



Programų sistemų analizė

## Programinės įrangos projektavimas

Funkcinis ir objektiškai orientuotas projektavimo  
metodai

Lina Vasiliauskienė

Grafinių sistemų katedra

Vilniaus Gedimino Technikos Universitetas  
2009-2010

### *Turinys*

- Funkcinis projektavimas
  - Funkcinio projektavimo metodika ir procesas
  - Duomenų srautų diagramos
  - Struktūrinės diagramos
  - Duomenų žodyno sudarymas
- Objektiškai orientuotas projektavimas
  - Objektiškai orientuoto projektavimo metodika ir procesas
  - Objektiškai orientuoto projektavimo pavyzdys
- Funkcinio ir objektiškai orientuoto projektavimo metodų palyginimas

L. Vasiliauskienė. P| projektavimas

## ***Funkcinio projektavimo metodika***

- Funkcinio projektavimo metodika pagrįsta programos dekompozicija į rinkinį tarpusavyje sąveikaujančių funkcijų, įskaitant sistemos būseną, kurios informacija jos dalinasi (vidinė funkcijų būsenos informacija yra palaikoma tik jų vykdymo metu)
- Funkcinio projektavimo atveju algoritmas saugomas pačioje funkcijoje, o sistemos būsenos informacija žinoma visiems programos komponentams

L. Vasiliauskienė. PJ projektavimas

## ***Funkcinio projektavimo taikymas***

- Funkcinis projektavimo metodas yra efektyviausias projektuojant sistemą, kur saugomos informacijos kiekis apie jos būseną minimalus ir yra aiškus informacijos dalijimasis tarp jos komponentų
- Kai kuriose situacijose tokia architektūra kelia problemų, nes galimi atvejai, kuomet kuri nors funkcija pakeičia sistemos būseną taip, kaip kitos funkcijos to nesitiki

L. Vasiliauskienė. PJ projektavimas

## *Funkcinio projektavimo procesas*

- **Duomenų srautų projektavimas.** Jis leidžia atskleisti, kaip vyksta duomenų mainai tarp sistemos funkcijų, ir kaip jos įėjimo duomenis transformuoja į išėjimo duomenis
- **Struktūrinė dekompozicija.** Pavaizduoja sistemos funkcijų hierarchiją, pradedant nuo aukščiausio lygio funkcijų ir baigiant žemiausio lygio funkcijomis
- **Detalus projektavimas.** Jo metu išsamiai aprašomos projektavimo metu išskirtos sistemos dalys ir jų tarpusavio sąsajos

L. Vasiliauskienė. Pj projektavimas

## *Duomenų srautų projektavimas*

- Duomenų srautų projektavimo metu yra modeliuojamos funkcinės transformacijos, kurios gautus įėjimo duomenis transformuoja į pageidaujamus išėjimo duomenis





L. Vasiliauskienė. Pj projektavimas

## Duomenų srautų diagramos

- Duomenų srautų projektavimo rezultatai yra atvaizduojami duomenų srautų diagramomis
- Duomenų srautų diagramos yra patogus ir intuityviai suvokiamas sistemos atvaizdavimo būdas

L. Vasiliauskienė. Pj projektavimas

## Duomenų srautų diagramos žymėjimai

Diagramos elementas	Elemento aprašymas
	Funkcija, kuri paverčia įėjimo duomenis į išėjimo
	Duomenų saugykla
	Vartotojo sąveika su sistema, kurios metu pateikiami įėjimo duomenys arba gaunami išėjimo duomenys
	Duomenų srauto kryptis
“and”; “or”	Loginės išraiškos, kurios skirtos susieti duomenų srautus, kuomet daugiau negu vienas iš jų gali būti įėjimo arba išėjimo duomenimis

