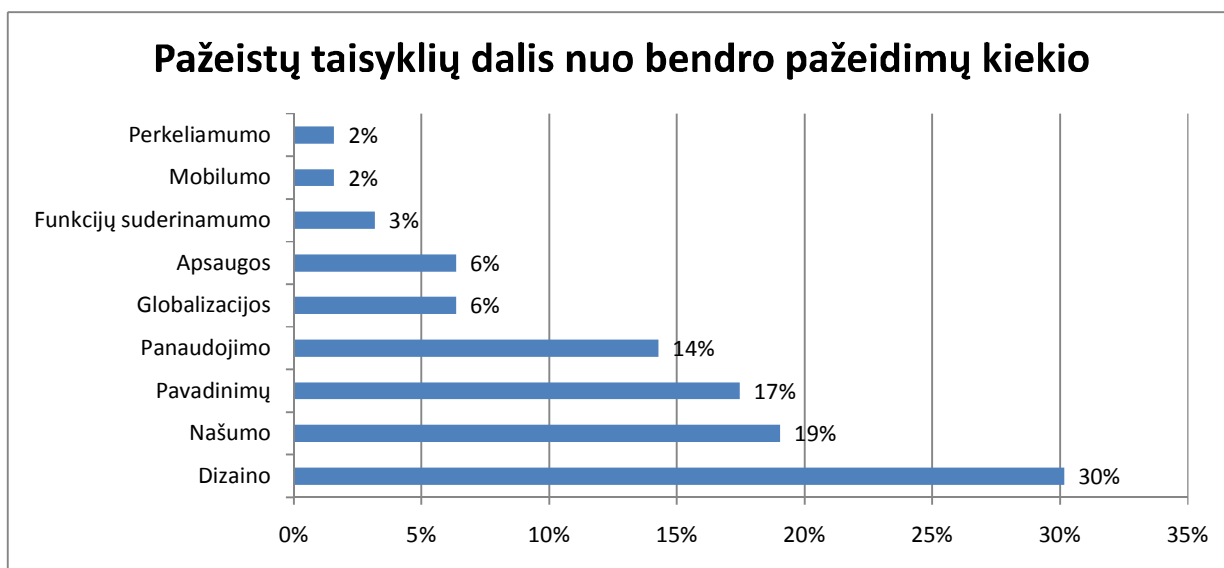
34 pav. Pažeistų taisyklių dalis rinkiniuose

Ankščiau padarytą išvadą galima iliustruoti bendra taisyklių ir jų pažeidimų kiekių palyginimų taisyklių rinkiniuose diagrama (žr. 35 pav.). Joje aiškiai matome bendrus taisyklių kiekius taisyklių rinkiniuose ir kiek iš jų buvo pažeista statinės analizės metu.



35 pav. Taisyklių ir jų pažeidimų kiekių palyginimai taisyklių rinkiniuose

Apskaičiavus kokia visų pažeidimų dalis teko kiekvienam taisyklių rinkiniui (žr. 36 pav.) buvo nustatyta, kad didžiausia pažeidimų dalis teko dizaino taisyklių rinkinio klaidoms. Mažiausia – perkeliamumo ir mobilumo taisyklių rinkiniams.



36 pav. Pažeidimų pasiskirstymas taisyklių rinkiniuose

4.5. Pažeistų taisyklių rinkiniuose vertinimas pagal pažeidimų lygius

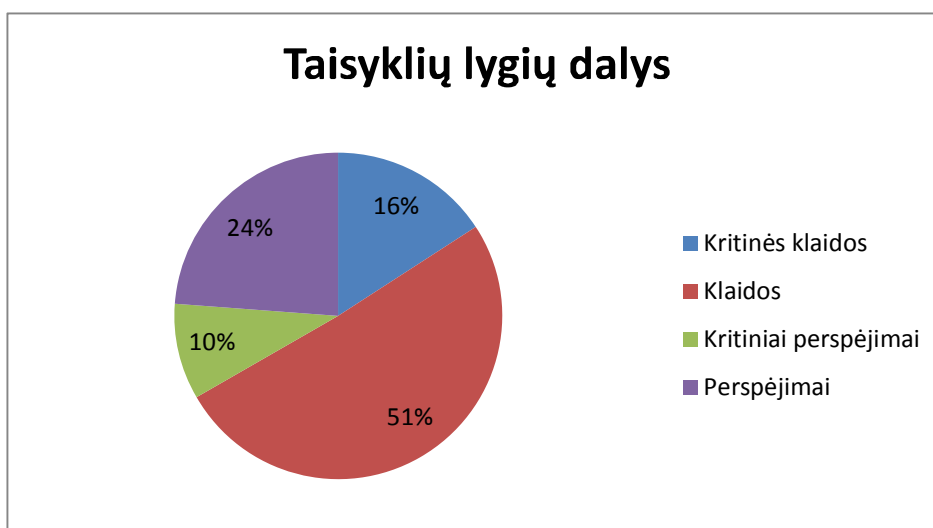
Nustačius bendrą pažeidimų statistiką, įdomu sužinoti kur ir kokie pažeidimai įvyko. Todėl pirmiausia išsiaiškinsime kokio lygio taisyklės buvo pažeistos (žr. 6 lentelė.).

6 lentelė. Pažeidimų lygiai taisyklių rinkiniuose

Taisyklių rinkiniai	Kritinės klaidos		Klaidos		Kritiniai perspėjimai		Perspėjimai	
	Kiekis	Dalis, %	Kiekis	Dalis, %	Kiekis	Dalis, %	Kiekis	Dalis, %
Dizaino	6	32%	10	53%	2	11%	1	5%
Globalizacijos	0	0%	4	100%	0	0%	0	0%
Funkcijų suderinamumo	0	0%	1	50%	0	0%	1	50%
Mobilumo	0	0%	1	100%	0	0%	0	0%
Pavadinimų	0	0%	9	82%	2	18%	0	0%
Našumo	0	0%	1	8%	2	17%	9	75%
Perkeliamumo	0	0%	1	100%	0	0%	0	0%
Apsaugos	3	75%	1	25%	0	0%	0	0%
Panaudojimo	1	11%	4	44%	0	0%	4	44%
Bendra:	10	16%	32	51%	6	10%	15	24%

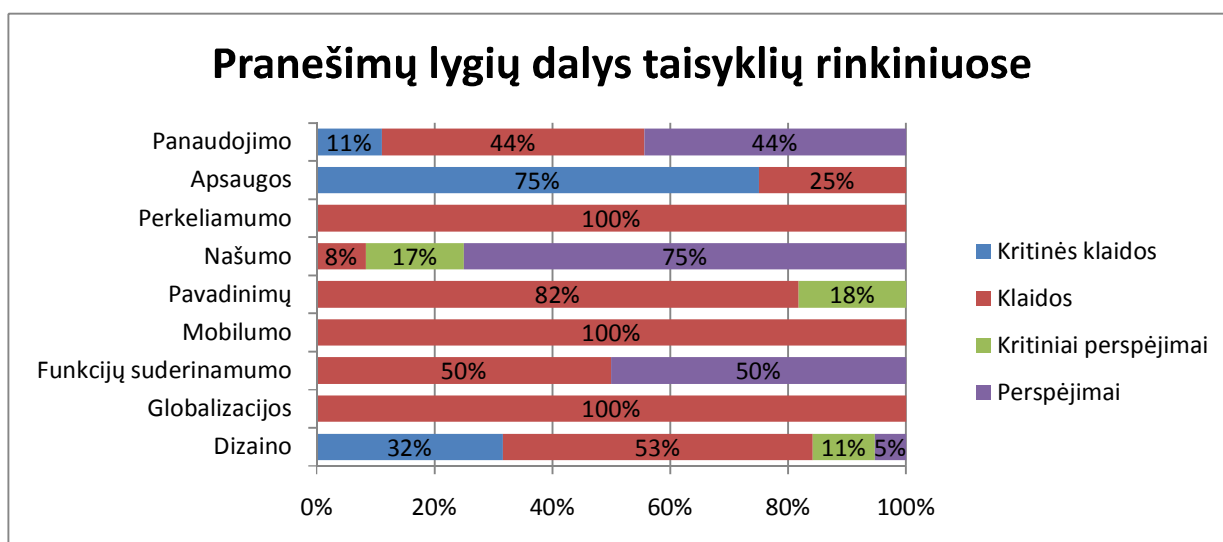
Dėl vaizdumo pateiksime keletą diagramų.

Kiekvienai taisyklei yra priskirtas tam tikras svarbos lygis (Taisyklės struktūros aprašymas pateiktas skyriuje „2.4.4 Taisyklės struktūra“). Atlikus statinę analizę pagal šiuos lygius yra suskirstomos visos pažeistos taisyklės (žr. 37 pav.). Daugiausia taisyklių, kaip matome, buvo klaidų lygio.



37 pav. Pažeistų taisyklių lygių dalys

Šiuos lygius turi visų taisyklių rinkinių taisyklės. Įvykus pažeidimams pagal juos galima suskirstyti taisyklių rinkinius pagal svarbą (žr. 38 pav.). Šiuo atžvilgiu galima teigti, kad daugiausia kritinių pažeidimų lygio turi apsaugos taisyklių rinkinys, klaidų lygio – globalizacijos, perkeliamumo ir mobilumo taisyklių rinkiniai (visos juose pažeistos taisyklės buvo šio lygio), kritinių perspėjimų – pavadinimų taisyklių rinkinyje, perspėjimų – funkcijų suderinamumo.



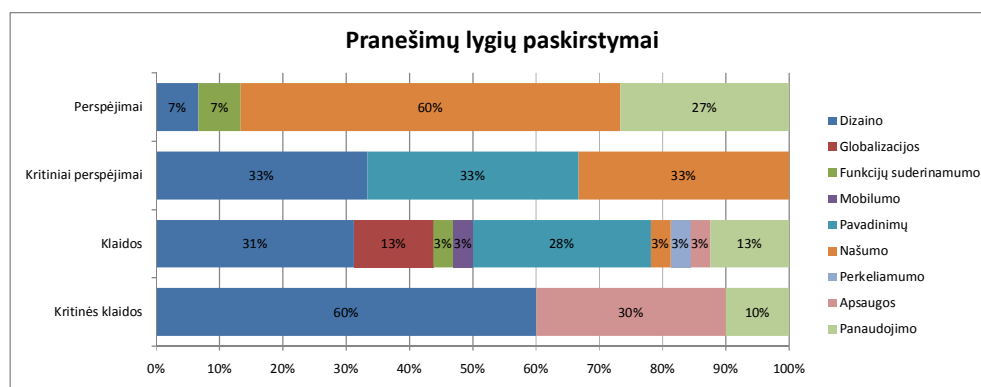
38 pav. Pažeistų taisyklių pažeidimų lygių pasiskirstymas taisyklių rinkiniuose

Tačiau paskaičiavus kiekvieno lygio pažeidimų pasiskirstymus kiekviename rinkinyje (žr. 7 lentelė.) ir juos palyginus (žr. 39 pav.), buvo pastebėta, kad daugiausia kritinių klaidų pažeidimų buvo dizaino taisyklių rinkinyje, klaidų lygio – dizaino ir perkeliamumo rinkiniuose, kritinių perspėjimų lygio – vienodai pasiskirstę dizaino, našumo ir pavadinimų rinkiniuose, perspėjimų lygio – našumo taisyklių rinkinyje.

