

Objektorienteeritud modelleerimine



Objektorienteeritud (OO) lähenemisviis domineerib tarkvara, rakenduste ja infosüsteemide loomisel, arendamisel ja kasutamisel. Selle olemus seisneb reaalsete ja abstraktsete süsteemide ja objektide oleku ja käitumise modelleerimises arvutil tarkvara-objektide abil.

Sisu

| | |
|--|----|
| Objektorienteeritud lähenemisviisi olemus ja põhimõisted | 3 |
| Unifitseeritud modelleerimiskeel UML | 4 |
| Klassid ja objektid | 4 |
| Järgnevad on veel mõned klasside kirjeldamise näited | 6 |
| Klass Ristkülik süsteemis Geomeetria: | 6 |
| Klassisümbolite erinevad esitusviisid | 7 |
| Seosed ja klassidiagrammid | 7 |
| Agregatsioon ja kompositsioon | 7 |
| Assotsiatsioon | 8 |
| Üldistus (<i>generalization</i>) | 8 |
| Veel mõned klassidiagrammide näited: | 9 |
| Objektid ja klassid süsteemis Scratch | 11 |
| Klass Sprait Scratchis | 11 |
| Klass Sprait ja sellega seotud klassid: | 11 |
| UML tegevuskeemid | 12 |
| Standardkujudid (sümbolid) | 12 |
| Pangaautomaatide süsteem | 12 |
| Pangaautomaat | 13 |
| Tegevuste jaotus objektide vahel | 14 |
| Pangaautomaat. PIN koodi kontroll | 15 |

Objektorienteeritud lähenemisviisi olemus ja põhimõisted

Objektorienteeritud (OO) lähenemisviisi domineerib tarkvara, rakenduste ja infosüsteemide loomisel, arendamisel ja kasutamisel. Selle olemus seisneb nii reaalsete kui abstraktsete süsteemide ja objektide oleku ning käitumise modelleerimises **tarkvaraobjektide** abil.

Tarkvaraobjekt on omavahel seotud andmete ja programmide (protseduuride) kogum. Arvuti süsteemi- ja rakendustarkvara põhineb samuti tarkvaraobjektidel, milleks on dokumendid, vormid, aknad, tööriistaribad, ohjurid jms. Uuemad programmeerimiskeeled on objektorienteeritud, võimaldades luua ja kasutada tarkvaraobjekte.

OO lähenemisviisi põhimõisted on järgmised: **objekt**, **klass**, objekti **omadused** ja **tegevused**, **seosed** objektide vahel, **sündmused**.

Objekt on piiritletud osa süsteemist ning teda võib vaadelda omaette tervikuna, milleks on **konkreetne** ese, seade, isik, hoone, asutus, fail, dokument, graafiline kujund jms. Objektid on näiteks Juku jalgratas, Pille jalgratas, Peetri arvuti, õpilane Juku Naaskel, Juku pere suvila, Juku pall, Tallinna Tehnikaülikooli kampus, must kolmnurk, konkreetne dokument jms.



Klass on ühetüübiliste objektide hulk – abstraktsioon ehk üldmõiste: jalgratas, arvuti, õpilane, suvila, hoone, ülikool, Exceli tööleht, Wordi dokument või muud.

Konkreetne objekt on klassi eksemplar (ka ilming, isend, olem).



Klassi Jalgratas eksemplarid

Klassi Hulknurk eksemplarid

Omadused (atribuudid) on karakteristikud, mis identifitseerivad objekti ja iseloomustavad selle olekut, väljanägemist, ...

Kasutatavate omaduste valik sõltub rakenduse liigist, eesmärkidest, ...

Konkreetses ratta atribuudid: liik, mark, mõõtmed, värv, käikude arv, olek, ...

Tarkvaraobjekti ratas atribuudid müügisüsteemis: samad, mis eelmisel rattal ning lisaks veel näiteks tarnija, kauplus, ...

Arvutisimulatsioonis on atribuutideks (pildi) suurus, kiirus, asukoht, ...

Graafikaobjekt: laius, kõrgus, asukoht ekraanil, täitevärv, pindala, ümbermõõt, ...

Tegevused (meetodid, operatsioonid), mida sooritab objekt ise või sooritatakse objektiga. Valik sõltub süsteemi liigist, eesmärkidest, ...

Rataste müük: ratta lisamine või eemaldamine, andmete muutmine, ...

Simulatsioon: kiiruse suurendamine, pidurdamine, peatamine, ...

Graafikaobjekt: teisaldamine, kopeerimine, mõõtmete muutmine, täitevärvi muutmine, ...