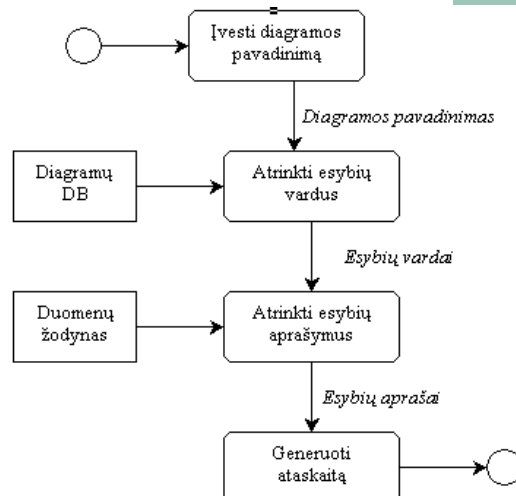
L. Vasiliauskienė. Pj projektavimas

## Pavyzdys

- Panagrinėkime ataskaitų generavimo programą:
  - Vartotojas programai nurodo diagramos pavadinimą, o ataskaitų generatorius suranda visas esybes, priklausančias nurodytai diagramai, ir jų aprašymus duomenų žodyne
  - Atrinkus visą reikiamą informaciją, sugeneruojama ataskaita

L. Vasiliauskienė. P] projektavimas

## Ataskaitų generatoriaus duomenų srautų diagrama



L. Vasiliauskienė. P] projektavimas

## ***Duomenų srautų diagramos rezultatas***

- Duomenų srautų diagramos parodo duomenų transformacijas, tačiau nesiūlo, kaip reikėtų jas realizuoti
- Jomis aprašyta sistema gali būti realizuota kaip viena programa, sudaryta iš funkcijų ir procedūrų, arba kaip visa eilė tarpusavyje sąveikaujančių programų

L. Vasiliauskienė. Pj projektavimas

## ***Funkcinis projektavimas - struktūrinė dekompozicija***

- Struktūrinė dekompozicija leidžia parodyti, kaip kuri nors viena funkcija yra realizuota kitų funkcijų, iš kurių ji yra sudaryta ir kurias iškviečia vykdymo metu






L. Vasiliauskienė. Pj projektavimas

## Struktūrinės diagramos

- Struktūrinės dekompozijos rezultatams atvaizduoti naudojamos struktūrinės diagramos, kuriose pavaizduota sistemą sudarančių funkcijų hierarchija, iš kurios matyti, kaip viena funkcija iškviečia kitas

L. Vasiliauskienė. Pj projektavimas

## Struktūrinės diagramos žymėjimai

Diagramos elementas	Elemento aprašymas
	Sistemos funkcija
	Funkcijų susiejimo ryšys
	Įėjimo ir išėjimo duomenys (parametrai arba bendri kintamieji)
	Duomenų saugykla
	Vartotojo įvedami duomenys

L. Vasiliauskienė. Pj projektavimas

## ***Struktūrinės diagramos sudarymas***

- Paprastai duomenų srautų diagrama yra paverčiama į struktūrinę diagramą
- Tai nėra automatinis procesas – jis reikalauja projektuotojo įžvalgumo ir kūrybiškumo

L. Vasiliauskienė. Pj projektavimas

## ***Struktūrinės diagramos sudarymo taisyklės (1)***

- Dauguma verslo sistemų, kurioms funkcinis projektavimas yra pats tinkamiausias, yra sudarytos iš trijų dalių:
  - duomenų įvedimo, apimančio jų kontrolę ir tikrinimą
  - duomenų apdorojimo
  - duomenų išvedimo (dažniausiai ataskaitos pavidalu) arba jų išsaugojimo į failą

L. Vasiliauskienė. Pj projektavimas



## ***Struktūrinės diagramos sudarymo taisyklės (2)***

- Jeigu reikalinga duomenų kontrolė, ją realizuojančios funkcijos priskiriamos duomenų įvedimo funkcijoms
- Atitinkamai ataskaitų formavimo, saugojimo į failus ir pan. funkcijos turi būti priskirtos duomenų išvedimo funkcijoms