7 lentelė. Pažeidimų lygių pasiskirstymai taisyklių rinkiniuose

Taisyklių rinkiniai	Kritinės klaidos		Klaidos		Kritiniai perspėjimai		Perspėjimai	
	Kiekis	Dalis, %	Kiekis	Dalis, %	Kiekis	Dalis, %	Kiekis	Dalis, %
Dizaino	6	60%	10	31%	2	33%	1	7%
Globalizacijos	0	0%	4	13%	0	0%	0	0%
Funkcijų suderinamumo	0	0%	1	3%	0	0%	1	7%
Mobilumo	0	0%	1	3%	0	0%	0	0%
Pavadinimų	0	0%	9	28%	2	33%	0	0%
Atlikimo	0	0%	1	3%	2	33%	9	60%
Perkeliamumo	0	0%	1	3%	0	0%	0	0%
Apsaugos	3	30%	1	3%	0	0%	0	0%
Panaudojimo	1	10%	4	13%	0	0%	4	27%

Pagal pateiktą lentelę, pateikiami vaizdūs duomenys:



39 pav. Pažeidimų lygių pasiskirstymai rinkiniuose

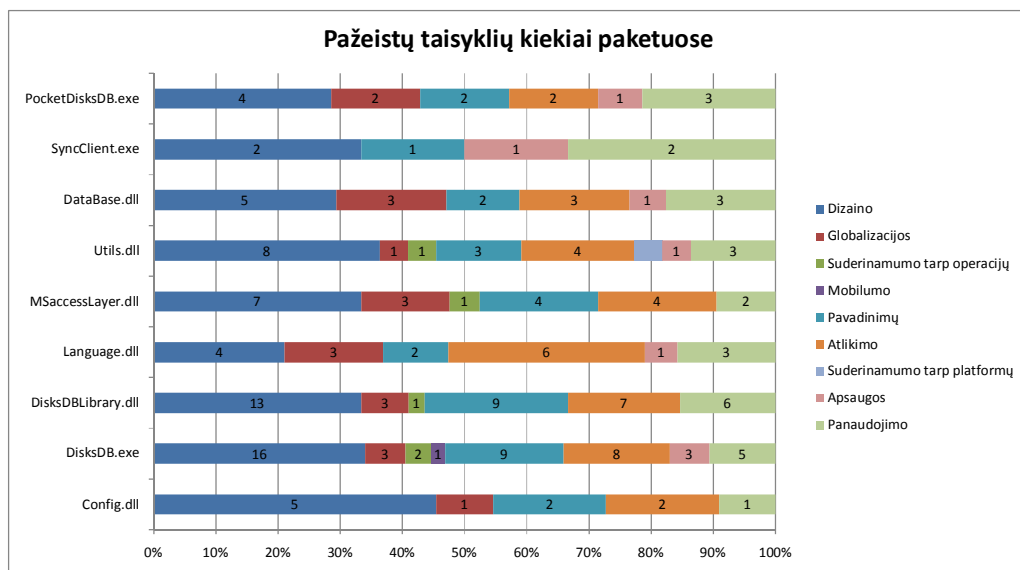
Iš to galima daryti išvadą, kad daugiausia visų lygių pažeidimų, o ypač tų į kuriuos reikia reaguoti pirmiausia, turi dizaino taisyklių rinkinio taisyklės.

4.6. Pažeistų taisyklių paketuose vertinimas pagal taisyklių rinkinius

Atlikus pažeistų taisyklių paketuose rezultatų apibendrinimą, buvo gauti apibendrinti duomenys (žr. 8 lentelė.). Joje pateikiami paketuose pažeistų kiekvieno rinkinio taisyklių kiekiai ir jų dalys paketuose.

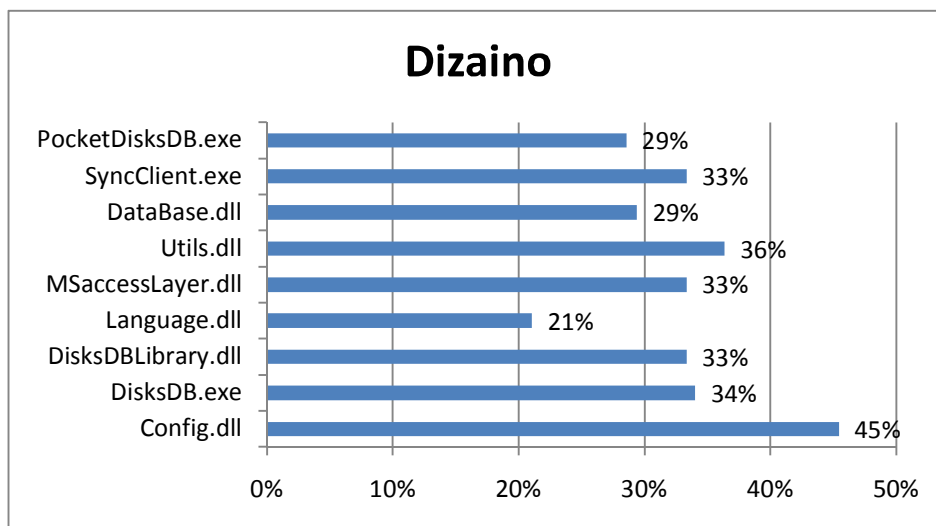
8 lentelė. Taisyklių rinkinių taisyklių kiekiai ir dalys paketuose

Taisyklių rinkiniai	Config.dll		DisksDB.exe		DisksDBLibrary.dll		Language.dll		MSaccessLayer.dll		Utils.dll		DataBase.dll		SyncClient.exe		PocketDisksDB.exe	
	Kiekis	Dalis,%	Kiekis	Dalis,%	Kiekis	Dalis,%	Kiekis	Dalis,%	Kiekis	Dalis,%	Kiekis	Dalis,%	Kiekis	Dalis,%	Kiekis	Dalis,%	Kiekis	Dalis,%
Dizaino	5	45%	16	34%	13	33%	4	21%	7	33%	8	36%	5	29%	2	33%	4	29%
Globalizacijos	1	9%	3	6%	3	8%	3	16%	3	14%	1	5%	3	18%	0	0%	2	14%
Funkcijų suderinamumo	0	0%	2	4%	1	3%	0	0%	1	5%	1	5%	0	0%	0	0%	0	0%
Mobilumo	0	0%	1	2%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%
Pavadinimų	2	18%	9	19%	9	23%	2	11%	4	19%	3	14%	2	12%	1	17%	2	14%
Našumo	2	18%	8	17%	7	18%	6	32%	4	19%	4	18%	3	18%	0	0%	2	14%
Perkeliamumo	0	0%	0	0%	0	0%	0	0%	0	0%	1	5%	0	0%	0	0%	0	0%
Apsaugos	0	0%	3	6%	0	0%	1	5%	0	0%	1	5%	1	6%	1	17%	1	7%
Panaudojimo	1	9%	5	11%	6	15%	3	16%	2	10%	3	14%	3	18%	2	33%	3	21%

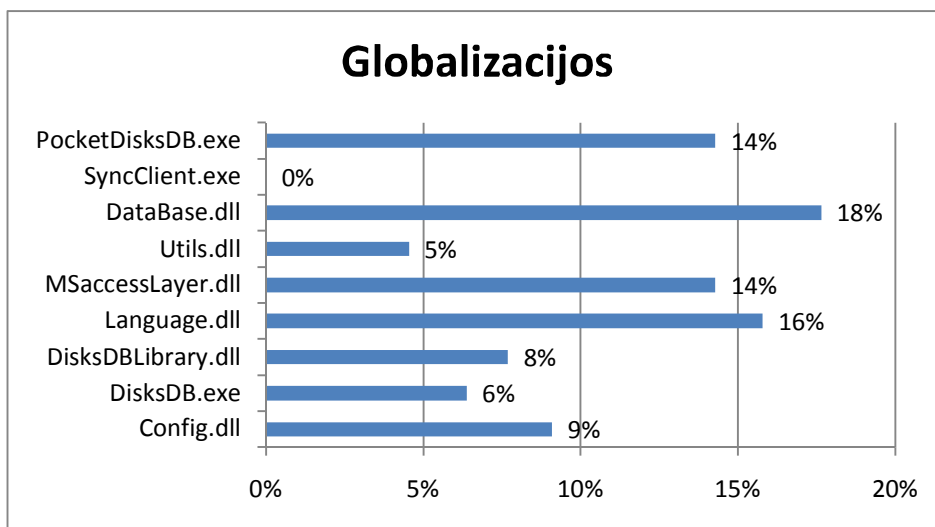


40 pav. Pažeistų taisyklių kiekių paketuose palyginimas

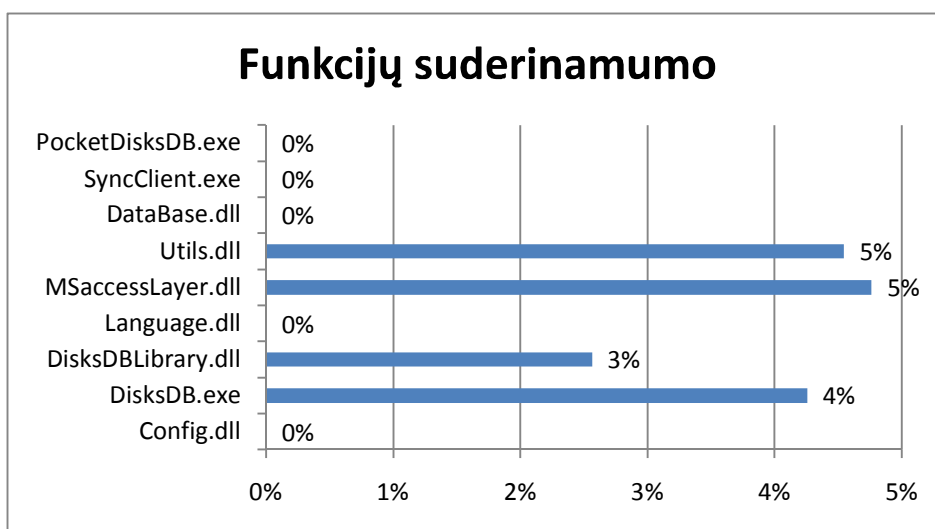
Rinkinių taisyklių pažeidimų paketuose palyginimui yra pateikiamos diagramos (žr. nuo 41 pav. iki 49 pav.). Jose galima pamatyti kokiam pakete tam tikro taisyklių rinkinio taisyklių pažeidimų buvo daugiausia ar mažiausia.