Sündmused. Objekt võib reageerida teatud välistele mõjudele: muudab olekut, käivitab tegevuse, ...

Konkreetne ratas: kokkupõrge, ...

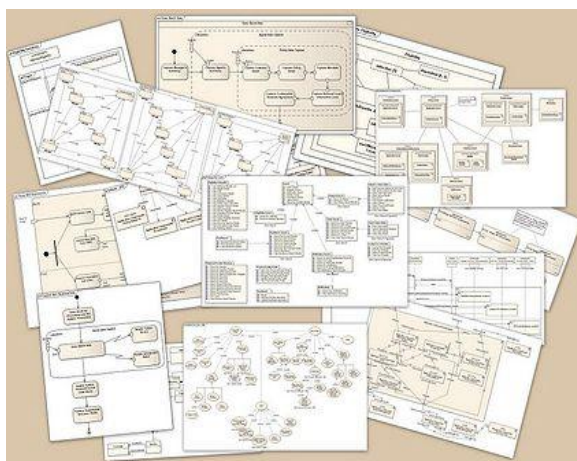
Simulatsioon: vajutus tühikule – alusta sõitu, hiireklõps – peatu jms.

Seosed (suhted, relatsioonid). Objektidel võivad olla seosed teiste objektidega. Seosed on erineva olemuse ja tähendusega:

- omandisuhe: jalgratas kuulub Jukule, arvuti kuulub Peetrile,
- töösuhe: Peeter Kask töötab SEB pangas,
- sisaldavuse suhe: protsessor on arvutis, tööleht leht kuulub töövihikusse.

Unifitseeritud modelleerimiskeel UML

Objektorienteeritud graafiline keel on süsteemide visuaalseks analüüsiks, kavandamiseks, loomiseks ja dokumenteerimiseks. Keeles on kindel valik lihtsad graafilisi kujutisi (sümboleid) infosüsteemide ja rakenduste erinevate komponentide, olemite ja tegevuste tähistamiseks ning elementide ühendamiseks. Sümbolite sisse paigutatakse kindla tähenduse ja struktuuriga tekstid. Loodavate diagrammide abil saab kirjeldada süsteemi struktuuri, olekuid, tegevusi jms ning see on suhtlemisvahendiks informaatikute ja teiste erialade spetsialistide vahel.



Kasutatavate diagrammide tüübid ning nende detailiseerimise aste sõltuvad süsteemi iseloomust ja arendusprotsessi faasist: ülesande püstitus, analüüs, projekteerimine (disain) jms.

Süsteemide struktuuri, oleku ja käitumise kirjeldamiseks on erinevat tüüpi **diagramme** (skeeme ehk mudeleid):

- klassi- ja objektidiagrammid
- tegevusdiagrammid
- kasutusjuhtude diagrammid
- olekudiagrammid
- ...

Peamised UML diagrammid ülesande püstituse, analüüsi ja disaini faasis on **klassi-** ja **tegevusdiagrammid**.

Klassid ja objektid

UMLi diagrammidel esitatakse klass ristkülikuna, millel on üldjuhul kolm sektsiooni:

- nimi
- atribuudid (omadused)
- operatsioonid (tegevused, meetodid)

Nendest kohustuslik on ainult nimi.

Järgnevas näites on toodud kooli infosüsteemi klassi **Õpilane** lihtsustatud kirjeldus.

| Õpilane | Klassi nimi. Nimi peab algama suurtähega. |
|--|---|
| kood eesnimi perenimi sünniaeg aadress klass ... | Atribuudid (omadused ehk parameetrid) Igal atribuudil on nimi . Võib olla näidatud ka tüüp : arv, tekst, kuupäev jm. Konkreetsetel objektidel on omadusel kindel väärtus: K20567, Juku, Naaskel, 26.10.2001, Spordi 5-9, 4B, ... K20573, Pille, Pilt, 23.06.2001, Teatri 7-13, 4A, ... |
| lisamine() eemaldamine() muutmine() | Operatsioonid (tegevused ehk meetodid) Esitatakse tavaliselt kujul nimi() . Siin on meetodite tähendus järgmine: lisamine() – uue õpilase lisamine, eemaldamine() – õpilase eemaldamine, muutmine() – õpilase andmete muutmine. |

| Arvuti | Siin on veel ühe klassi kirjelduse näide: klass Arvuti müügisüsteemis. |
|---|---|
| mark protsessor taktsagedus sisemälu_maht kõvaketta_maht hind / arv ... | Üheks omaduseks on olemasolevate arvutite arv antud grupis. Tegemist on nn muutuva ehk arvutatava atribuudiga. Sama marki arvutite lisamisel või eemaldamisel muudetakse omadust arv. Tuletatava omaduse tunnuseks on kaldkriips (/) omaduse nimetuse ees. |
| lisamine() eemaldamine() muutmine() ... | |