L. Vasiliauskienė. PJ projektavimas

## ***Struktūrinės diagramos sudarymo taisyklės (3)***

- Kiekviena funkcija turi atlikti tik vieną užduotį ir daugiau nieko
- Kiekvienas struktūrinės diagramos mazgas gali turėti ne daugiau 7 jam pavaldžių mazgų
- Jeigu mazgas teturi vieną pavaldų mazgą, tai reiškia, kad pastarasis negali būti atskira funkcija
- Jeigu mazgas turi labai daug jam pavaldžių mazgų, vadinasi, sistema išskaidyta per smulkiai

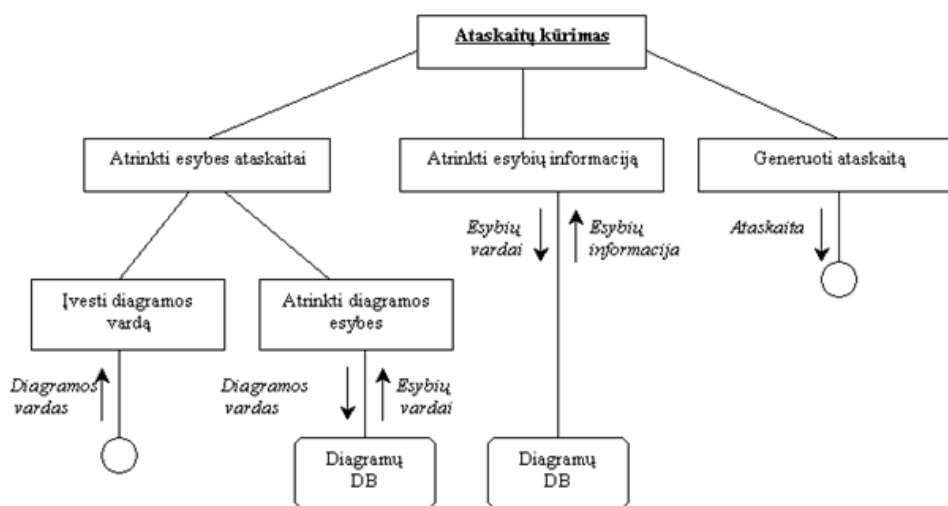
L. Vasiliauskienė. PJ projektavimas

## Duomenų srautų diagramos transformacija į struktūrinę diagramą

- Duomenų apdorojimo transformacijų identifikavimas
- Duomenų įvedimo transformacijų identifikavimas
- Duomenų išvedimo transformacijų identifikavimas

L. Vasiliauskienė. Pj projektavimas

## Ataskaitų generatoriaus struktūrinė diagrama



L. Vasiliauskienė. Pj projektavimas

## ***Funkcinis projektavimas - detalus projektavimas***

- Detalaus projektavimo metu aprašoma kiekviena sistemos funkcija: jos algoritmas, įėjimai ir išėjimai
- Detalus projektavimas padeda išryškinti duomenų srautų ir struktūrinių diagramų spragas, todėl jos šiame etape dažnai yra modifikuojamos

L. Vasiliauskienė. Pj projektavimas

## ***Funkcinis projektavimas - duomenų žodynas***

- Funkcijų aprašymai yra talpinami į duomenų žodyną, kuris leidžia išvengti to paties funkcijos vardo panaudojimo kelis kartus ir atskleisti skaitytojams, kaip projektuotojas įsivaizduoja būsimos sistemos struktūrą
- Duomenų žodynuose gali būti saugomi tiek trumpi, neformalūs funkcijų aprašymai, tiek ir detalus funkcijų apibūdinimas kokia nors projektavimo aprašymo kalba

L. Vasiliauskienė. Pj projektavimas

## Ataskaitų generatoriaus duomenų žodyno ištrauka

Esybė	Tipas	Aprašymas
Diagramos vardas	STRING	Diagramos vardas, jai priskirtas projektuotojo
Įvesti diagramos vardą	FUNCTION	<i>Įėjimas:</i> Diagramos vardas <i>Funkcija:</i> funkcija sąveikauja su vartotoju tam, kad jis galėtų įvesti diagramos, kurios esybių informacijos ataskaitą jis nori gauti, vardą <i>Išėjimas:</i> Diagramos vardas