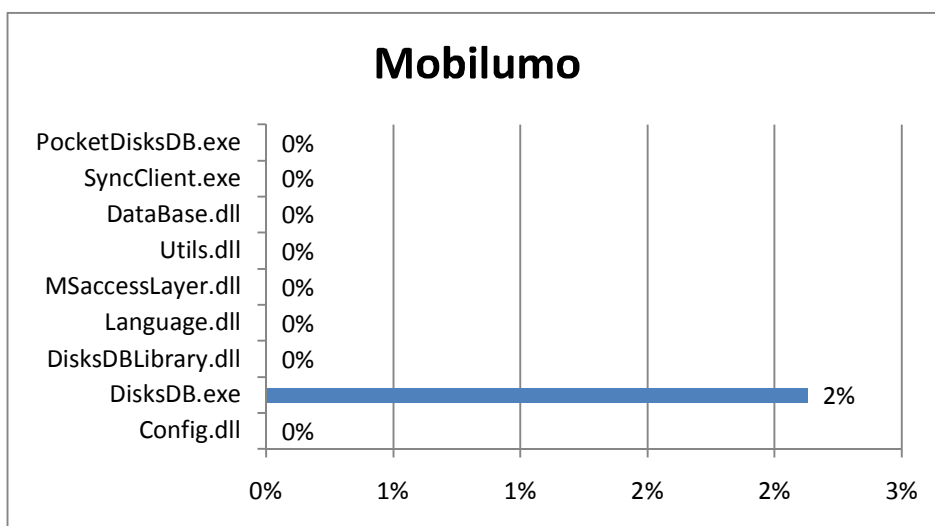
41 pav. Pažeistų dizaino rinkinio taisyklių dalys paketuose



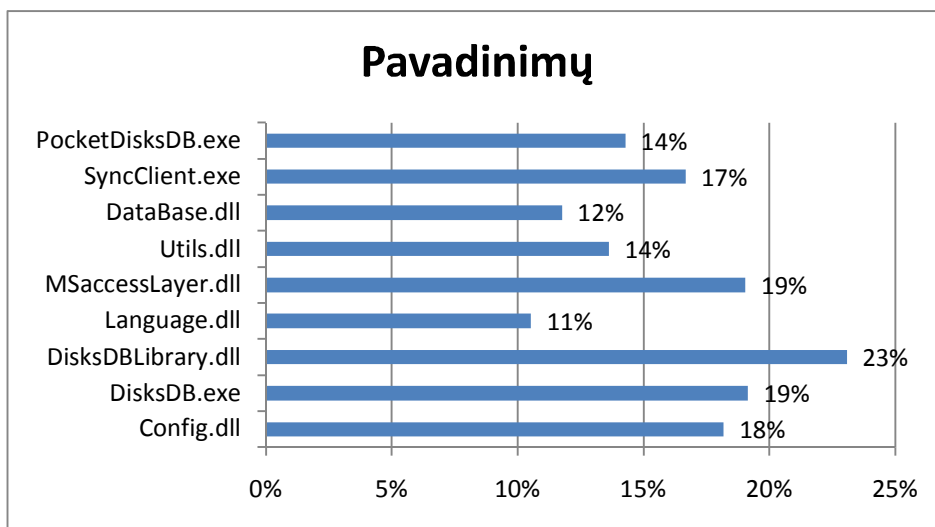
42 pav. Pažeistų globalizacijos rinkinio taisyklių dalys paketuose



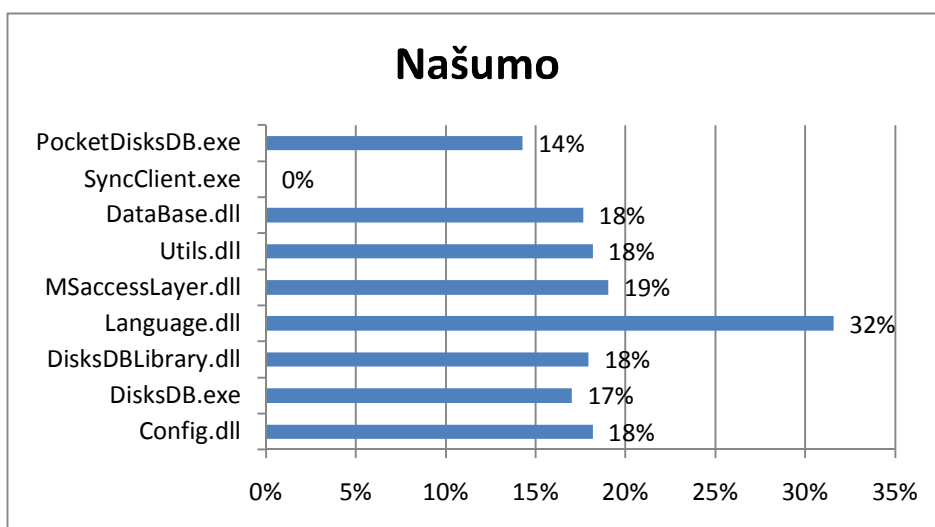
43 pav. Pažeistų funkcijų suderinamumo rinkinio taisyklių dalys paketuose



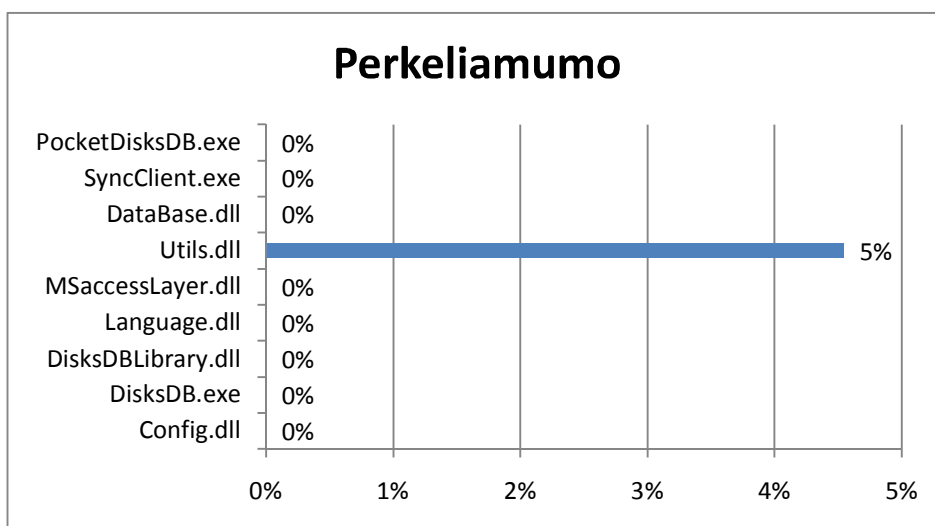
44 pav. Pažeistų mobilumo rinkinio taisyklių dalys paketuose



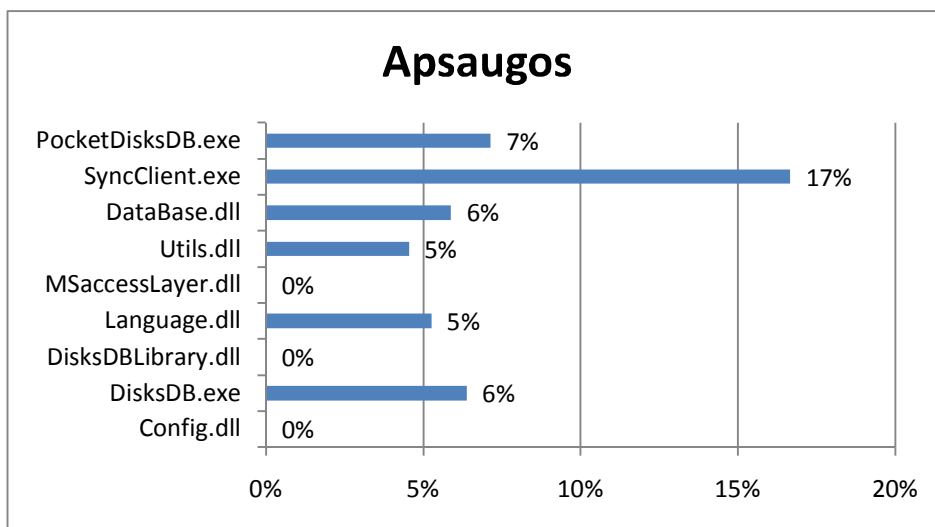
45 pav. Pažeistų pavadinimų rinkinio taisyklių dalys paketuose



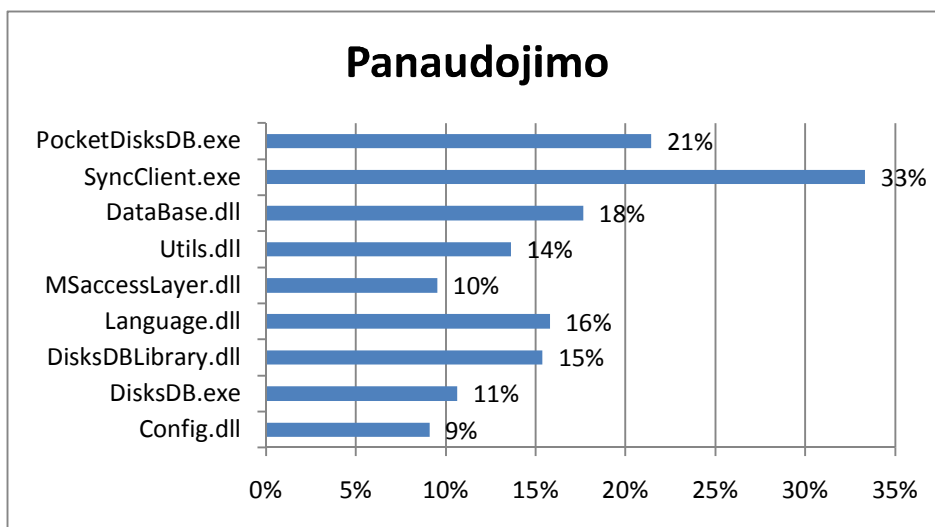
46 pav. Pažeistų našumo rinkinio taisyklių dalys paketuose



47 pav. Pažeistų perkeliavimo rinkinio taisyklių dalys paketuose



48 pav. Pažeistų apsaugos rinkinio taisyklių dalys paketuose



49 pav. Pažeistų panaudojimo rinkinio taisyklių dalys paketuose

4.7. Pažeistų taisyklių tyrimas paketuose

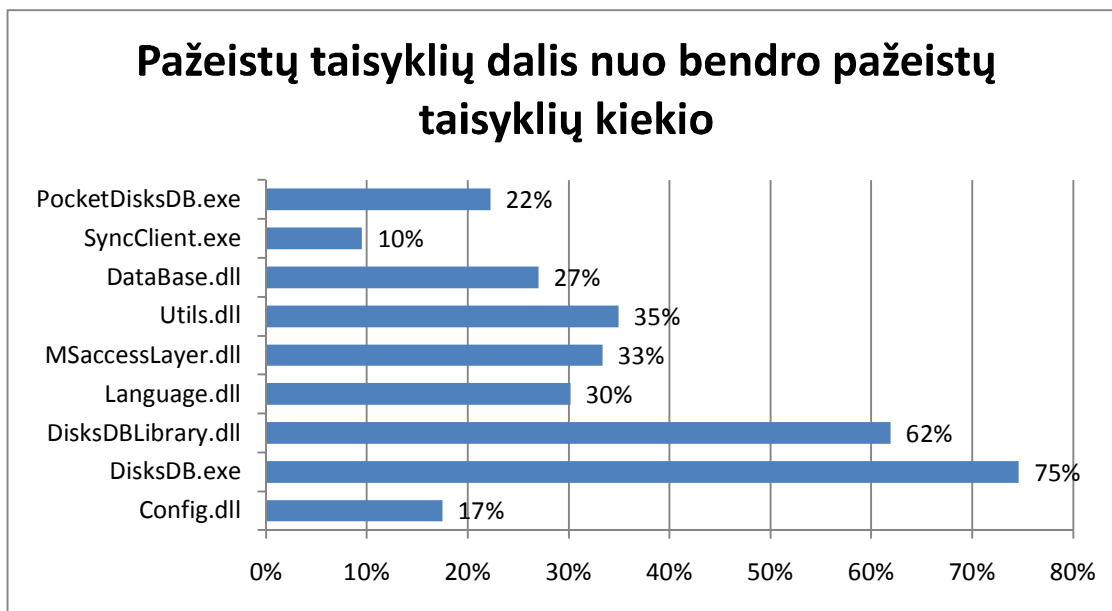
Apibendrinus FxCop pateiktus rezultatus paketuose buvo gauta bendra pažeidimų suvestinė (žr. 9 lentelė.). Joje pateikiama pažeidimų kiekis, jo dalis nuo bendro taisyklių ir nuo bendro pažeidimų kiekio, efektyvių eilučių kiekis bei jų skaičius vienam pranešimui pakete.

9 lentelė. Bendra pažeidimų paketuose suvestinė

Paketas	Pažeistų taisyklių kiekis	Pažeistų taisyklių dalis	Pažeistų taisyklių dalis nuo bendro pažeistų taisyklių kiekio	Efektyvių eilučių kiekis	Eilučių kiekis, kuriom tenka vienas pažeidimas
Config.dll	11	5%	17%	1817	165.2
DisksDB.exe	47	23%	75%	428	9.1
DisksDBLibrary.dll	39	19%	62%	641	16.4
Language.dll	19	9%	30%	6693	352.3
MSaccessLayer.dll	21	10%	33%	622	29.6
Utils.dll	22	11%	35%	935	42.5
DataBase.dll	17	8%	27%	3180	187.1

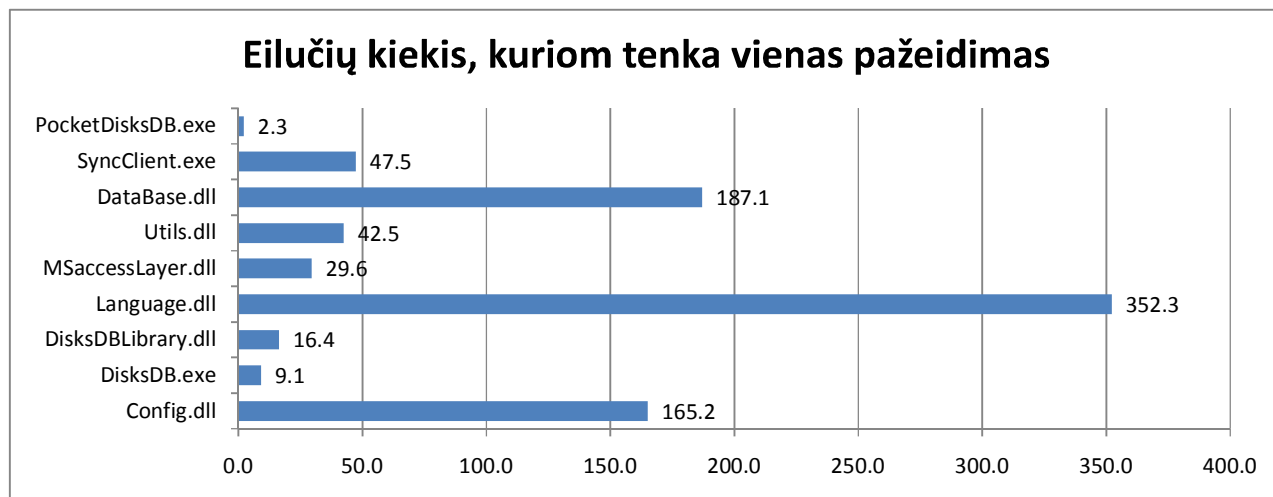
SyncClient.exe	6	3%	10%	285	47.5
PocketDisksDB.exe	14	7%	22%	32	2.3
Bendra:	63	31%		14633	232.3

Palyginus pažeidimus kiekvienam paketui galima daryti išvadą, kad daugiausia pažeidimų turi DisksDB.exe ir DisksDBLibrary.dll paketai, o mažiausia SyncClient.exe ir Config.dll.



50 pav. Pažeistų taisyklių dalis nuo bendro pažeistų taisyklių kiekio

Tačiau šios reikšmės apskaičiuotos nevienodo dydžio projektams. Todėl suvienodinant paskaičiuota kiek eilučių tenka vienam pažeidimui kiekviename pakete.



51 pav. Eilučių kiekis, kuriom tenka vienas pažeidimas

Šiuo požiūriu daugiausia pažeidimų tenka PocketDisksDB.exe ir DisksDB.exe.

4.8. Naujos taisyklės testavimas

Patobulinat testavimą buvo sukurta nauja taisyklė, kuri turėjo aptikti papildomų klaidų. Atlikus bandymą su sistema ši taisyklė buvo pažeista du kartus: DisksDB.exe ir DisksDBLibrary.dll.

5. Išvados

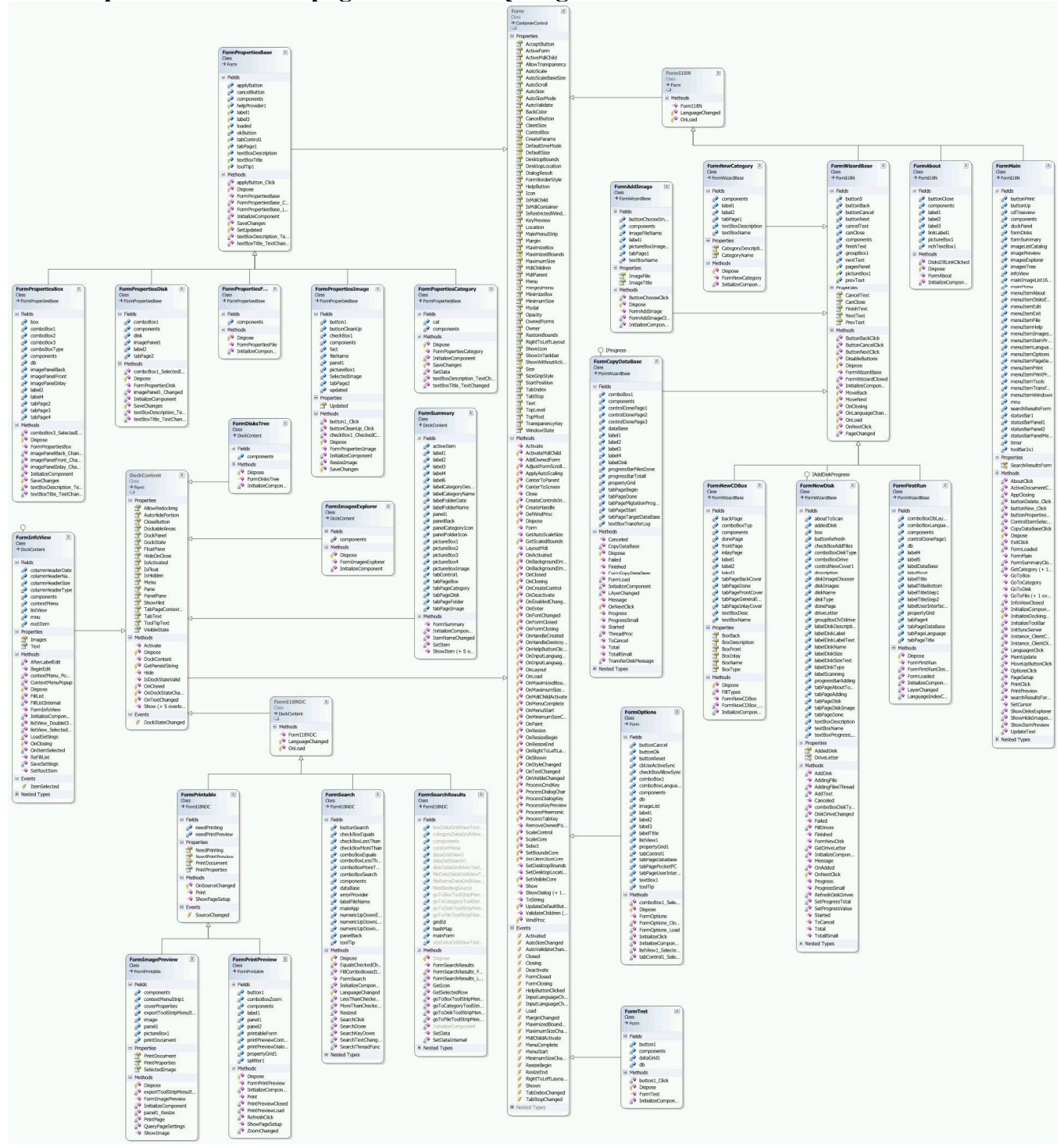
- Pagal užsakovo reikalavimus, buvo suprojektuota ir sukurta diskų tvarkymo sistema skirta darbui ant dviejų platformų;
- Apžvelgti statinės analizės metodai, statinio kodo analizės veikimas, ir pasirinktas tinkamiausias sukurto sistemos kodo analizei įrankis FxCop
- FxCop įrankiu buvo išanalizuotas sistemos programinis kodas, kuris pateikė rezultatus;
- Atlikta statinės analizės rezultatų analizė, kurios metu buvo nustatyta kurios klaidos bus taisomos o kurios ne;
- Buvo sukurta statinės kodo analizės taisyklė FxCop įrankiui, kuri parodė dar neaptiktas sistemos klaidas;
- Atsižvelgiant į susistemintus rezultatus buvo pataisytos esminės sistemoje esančios klaidos.