L. Vasiliauskienė. PĮ projektavimas

## Objektiškai orientuotas projektavimas

- Objektiškai orientuota analizė (OOA), projektavimas (OOP) ir programavimas (OOPRG) yra tarpusavyje susijusios, bet ne tos pačios sąvokos
  - OOA metu sudaromas objektų, priklausančių nagrinėjamai dalykinei sričiai, modelis
  - OOP metu sudaromas objektiškai orientuotos sistemos (PĮ) modelis, pagal kurį bus realizuoti PĮ keliami reikalavimai
  - OOPRG metu, naudojant pasirinktą OO programavimo kalbą (C++, Java ar kt.), realizuojamas OOP metu sudarytas sistemos modelis

L. Vasiliauskienė. PĮ projektavimas

## ***OOP metodologija – objektai***

- Objektas yra realaus pasaulio esybės abstrakcija
- Objektai yra nepriklausomi vienetai, kurių būseną ir realizacijos detalės yra paslėptos nuo kitų objektų (inkapsuliacija)
- Sistemos funkcionalumas išreiškiamas objektų teikiamomis paslaugomis (metodais)
- Objektai tarpusavyje sąveikauja žinučių pagalba

L. Vasiliauskienė. Pj projektavimas

## ***OOP metodologija – objektų klasės***

- Objektai yra skirstomi į klases
  - Tai pačiai klasei priklausantys objektai pasižymi tomis pačiomis savybėmis (atributais ir metodais)
  - Tarp objektų klasių gali egzistuoti paveldėjimo ryšys: vaiko klasės objektai paveldi visas tėvo klasės objekto savybes bei turi papildomas savo savybes

L. Vasiliauskienė. Pj projektavimas

## ***OOP procesas***

- Apibrėžiamas sistemos veikimo kontekstas
- Sudaroma sistemą sudarančių objektų klasių diagrama
- Sukuriami reikalingi statiniai ir dinaminiai modeliai ir/arba aprašai, apibūdinantys sistemą sudarančias klases

L. Vasiliauskienė. Pj projektavimas

## ***Objektiškai orientuotas projektavimas – panaudojimo atvejų diagramos***

- Panaudojimo atvejų diagramos vaizduoja sistemos vartotojus ir sistemos jiems teikiamas funkcijas

L. Vasiliauskienė. Pj projektavimas

## **Pavyzdys (1)**

- Nagrinėsime adresų knygutės programą, kurioje saugomas kontaktų sąrašas
- Apie kiekvieną kontaktą galima sužinoti tokią informaciją: vardą, pavardę, adresą, šalį, miestą, pašto indeksą ir telefono numerį
- Turi būti galimybė pridėti naują kontaktą, redaguoti jau esančių kontaktų informaciją (išskyrus vardą ir pavardę) bei pašalinti kontaktą
- Turi būti galima rūšiuoti kontaktų sąrašą:
  - abėcėlės tvarka pagal vardą ir pavardę
  - pagal pašto indeksą
- Turi būti galimybė atsispausdinti kontaktų sąrašą