6. Literatūra

- [1] Schwartzbach M. I., Lecture Notes on Static Analysis, BRICS, Department of Computer Science, University of Aarhus, Denmark, 2004
- [2] Chen R., Köb D., Peischl B., Wotawa F., Static and Dynamic Analysis in Automated Debugging, 2007
- [3] Wagner D., Dean D., Intrusion Detection via Static Analysis, 2007
- [4] Livshits V. B., Lam M. S., Finding Security Vulnerabilities in Java Applications with Static Analysis, Computer Science Department, Stanford University
- [5] Chess B., West J., Secure Programming with Static Analysis, Pearson Education, Inc., 2007. ISBN 0-321-42477-8
- [6] Chess B., McGraw G., Static Analysis for Security, IEEE Security and Privacy, 2004, psl. 76 – 79.
- [7] Sommerville I., Software Engineering, 6th edition, 2000 ,Chapter 19
- [8] Kresowaty J., FxCop and Code Analysis: Writing Your Own Custom Rules, 2008
- [9] Microsoft Corporation, Microsoft Dynamics ISV Software Solution Test Guidelines Microsoft® Dynamics™ AX 4.0, 2007
- [10] Managed Code Analysis, chapter 8, žiūrēta [2008-10-20], prieiga internete < http://media.techtarget.com/digitalguide/images/Misc/professionalvsts_ch08.pdf >
- [11] Vipindeep V., Pankaj J., List of Common Bugs and Programming Practices to avoid them, Indian Institute of Technology, Kanpur, 2005
- [12] Binkley D., Source Code Analysis: A Road Map, IEEE Computer Society, 2007
- [13] Kleidermacher D. N., Integrating Static Analysis into a Secure Software Development Process, IEEE Computer Society, 2008
- [14] Shah A., .NET Code Analysis Tool : FxCop, 2007 Issue žiūrēta [2008-11-04], prieiga internete < http://www.einfochips.com/download/dash_may_tech.pdf >

7. Priedai

1 priedas. DisksDB pagrindinė klasių diagrama



2 priedas. FxCop sugeneruotos ataskaitos fragmentas

FxCop 1.35 Analysis Report

Expand All **Collapse All**

{ } DisksDB.Pocket.Database (1)
1 Message for DisksDB.Pocket.Database
Message Certainty Resolution

Level	Message Level	Certainty	Resolution
▶	Error	80	The compound word 'DataBase' in namespace 'DisksDB.Pocket.Database' exists as a discrete term. If your usage is intended to be single word, case it as 'Database'.

{ } DisksDB.Pocket.Utilis (1)
1 Message for DisksDB.Pocket.Utilis
Message Level Certainty Resolution

Level	Message Level	Certainty	Resolution
▶	CriticalWarning	75	Correct the spelling of the unrecognized token 'Utilis' in namespace 'DisksDB.Pocket.Utilis'.

\$(ProjectDir)/source/DesktopPC/DisksDB/bin/Release/Config.dll (13)
config.dll (13)
4 Messages for config.dll

Message Level	Certainty	Resolution	
▶	CriticalError	95	No valid permission requests were found for assembly 'Config'. You should always specify the minimum security permissions using SecurityAction.RequestMinimum.
▶	CriticalError	95	Sign 'Config' with a strong name key.
▶	Error	95	'Config' should be marked with CLSCompliantAttribute and its value should be true.
▶	CriticalError	95	Mark 'Config' as ComVisible(false) at the assembly level, then mark all types within the assembly that should be exposed to Com clients as ComVisible(true).

{ } DisksDB.Config (9) Config (9)
2 Messages for Config

Message Level	Certainty	Resolution	
▶	Error	95	The type name 'Config' conflicts in whole or in part with the namespace name 'DisksDB.Config'. Change either name to eliminate the conflict.
▶	CriticalError	95	Implement IDisposable on 'DisksDB.Config.Config' as it instantiates members of the following IDisposable types: DisksDB.Config.DataSetConfig

.cctor() (1)
1 Message for .cctor()

Message Level	Certainty	Resolution	
▶	Warning	90	Config.Config() initializes field _jsntace of type DisksDB.Config.Config to null. Remove this initialization as it will be done automatically by the runtime.

.ctor() (1)
1 Message for .ctor()

Message Level	Certainty	Resolution	
▶	CriticalError	95	Modify 'Config.Config()' to catch a more specific exception than 'System.Exception' or rethrow the exception.

★ AppID (2)
2 Messages for AppID

Message Level	Certainty	Resolution	
▶	Warning	95	The 'this' parameter (or 'Me' in VB) of Config.AppID:String is never used. Mark the member as static (or Shared in VB) or use 'this'/'Me' in the method body least one property accessor, if appropriate.
▶	Error	85	'Id' is an abbreviation and therefore is not subject to acronym casing guidelines. Correct the capitalization of 'ID' in member name 'Config.AppID:String' by it to 'Id'.

GetValue(System.String,System.Int32):System.Int32 (1)
1 Message for GetValue(System.String,System.Int32):System.Int32
Message Certainty Resolution

Level	Message Level	Certainty	Resolution
▶	Error	95	Config.GetValue(String, Int32):Int32 makes a call to System.Int32.Parse(System.String) that does not explicitly provide an IFormatProvider. This should be with a call to System.Int32.Parse(System.String,IFormatProvider).

Save():System.Void (1)
1 Message for Save():System.Void

Message Level	Certainty	Resolution	
▶	CriticalError	95	Modify 'Config.Save():Void' to catch a more specific exception than 'System.Exception' or rethrow the exception.

SetValue(System.String,System.Int32):System.Void (1)
1 Message for SetValue(System.String,System.Int32):System.Void
Message Certainty Resolution

Level	Message Level	Certainty	Resolution
▶	Error	95	Config.SetValue(String, Int32):Void makes a call to System.Int32.ToString that does not explicitly provide an IFormatProvider. This should be replaced with System.Int32.ToString(System.IFormatProvider).

\$(ProjectDir)/source/DesktopPC/DisksDB/bin/Release/Language.dll (31)
language.dll (31)
4 Messages for language.dll


Message Level	Certainty	Resolution	
▶	CriticalError	95	No valid permission requests were found for assembly 'Language'. You should always specify the minimum security permissions using SecurityAction.RequestMinimum.
▶	CriticalError	95	Sign 'Language' with a strong name key.
▶	Error	95	'Language' should be marked with CLSCompliantAttribute and its value should be true.
▶	CriticalError	95	Mark 'Language' as ComVisible(false) at the assembly level, then mark all types within the assembly that should be exposed to Com clients as ComVisible(true)

{ } DisksDB.Language (27) FormI18N (1)
LanguageChanged(System.Object,System.EventArgs):System.Void (1)
1 Message for LanguageChanged(System.Object,System.EventArgs):System.Void

Message Level	Certainty	Resolution	
▶	CriticalError	40	Consider making FormI18N.LanguageChanged(Object, EventArgs):Void not externally visible.

FormI18NDC (2)
1 Message for FormI18NDC
Message Certainty Resolution

Level	Message Level	Certainty	Resolution
▶	Error	85	Correct the capitalization of type name 'FormI18NDC'.

 LanguageChanged(System.Object,System.EventArgs):System.Void (1)
 1 Message for LanguageChanged(System.Object,System.EventArgs):System.Void
Message

Level Certainty Resolution

- ▶ **CriticalError** 40 Consider making FormI18NDC.LanguageChanged(Object, EventArgs):Void not externally visible.
- FormTextsEditor (3)
 - 🔗 .ctor() (1)
 - ▶ **1 Message for .ctor()**
 - Message Certainty Resolution
 - ▶ **Error** 90 Be sure to set the DataSet Locale property wherever 'FormTextsEditor.FormTextsEditor()' creates an instance of that type.
 - 🔗 FormTextsEditor_Load(System.Object, System.EventArgs):System.Void (1)
 - ▶ **1 Message for FormTextsEditor_Load(System.Object, System.EventArgs):System.Void**
 - Message Certainty Resolution
 - ▶ **Error** 90 Be sure to set the DataTable Locale property wherever 'FormTextsEditor.FormTextsEditor_Load(Object, EventArgs):Void' creates an instance of that type.
 - 🔗 InitializeComponent():System.Void (1)
 - ▶ **1 Message for InitializeComponent():System.Void**
 - Message Certainty Resolution
 - ▶ **Error** 95 FormTextsEditor.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument resource table instead: 'Texts'
 - ▶ **Error** 95 FormTextsEditor.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument resource table instead: 'Update'
- GenericI18N (3)
 - ▶ **1 Message for GenericI18N**
 - Message Certainty Resolution
 - ▶ **Warning** 75 'DisksDB.Language.GenericI18N' is an internal class that is apparently never instantiated. If so, remove the code from the assembly. If this class is intended to contain only static methods, consider adding a private constructor to prevent the compiler from generating a default constructor.
 - 🔗 InternationalizeControl(System.Windows.Forms.Control):System.Void (1)
 - ▶ **1 Message for InternationalizeControl(System.Windows.Forms.Control):System.Void**
 - Message Certainty Resolution
 - ▶ **Warning** 95 Parameter 'c' of GenericI18N.InternationalizeControl(Control):Void is never used. Remove the parameter or use it in the method body.
 - 🔗 InternationalizeControl(System.Windows.Forms.MenuItem):System.Void (1)
 - ▶ **1 Message for InternationalizeControl(System.Windows.Forms.MenuItem):System.Void**
 - Message Certainty Resolution
 - ▶ **Warning** 95 Parameter 'mi' of GenericI18N.InternationalizeControl(MenuItem):Void is never used. Remove the parameter or use it in the method body.
- I18N (17)
 - 🔗 .ctor() (1)
 - ▶ **1 Message for .ctor()**
 - Message Certainty Resolution
 - ▶ **Warning** 90 I18N.I18N() initializes field i18n of type DisksDB.Language.I18N to null. Remove this initialization as it will be done automatically by the runtime.
 - 🔗 GetActiveLanguage():System.Int32 (3)
 - ▶ **3 Messages for GetActiveLanguage():System.Int32**
 - Message Certainty Resolution
 - ▶ **Warning** 75 'I18N.GetActiveLanguage():Int32' appears to have no upstream public or protected callers.
 - ▶ **CriticalError** 95 Modify 'I18N.GetActiveLanguage():Int32' to catch a more specific exception than 'System.Exception' or rethrow the exception.
 - ▶ **Warning** 95 The 'this' parameter (or 'Me' in VB) of I18N.GetActiveLanguage():Int32 is never used. Mark the member as static (or Shared in VB) or use 'this'/'Me' in its method body or at least one property accessor, if appropriate.
 - 🔗 GetData():DisksDB.Language.DataSetLanguage (1)
 - ▶ **1 Message for GetData():DisksDB.Language.DataSetLanguage**
 - Message Certainty Resolution
 - ▶ **CriticalError** 95 Modify 'I18N.GetData():DataSetLanguage' to catch a more specific exception than 'System.Exception' or rethrow the exception.
 - 🔗 GetText(System.String, System.String):System.String (2)
 - ▶ **2 Messages for GetText(System.String, System.String):System.String**
 - Message Certainty Resolution
 - ▶ **Warning** 95 The 'this' parameter (or 'Me' in VB) of I18N.GetText(String, String):String is never used. Mark the member as static (or Shared in VB) or use 'this'/'Me' in its method body or at least one property accessor, if appropriate.
 - ▶ **Warning** 95 Parameter 'key' of I18N.GetText(String, String):String is never used. Remove the parameter or use it in the method body.
 - 🔗 hasLanguages (1)
 - ▶ **1 Message for hasLanguages**
 - Message Certainty Resolution
 - ▶ **Warning** 75 It appears that field 'I18N.hasLanguages' is never used or is only ever assigned to. Use this field or remove it.
 - ★ Languages (1)
 - ▶ **1 Message for Languages**
 - Message Certainty Resolution
 - ▶ **Warning** 50 Change I18N.Languages:Language[] to return a collection or make it a method.
 - 🔗 MessageShow (System.Windows.Forms.IWin32Window, System.String, System.String, System.Windows.Forms.MessageBoxButtons, System.Windows.Forms.MessageBoxIcon):System.Windows.Forms.DialogResult (2)
 - ▶ **2 Messages for MessageShow (System.Windows.Forms.IWin32Window, System.String, System.String, System.Windows.Forms.MessageBoxButtons, System.Windows.Forms.MessageBoxIcon):System.Windows.Forms.DialogResult**
 - Message Certainty Resolution
 - ▶ **Warning** 95 The 'this' parameter (or 'Me' in VB) of I18N.MessageShow(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon):DialogResult is never used. Mark the member as static (or Shared in VB) or use 'this'/'Me' in the method body or at least one property accessor, if appropriate.
 - ▶ **Error** 95 Change I18N.MessageShow(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon):DialogResult to call the MessageBox.Show overload that takes MessageBoxButtons, and make sure to set MessageBoxButtons.RightToLeft and MessageBoxIcon.RightToLeft if RightToLeft is set to RightToLeft.Yes on the control. If it is set to RightToLeft.No, check its parent until a value indicating Yes or No is found.
 - 🔗 SetDbFile(System.String):System.Void (1)
 - ▶ **1 Message for SetDbFile(System.String):System.Void**
 - Message Certainty Resolution
 - ▶ **Warning** 75 'I18N.SetDbFile(String):Void' appears to have no upstream public or protected callers.
 - 🔗 SetFile(System.String):System.Boolean (1)
 - ▶ **1 Message for SetFile(System.String):System.Boolean**
 - Message Certainty Resolution
 - ▶ **Warning** 75 'I18N.SetFile(String):Boolean' appears to have no upstream public or protected callers.

Warning 75

UpdateData(DisksDB.Language.DataSetLanguage):System.Void (1)

1 Message for UpdateData(DisksDB.Language.DataSetLanguage):System.Void

Message Level	Certainty	Resolution
CriticalError	95	Modify 'I18N.UpdateData(DataSetLanguage):Void' to catch a more specific exception than 'System.Exception' or rethrow the exception.

UpdateLanguage():System.Void (3)

3 Messages for UpdateLanguage():System.Void

Message Level	Certainty	Resolution
Warning	75	'I18N.UpdateLanguage():Void' appears to have no upstream public or protected callers.
Error	95	I18N.UpdateLanguage():Void passes a literal as parameter 1 of a call to ApplicationException.ApplicationException(String). Retrieve the following string argument from a resource table instead: 'No Languages found at all'
Error	95	I18N.UpdateLanguage():Void creates an exception of type 'System.ApplicationException', an exception type that is not sufficiently specific and should never be raised by user code. If this exception instance might be thrown, use a different exception type.

Language (1)

1 Message for Language

Message Level	Certainty	Resolution
Error	95	The type name 'Language' conflicts in whole or in part with the namespace name 'DisksDB.Language'. Change either name to eliminate the conflict.

\$(ProjectDir)/source/PocketPC/PocketDisksDB/bin/Release/DataBase.dll (59)

database.dll (59)

3 Messages for database.dll

Message Level	Certainty	Resolution
CriticalError	95	No valid permission requests were found for assembly 'DataBase'. You should always specify the minimum security permissions using SecurityAction.RequestMinimum.
CriticalError	95	Sign 'DataBase' with a strong name key.
Error	95	'DataBase' should be marked with CLSCompliantAttribute and its value should be true.

DisksDB.Pocket.DataBase (56)

Config (3)

1 Message for Config

Message Level	Certainty	Resolution
Error	95	The type name 'Config' conflicts in whole or in part with the namespace name 'DisksDB.Config'. Change either name to eliminate the conflict.

.ctor() (1)

1 Message for .ctor()

Message Level	Certainty	Resolution
Warning	95	Replace the call to String.Equals("") in 'Config.Config()' with a call to String.IsNullOrEmpty. Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\Config.cs (Line 31)

DataBaseFile (1)

1 Message for DataBaseFile

Message Level	Certainty	Resolution
Error	80	The compound word 'DataBase' in property 'DataBaseFile' exists as a discrete term. If your usage is intended to be single word, case it as 'Database'. Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\Config.cs (Line 56)

DataBase (53)

3 Messages for DataBase

Message Level	Certainty	Resolution
Error	80	The compound word 'DataBase' in class 'DataBase' exists as a discrete term. If your usage is intended to be single word, case it as 'Database'.
Error	95	The type name 'DataBase' conflicts in whole or in part with the namespace name 'DisksDB.Pocket.DataBase'. Change either name to eliminate the conflict
CriticalError	95	Implement IDisposable on 'DisksDB.Pocket.DataBase.DataBase' as it instantiates members of the following IDisposable types: System.Data.SqlServerCe.SqlCeConnection

.cctor() (1)

1 Message for .cctor()

Message Level	Certainty	Resolution
Warning	90	DataBase.DataBase() initializes field instance of type DisksDB.Pocket.DataBase.DataBase to null. Remove this initialization as it will be done automatically by runtime. Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 1032)

BuildParameters(System.Data.SqlServerCe.SqlCeCommand,System.String[],System.Object[]):System.Void (2)

2 Messages for BuildParameters(System.Data.SqlServerCe.SqlCeCommand,System.String[],System.Object[]):System.Void

Message Level	Certainty	Resolution
Warning	95	The 'this' parameter (or 'Me' in VB) of DataBase.BuildParameters(SqlCeCommand, String[], Object[]):Void is never used. Mark the member as static (or Sha or use 'this!'/'Me' in the method body or at least one property accessor, if appropriate). Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 156)
Warning	95	Parameter 'parameters' of DataBase.BuildParameters(SqlCeCommand, String[], Object[]):Void is never used. Remove the parameter or use it in the method body. Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 156)

CheckDataBaseId(System.String):System.Void (2)

2 Messages for CheckDataBaseId(System.String):System.Void

Message Level	Certainty	Resolution
Error	95	DataBase.CheckDataBaseId(String):Void passes a literal as parameter 1 of a call to ApplicationException.ApplicationException(String). Retrieve the following argument from a resource table instead: 'Invalid Sync server response' Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 421)
Error	95	DataBase.CheckDataBaseId(String):Void passes a literal as parameter 1 of a call to ApplicationException.ApplicationException(String). Retrieve the following argument from a resource table instead: 'Server and PDA databases are not the same' Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 434)
Error	95	DataBase.CheckDataBaseId(String):Void creates an exception of type 'System.ApplicationException', an exception type that is not sufficiently specific and should be raised by user code. If this exception instance might be thrown, use a different exception type. Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 434)
Error	95	DataBase.CheckDataBaseId(String):Void creates an exception of type 'System.ApplicationException', an exception type that is not sufficiently specific and should be raised by user code. If this exception instance might be thrown, use a different exception type. Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 421)

CloseSqlConnection(System.Data.SqlServerCe.SqlCeConnection):System.Void (1)

1 Message for CloseSqlConnection(System.Data.SqlServerCe.SqlCeConnection):System.Void

Message Level	Certainty	Resolution
Warning	95	Parameter 'sqlCon' of DataBase.CloseSqlConnection(SqlCeConnection):Void is never used. Remove the parameter or use it in the method body. Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 85)

- ExecNonQuery(System.String,System.Boolean):System.Void (1)

1 Message for ExecNonQuery(System.String,System.Boolean):System.Void

Message Certainty Resolution

Level

 - Error** 95 Change DataBase.ExecuteNonQuery(String, Boolean):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and make sure to set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLeft.In check its parent until a value indicating Yes or No is found.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 122)
- ExecNonQuery(System.String,System.String[],System.Object[]):System.Void (1)

1 Message for ExecNonQuery(System.String,System.String[],System.Object[]):System.Void

Message Certainty Resolution

Level

 - Error** 95 Change DataBase.ExecuteNonQuery(String, String[], Object[]):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and make sure to set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLeft.In check its parent until a value indicating Yes or No is found.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 180)
- ExecScalar(System.String):System.Object (1)

1 Message for ExecScalar(System.String):System.Object

Message Certainty Resolution

Level

 - Error** 95 Change DataBase.ExecScalar(String):Object to call the MessageBox.Show overload that specifies MessageBoxOptions, and make sure to set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLeft.In check its parent until a value indicating Yes or No is found.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 145)
- FillCategories(System.Int64,System.Windows.Forms.ListView+ListViewItemCollection):System.Void (4)

4 Messages for FillCategories(System.Int64,System.Windows.Forms.ListView+ListViewItemCollection):System.Void

Message Level Certainty Resolution

 - Error** 80 DataBase.FillCategories(Int64, ListViewItemCollection):Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThrow and defines a catch(Exception) handler that is not matched with a generic catch clause (catch{} in C#, catch(System::Object*){} in mc++) for non-CLSCompliant exceptions. Either mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = true), eliminate the catch(Exception) clause by catching a specific set of Exceptions or add a generic catch handler that duplicates the code contained within the catch(Exception) handler.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 882)
 - CriticalError** 95 Modify 'DataBase.FillCategories(Int64, ListViewItemCollection):Void' to catch a more specific exception than 'System.Exception' or rethrow the exception

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 882)
 - Error** 95 Change DataBase.FillCategories(Int64, ListViewItemCollection):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and make set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLeft.Inherit, check its parent until a value indicating Yes or No is found.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 884)
 - Error** 95 Validate parameter 'items' passed to externally visible method DataBase.FillCategories(Int64, ListViewItemCollection):Void.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 856)
- FillDataTable(System.String,System.Data.DataTable):System.Void (1)

1 Message for FillDataTable(System.String,System.Data.DataTable):System.Void

Message Level Certainty Resolution

 - Error** 95 Change DataBase.FillDataTable(String, DataTable):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and make sure to set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLeft.In check its parent until a value indicating Yes or No is found.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 211)
- FillDisks(System.Int64,System.Windows.Forms.ListView+ListViewItemCollection):System.Void (4)

4 Messages for FillDisks(System.Int64,System.Windows.Forms.ListView+ListViewItemCollection):System.Void

Message Level Certainty Resolution

 - Error** 80 DataBase.FillDisks(Int64, ListViewItemCollection):Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThrows = t defines a catch(Exception) handler that is not matched with a generic catch clause (catch{} in C#, catch(System::Object*){} in mc++) for non-CLSCompliant exceptions. Either mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = true), eliminate the catch(Exception) clause by catching a s of Exceptions or add a generic catch handler that duplicates the code contained within the catch(Exception) handler.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 910)
 - CriticalError** 95 Modify 'DataBase.FillDisks(Int64, ListViewItemCollection):Void' to catch a more specific exception than 'System.Exception' or rethrow the exception.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 910)
 - Error** 95 Change DataBase.FillDisks(Int64, ListViewItemCollection):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and make sure MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLe check its parent until a value indicating Yes or No is found.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 912)
 - Error** 95 Validate parameter 'items' passed to externally visible method DataBase.FillDisks(Int64, ListViewItemCollection):Void.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 895)
- FillFilesByDisk(System.Int64,System.Windows.Forms.ListView+ListViewItemCollection):System.Void (4)