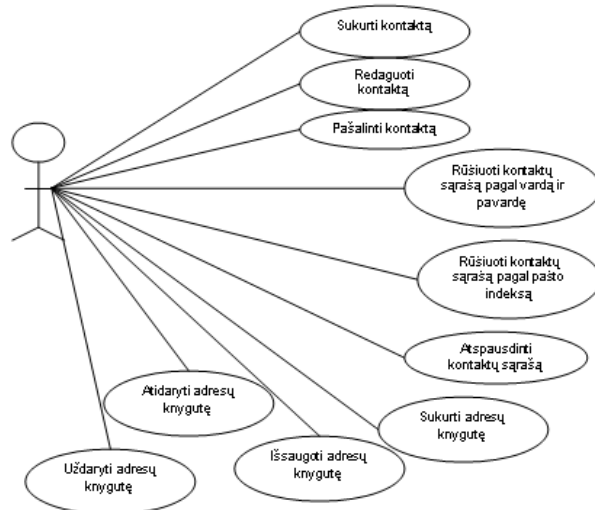
L. Vasiliauskienė. Pj projektavimas

## **Pavyzdys (2)**

- Turi būti galima:
  - sukurti naują adresų knygutę
  - atidaryti jau esamą adresų knygutę
  - uždaryti darbinę adresų knygutę
  - užsaugoti adresų knygutėje atliktus pakeitimus, jei jų buvo
- Vartotojui turi būti galimybė dirbti su keletu adresų knygutėlių, kiekvieną jų rodant atskiruose languose

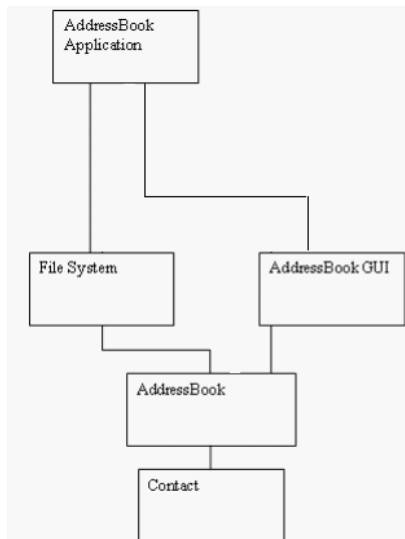
L. Vasiliauskienė. Pj projektavimas

## Adresų knygutės panaudojimo atvejų diagrama



L. Vasiliauskienė. Pj projektavimas

## Objektiškai orientuotas projektavimas - adresų knygutės programos klasių diagrama



L. Vasiliauskienė. Pj projektavimas



## Adresų knygutės programos klasių aprašymas

- **AddressBook Application:** pagrindinė klasė, atsakinga už programos paleidimą, File System ir GUI objektų sukūrimą
- **File System:** klasė, atsakinga už adresų knygutės užkrovimą ir išsaugojimą
- **AddressBook GUI:** klasė, atsakinga už adresų knygutės valdymą
- **AddressBook:** klasė, atsakinga už joje esančių kontaktų valdymą

L. Vasiliauskienė. Pj projektavimas

## Adresų knygutės programos klasės AddressBook detalus aprašymas

```
AddressBook
- collection: Person [] or Vector
- count: int (only if an array is used for collection)
- file: File
- changedSinceLastSave: boolean

+ AddressBook()
+ getNumberOfPersons(): int
+ addPerson(String firstName, String lastName, String address,
            String city, String state, String zip, String phone)
+ getFullNameOfPerson(int index): String
+ getOtherPersonInformation(int index): String[]
+ updatePerson(int index, String address, String city,
              String state, String zip, String phone)
+ removePerson(int index)
+ sortByName()
+ sortByZip()
+ printAll()
+ getFile(): File
+ getTitle(): String
+ setFile(File file)
+ getChangedSinceLastSave(): boolean
+ setChangedSinceLastSave(boolean changedSinceLastSave)
```

L. Vasiliauskienė. Pj projektavimas

## ***Funkcinio ir objektiškai orientuoto projektavimo metodų palyginimas***

- Funkcinio projektavimo privalumai ir trūkumai:
  - Funkcinis projektavimas leidžia geriau atskleisti sistemos funkcionavimą, tačiau skaitytojui suteikia mažai informacijos apie esybes, kuriomis manipuliuoja sistema
- Objektiškai orientuoto projektavimo privalumai ir trūkumai
  - Objektinis projektavimas leidžia labai detalai aprašyti sistemos naudojamas esybes, lengviau pasidūoda modifikavimui
  - Tačiau objektinis projektavimas neleidžia taip detalai atskleisti sistemoje vykdomų procesų, kaip funkcinis projektavimas

L. Vasiliauskienė. Pj projektavimas

***Klausimai ir pasiūlymai***

L. Vasiliauskienė. Pj projektavimas