4 Messages for FillFilesByDisk(System.Int64,System.Windows.Forms.ListView+ListViewItemCollection):System.Void

Message Level Certainty Resolution

 - Error** 80 DataBase.FillFilesByDisk(Int64, ListViewItemCollection):Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThro and defines a catch(Exception) handler that is not matched with a generic catch clause (catch{} in C#, catch(System::Object*){} in mc++) for non-CLSCompliant exceptions. Either mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = true), eliminate the catch(Exception) clause b catching a specific set of Exceptions or add a generic catch handler that duplicates the code contained within the catch(Exception) handler.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 968)
 - CriticalError** 95 Modify 'DataBase.FillFilesByDisk(Int64, ListViewItemCollection):Void' to catch a more specific exception than 'System.Exception' or rethrow the exceptio

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 968)
 - Error** 95 Change DataBase.FillFilesByDisk(Int64, ListViewItemCollection):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and mak set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLef.Inherit, check its parent until a value indicating Yes or No is found.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 970)
 - Error** 95 Validate parameter 'items' passed to externally visible method DataBase.FillFilesByDisk(Int64, ListViewItemCollection):Void.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 953)
- FillFilesByFile(System.Int64,System.Windows.Forms.ListView+ListViewItemCollection):System.Void (4)

4 Messages for FillFilesByFile(System.Int64,System.Windows.Forms.ListView+ListViewItemCollection):System.Void

Message Level Certainty Resolution

 - Error** 80 DataBase.FillFilesByFile(Int64, ListViewItemCollection):Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThrow and defines a catch(Exception) handler that is not matched with a generic catch clause (catch{} in C#, catch(System::Object*){} in mc++) for non-CLSCompliant exceptions. Either mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = true), eliminate the catch(Exception) clause b catching a specific set of Exceptions or add a generic catch handler that duplicates the code contained within the catch(Exception) handler.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 996)
 - CriticalError** 95 Modify 'DataBase.FillFilesByFile(Int64, ListViewItemCollection):Void' to catch a more specific exception than 'System.Exception' or rethrow the exception

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 996)
 - Error** 95 Change DataBase.FillFilesByFile(Int64, ListViewItemCollection):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and make set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLef.Inherit, check its parent until a value indicating Yes or No is found.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 998)

- ▶ **Error** 95 Validate parameter 'items' passed to externally visible method DataBase.FillFilesByFile(Int64, ListViewItemCollection):Void.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 981)
- FindRootCategories(System.Windows.Forms.ListView+ListViewItemCollection):System.Void (1)
 - ▶ **1 Message for FillRootCategories(System.Windows.Forms.ListView+ListViewItemCollection):System.Void**
 - Message Level Certainty Resolution
 - ▶ **Error** 95 Validate parameter 'items' passed to externally visible method DataBase.FillRootCategories(ListViewItemCollection):Void.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 843)
- FindFiles(System.String,System.Int64,System.Int64,System.Boolean,System.Boolean):System.Data.DataTable (1)
 - ▶ **1 Message for FindFiles(System.String,System.Int64,System.Int64,System.Boolean,System.Boolean):System.Data.DataTable**
 - Message Level Certainty Resolution
 - ▶ **Error** 95 Validate parameter 'fileName' passed to externally visible method DataBase.FindFiles(String, Int64, Int64, Boolean, Boolean):DataTable.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 1022)
- FormatSize(System.Int64):System.String (2)
 - ▶ **2 Messages for FormatSize(System.Int64):System.String**
 - Message Level Certainty Resolution
 - ▶ **Warning** 95 The 'this' parameter (or 'Me' in VB) of DataBase.FormatSize(Int64):String is never used. Mark the member as static (or Shared in VB) or use 'this/Me' in the body or at least one property accessor, if appropriate.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 922)
 - ▶ **Error** 95 DataBase.FormatSize(Int64):String makes a call to System.Double.ToString(System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Double.ToString(System.String,System.IFormatProvider).
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 938)
 - ▶ **Error** 95 DataBase.FormatSize(Int64):String makes a call to System.Double.ToString(System.String,System.IFormatProvider) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Double.ToString(System.String,System.IFormatProvider).
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 944)
 - ▶ **Error** 95 DataBase.FormatSize(Int64):String makes a call to System.Double.ToString(System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Double.ToString(System.String,System.IFormatProvider).
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 926)
 - ▶ **Error** 95 DataBase.FormatSize(Int64):String makes a call to System.Double.ToString(System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Double.ToString(System.String,System.IFormatProvider).
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 932)
- SendMessage(System.String,System.Net.Sockets.TcpClient):System.Void (1)
 - ▶ **1 Message for SendMessage(System.String,System.Net.Sockets.TcpClient):System.Void**
 - Message Level Certainty Resolution
 - ▶ **Warning** 95 The 'this' parameter (or 'Me' in VB) of DataBase.SendMessage(String, TcpClient):Void is never used. Mark the member as static (or Shared in VB) or use 'this' in the method body or at least one property accessor, if appropriate.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 406)
- Synchronize(DisksDB.Pocket.DataBase.ISyncProgress):System.Void (2)
 - ▶ **2 Messages for Synchronize(DisksDB.Pocket.DataBase.ISyncProgress):System.Void**
 - Message Level Certainty Resolution
 - ▶ **Error** 80 DataBase.Synchronize(ISyncProgress):Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThrows = true) and does not have a catch(Exception) handler that is not matched with a generic catch clause (catch{} in C#, catch(System::Object*){} in mc++) for non-CLSCompliant exceptions. Either mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = true), eliminate the catch(Exception) clause by catching a specific set of Exceptions or add a generic catch handler that duplicates the code contained within the catch(Exception) handler.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 835)
 - ▶ **Error** 80 DataBase.Synchronize(ISyncProgress):Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThrows = true) and does not have a catch(Exception) handler that is not matched with a generic catch clause (catch{} in C#, catch(System::Object*){} in mc++) for non-CLSCompliant exceptions. Either mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = true), eliminate the catch(Exception) clause by catching a specific set of Exceptions or add a generic catch handler that duplicates the code contained within the catch(Exception) handler.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 808)
 - ▶ **CriticalError** 95 Modify 'DataBase.Synchronize(ISyncProgress):Void' to catch a more specific exception than 'System.Exception' or rethrow the exception.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 835)
- UpdateBoxes(System.String):System.Void (2)
 - ▶ **2 Messages for UpdateBoxes(System.String):System.Void**
 - Message Level Certainty Resolution
 - ▶ **Error** 95 DataBase.UpdateBoxes(String):Void makes a call to System.Int64.Parse(System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int64.Parse(System.String,System.IFormatProvider).
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 702)
 - ▶ **Error** 95 DataBase.UpdateBoxes(String):Void makes a call to System.Int64.Parse(System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int64.Parse(System.String,System.IFormatProvider).
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 695)
 - ▶ **Warning** 95 Replace the call to String.Equals(String.Empty) in 'DataBase.UpdateBoxes(String):Void' with a call to String.IsNullOrEmpty.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 702)
 - ▶ **Warning** 95 Replace the call to String.Equals(String.Empty) in 'DataBase.UpdateBoxes(String):Void' with a call to String.IsNullOrEmpty.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 695)
- UpdateCategories(System.String):System.Void (2)
 - ▶ **2 Messages for UpdateCategories(System.String):System.Void**
 - Message Level Certainty Resolution
 - ▶ **Error** 95 DataBase.UpdateCategories(String):Void makes a call to System.Int64.Parse(System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int64.Parse(System.String,System.IFormatProvider).
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 469)
 - ▶ **Error** 95 DataBase.UpdateCategories(String):Void makes a call to System.Int64.Parse(System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int64.Parse(System.String,System.IFormatProvider).
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 462)
 - ▶ **Warning** 95 Replace the call to String.Equals(String.Empty) in 'DataBase.UpdateCategories(String):Void' with a call to String.IsNullOrEmpty.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 469)
 - ▶ **Warning** 95 Replace the call to String.Equals(String.Empty) in 'DataBase.UpdateCategories(String):Void' with a call to String.IsNullOrEmpty.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 462)
- UpdateDisks(System.String,System.Net.Sockets.TcpClient,System.IO.StreamReader):System.Void (4)
 - ▶ **4 Messages for UpdateDisks(System.String,System.Net.Sockets.TcpClient,System.IO.StreamReader):System.Void**
 - Message Level Certainty Resolution
 - ▶ **Error** 80 DataBase.UpdateDisks(String, TcpClient, StreamReader):Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThrows = true) and does not have a catch(Exception) handler that is not matched with a generic catch clause (catch{} in C#, catch(System::Object*){} in mc++) for non-CLSCompliant exceptions. Either mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = true), eliminate the catch(Exception) clause by catching a specific set of Exceptions or add a generic catch handler that duplicates the code contained within the catch(Exception) handler.
 - Source:** C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 586)
 - ▶ **Error** 80 DataBase.UpdateDisks(String, TcpClient, StreamReader):Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThrows = true) and does not have a catch(Exception) handler that is not matched with a generic catch clause (catch{} in C#, catch(System::Object*){} in mc++) for non-CLSCompliant exceptions. Either mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = true), eliminate the catch(Exception) clause by catching a specific set of Exceptions or add a generic catch handler that duplicates the code contained within the catch(Exception) handler.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 571)

Error 80 DataBase.UpdateDisks(String, TcpClient, StreamReader):Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThrow true) and defines a catch(Exception) handler that is not matched with a generic catch clause (catch{ } in C#, catch(System::Object*){ } in mc++) for n CLSCompliant exceptions. Either mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = true), eliminate the catch(Exception) clause b catching a specific set of Exceptions or add a generic catch handler that duplicates the code contained within the catch(Exception) handler.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 560)

CriticalError 95 Modify 'DataBase.UpdateDisks(String, TcpClient, StreamReader):Void' to catch a more specific exception than 'System.Exception' or rethrow the excepti

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 586)

CriticalError 95 Modify 'DataBase.UpdateDisks(String, TcpClient, StreamReader):Void' to catch a more specific exception than 'System.Exception' or rethrow the excepti

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 571)

Error 95 DataBase.UpdateDisks(String, TcpClient, StreamReader):Void makes a call to System.Int64.Parse(System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int64.Parse(System.String, System.IFormatProvider).

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 545)

Error 95 DataBase.UpdateDisks(String, TcpClient, StreamReader):Void makes a call to System.Int64.Parse(System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int64.Parse(System.String, System.IFormatProvider).

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 521)

Warning 95 Replace the call to String.Equals(String.Empty) in 'DataBase.UpdateDisks(String, TcpClient, StreamReader):Void' with a call to String.IsNullOrEmpty.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 545)

Warning 95 Replace the call to String.Equals(String.Empty) in 'DataBase.UpdateDisks(String, TcpClient, StreamReader):Void' with a call to String.IsNullOrEmpty.

Source: C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 521)

UpdateFiles (System.String, System.Net.Sockets.TcpClient, System.IO.StreamReader, System.Data.SqlClient.SqlCeConnection, System.Data.SqlClient.SqlCeTransaction):S

9 Messages for UpdateFiles (System.String, System.Net.Sockets.TcpClient, System.IO.StreamReader, System.Data.SqlClient.SqlCeConnection, System.Data.SqlClient.SqlServerCe.Sq

Message Level	Certainty	Resolution
Error 80		DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThrows = true) and defines a catch(Exception) handler that is not matched with a generic catch clause (catch{ catch(System::Object*){ } in mc++) for non-CLSCompliant exceptions. Either mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = eliminate the catch(Exception) clause by catching a specific set of Exceptions or add a generic catch handler that duplicates the code contained within th (Exception) handler.
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 663)
CriticalError 95		Modify 'DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void' to catch a more specific exception than 'System.Exception' or rethrow the exception.
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 663)
Warning 95		The 'this' parameter (or 'Me' in VB) of DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void is never used. member as static (or Shared in VB) or use 'this/Me' in the method body or at least one property accessor, if appropriate.
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 609)
Warning 95		Parameter 'sr' of DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void is never used. Remove the paramete in the method body.
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 609)
Warning 95		Parameter 'tcpClient' of DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void is never used. Remove the pa or use it in the method body.
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 609)
Error 95		DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void makes a call to System.Int32.Parse(System.String) th not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int32.Parse(System.String, System.IFormatProvider).
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 648)
Error 95		DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void makes a call to System.Int64.Parse(System.String) th not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int64.Parse(System.String, System.IFormatProvider).
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 647)
Error 95		DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void makes a call to System.Int64.Parse(System.String) th not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int64.Parse(System.String, System.IFormatProvider).
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 649)
Error 95		DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void makes a call to System.Int64.Parse(System.String) th not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int64.Parse(System.String, System.IFormatProvider).
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 611)
Error 95		DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void makes a call to System.Int64.Parse(System.String) th not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int64.Parse(System.String, System.IFormatProvider).
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 646)
Error 95		Change DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void to call the MessageBox.Show overload that sp MessageBoxOptions, and make sure to set MessageBoxOptions.RightAlign and MessageBoxOptions.RightToLeft if RightToLeft is set to RightToLeft.Yes on parent control. If it is set to RightToLeft.Inherit, check its parent until a value indicating Yes or No is found.
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 665)
Warning 95		Replace the call to String.Equals(String.Empty) in 'DataBase.UpdateFiles(String, TcpClient, StreamReader, SqlConnection, SqlCeTransaction):Void' wi String.IsNullOrEmpty.
Source:		C:\Users\JL\Desktop\source\PocketPC\DataBase\DataBase.cs (Line 649)

\$(ProjectDir)\source\PocketPC\PocketDisksDB\bin\Release\DisksDB.exe (21)

disksdb.exe (21)

4 Messages for disksdb.exe

Message Level	Certainty	Resolution
CriticalError 95		No valid permission requests were found for assembly 'DisksDB'. You should always specify the minimum security permissions using SecurityAction.RequestMinimum.
CriticalError 95		Sign 'DisksDB' with a strong name key.
Error 95		'DisksDB' should be marked with CLSCompliantAttribute and its value should be true.
Error 95		Add STAThreadAttribute to assembly 'DisksDB'.

{ DisksDB.Pocket

(2) Startup (2)

2 Messages for Startup

Message Level	Certainty	Resolution
Error 80		The compound word 'Startup' in class 'Startup' exists as a discrete term. If your usage is intended to be single word, case it as 'Startup'.
Error 90		Remove the public constructors from 'Startup'.

{ DisksDB.Pocket.Properties

(1) Resources (1)

1 Message for Resources

Message Level	Certainty	Resolution
Warning 75		'DisksDB.Pocket.Properties.Resources' is an internal class that is apparently never instantiated. If so, remove the code from the assembly. If this class is inten contain only static methods, consider adding a private constructor to prevent the compiler from generating a default constructor.

{ DisksDB.Pocket.UserInterface

(13) FormMain (12)

CleanClick(System.Object, System.EventArgs):System.Void (3)

3 Messages for CleanClick(System.Object, System.EventArgs):System.Void

Message Level	Certainty	Resolution
---------------	-----------	------------

- ▶ **Error** 95 FormMain.CleanClick(Object, EventArgs):Void passes a literal as parameter 1 of a call to MessageBox.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton):DialogResult. Retrieve the following string argument from a resource table instead: 'Clean database ?'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 71)
- ▶ **Error** 95 FormMain.CleanClick(Object, EventArgs):Void passes a literal as parameter 1 of a call to MessageBox.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton):DialogResult. Retrieve the following string argument from a resource table instead: 'DataBase has been resetted'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 77)
- ▶ **Error** 95 FormMain.CleanClick(Object, EventArgs):Void passes a literal as parameter 2 of a call to MessageBox.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton):DialogResult. Retrieve the following string argument from a resource table instead: 'Clean database'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 71)
- ▶ **Error** 95 FormMain.CleanClick(Object, EventArgs):Void passes a literal as parameter 2 of a call to MessageBox.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton):DialogResult. Retrieve the following string argument from a resource table instead: 'DataBase'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 77)
- ▶ **Error** 85 Correct the spelling of the unrecognized token 'resetted' in the literal 'DataBase has been resetted'.
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 77)
- ▶ **Error** 95 Change FormMain.CleanClick(Object, EventArgs):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and make sure to set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLeft.In check its parent until a value indicating Yes or No is found.
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 77)
- ▶ **Error** 95 Change FormMain.CleanClick(Object, EventArgs):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and make sure to set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLeft.In check its parent until a value indicating Yes or No is found.
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 71)

InitializeComponent():System.Void (3)
 3 Messages for InitializeComponent():System.Void
 Message Certainty Resolution

- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to ColumnHeader.set_Text(String):Void. Retrieve the following string argument resource table instead: 'Date'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 180)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to ColumnHeader.set_Text(String):Void. Retrieve the following string argument resource table instead: 'Name'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 175)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to ColumnHeader.set_Text(String):Void. Retrieve the following string argument resource table instead: 'Size'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 185)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from a table instead: 'Catalog'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 130)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from a table instead: 'DisksDB'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 332)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from a table instead: 'File name'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 211)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from a table instead: 'Max size'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 248)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from a table instead: 'Min size'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 219)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from a table instead: 'Search'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 304)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from a table instead: 'Search'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 204)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from a table instead: 'Search Results'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 313)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from a table instead: 'Use max'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 286)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from a table instead: 'Use min'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 277)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to MenuItem.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Clean Data'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 89)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to MenuItem.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Exit'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 84)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to MenuItem.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Synchronize'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 94)
- ▶ **Error** 95 FormMain.InitializeComponent():Void passes a literal as parameter 1 of a call to MenuItem.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Tools'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.Designer.cs (Line 80)

OnSyncDone(System.Boolean):System.Void (2)
 2 Messages for OnSyncDone(System.Boolean):System.Void
 Message Certainty Resolution

- ▶ **Error** 95 FormMain.OnSyncDone(Boolean):Void passes a literal as parameter 1 of a call to MessageBox.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton):DialogResult. Retrieve the following string argument from a resource table instead: 'Synchronization Done'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 301)
- ▶ **Error** 95 FormMain.OnSyncDone(Boolean):Void passes a literal as parameter 1 of a call to MessageBox.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton):DialogResult. Retrieve the following string argument from a resource table instead: 'Synchronization failed. Make sure that: 1. Di running on your desktop computer 2. DisksDB is configured to allow synchronization 3. PocketPC device is connected to your PC by cradle. 4. PocketPC device disconnected during synchronization process'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 305)
- ▶ **Error** 95 FormMain.OnSyncDone(Boolean):Void passes a literal as parameter 2 of a call to MessageBox.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton):DialogResult. Retrieve the following string argument from a resource table instead: 'Synchronization'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 305)
- ▶ **Error** 95 FormMain.OnSyncDone(Boolean):Void passes a literal as parameter 2 of a call to MessageBox.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton):DialogResult. Retrieve the following string argument from a resource table instead: 'Synchronization'
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 301)
- ▶ **Error** 95 Change FormMain.OnSyncDone(Boolean):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and make sure to set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLeft.In check its parent until a value indicating Yes or No is found.
Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 305)
- ▶ **Error** 95 Change FormMain.OnSyncDone(Boolean):Void to call the MessageBox.Show overload that specifies MessageBoxOptions, and make sure to set MessageBoxOptions.RightAlign and MessageBoxOptions.RtlReading if RightToLeft is set to RightToLeft.Yes on the parent control. If it is set to RightToLeft.In check its parent until a value indicating Yes or No is found.

Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 301)

OptionsClick(System.Object, System.EventArgs):System.Void (1)

1 Message for OptionsClick(System.Object, System.EventArgs):System.Void

Message Level	Certainty	Resolution
Warning	75	'FormMain.OptionsClick(Object, EventArgs):Void' appears to have no upstream public or protected callers.

Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 99)

SynchThreadFunc():System.Void (2)

2 Messages for SynchThreadFunc():System.Void

Message Level	Certainty	Resolution
Error	80	FormMain.SynchThreadFunc():Void is defined in an assembly that is not marked RuntimeCompatibility(WrapNonExceptionThrows = true) and defines a (Exception) handler that is not matched with a generic catch clause (catch{} in C#, catch(System::Object*){} in mc++) for non- CLSCompliant exceptions. To mark the assembly as RuntimeCompatibility(WrapNonExceptionThrows = true), eliminate the catch(Exception) clause by catching a specific set of Exceptions and add a generic catch handler that duplicates the code contained within the catch(Exception) handler.
CriticalError	95	Modify 'FormMain.SynchThreadFunc():Void' to catch a more specific exception than 'System.Exception' or rethrow the exception.

Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 263)

UpdateSyncText(System.String):System.Void (1)

1 Message for UpdateSyncText(System.String):System.Void

Message Level	Certainty	Resolution
Error	95	FormMain.UpdateSyncText(String):Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Synchronizing: ___'

Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormMain.cs (Line 275)

FormOptions (1)

InitializeComponent():System.Void (1)

1 Message for InitializeComponent():System.Void

Message Level	Certainty	Resolution
Error	95	FormOptions.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Cancel'
Error	95	FormOptions.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Password'
Error	95	FormOptions.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Pocket DisksDB'
Error	95	FormOptions.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Save password'
Error	95	FormOptions.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Server'
Error	95	FormOptions.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Synchronization options'
Error	95	FormOptions.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'Use ActiveSync'
Error	95	FormOptions.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'User'
Error	95	FormOptions.InitializeComponent():Void passes a literal as parameter 1 of a call to Control.set_Text(String):Void. Retrieve the following string argument from resource table instead: 'label1'

Source: C:\Users\JL\Desktop\source\PocketPC\PocketDisksDB\UserInterface\FormOptions.Designer.cs (Line 146)

DisksDB.Pocket.Utilis (1)

MyResources (1)

1 Message for MyResources

Message Level	Certainty	Resolution
Error	90	Remove the public constructors from 'MyResources'.