Allpool on toodud näiteks tabel, mis sisaldab andmeid arvutite (objektid) kohta: arvutite omadusi.

| Mark | Protsessori tüüp | Taktsagedus | Sisemälu, MB | Kõvaketas, GB | Hind, € | Arv |
|---------|------------------|-------------|--------------|---------------|---------|-----|
| Aragorn | AMD Athlon64 | 3,0 | 512 | 160 | 416 | 13 |
| Balrog | Intel Celeron | 2,8 | 256 | 80 | 349 | 21 |
| Eldar | Intel Xeon | 3,0 | 512 | 160 | 388 | 7 |
| Elend | Intel Celeron | 4,0 | 2048 | 400 | 1083 | 0 |
| Faram | AMD Athlon64 | 3,0 | 512 | 210 | 499 | 12 |
| Frodo | AMD Athlon64 | 3,0 | 4072 | 500 | 1108 | 25 |
| Gandalf | Intel Xeon | 3,2 | 512 | 160 | 708 | 2 |
| Kalvar | Intel Xeon | 3,0 | 512 | 160 | 583 | 7 |
| Marvin | Intel Celeron | 3,0 | 512 | 160 | 499 | 0 |
| Rohan | AMD Athlon64 | 3,5 | 512 | 160 | 491 | 0 |
| Sauron | Intel Xeon | 2,5 | 256 | 180 | 283 | 5 |

Järgnevad on veel mõned klasside kirjeldamise näited.

| Raamat |
|---|
| ISBN autor nimetus aasta kirjastus hind ... |
| lisa() eemalda() muuda_andmeid() ... |

| Fail |
|--|
| nimi tüüp maht loodud muudetud ... |
| ava() sulge() eemalda() muuda_nime() ... |

| Tööleht |
|--|
| nimi ridade arv veergude arv aktiivsus |
| teisaldamine() kopeerimine() eemaldamine() aktiveeri() ... |

| Kujund |
|---|
| nimi x-koordinaat y-koordinaat laius kõrgus ... |
| lisa() teisalda() kopeeri() muuda_laiust() muuda_kõrgust() muuda_värvi() |

Klass Ristkülik süsteemis Geomeetria:

| Ristkülik |
|---|
| laius kõrgus / pindala / übermõõt / diagonaal ... |
| leia_pindala() leia_über() leia_diagonaal() muuda() ... |

$$S = a \cdot b$$

$$P = 2(a + b)$$

$$d = \sqrt{a^2 + b^2}$$

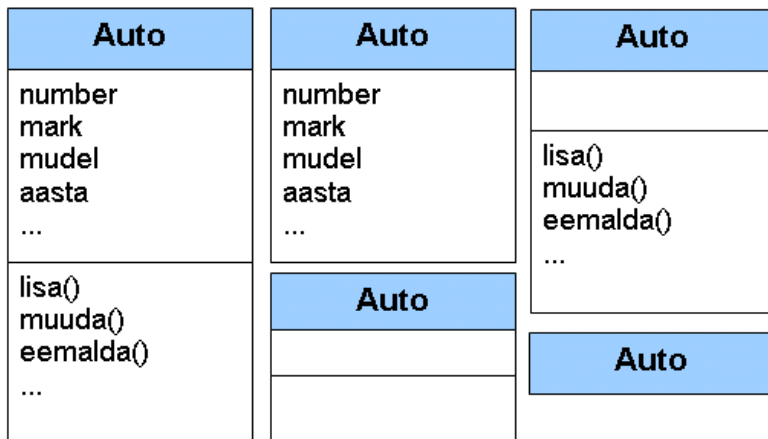
NB! Sisestatavad andmed: laius (a) ja kõrgus (b) ning

tuletatavad omadused: pindala (S), übermõõt (P) ja diagonaal (d).

Viimaste ees võib tuletatava omaduse tunnusena olla / (kaldkriips), aga pole kohustuslik.

Klassisümbolite erinevad esitusviisid

Antud juhul võib kasutada erinevaid variante: tühjad seksioonid, ärajäetud seksioonid jms, kuid sageli kasutatakse ainult nime (diagrammidel).



Seosed ja klassidiagrammid

Süsteemi kuuluvatel objektidel on omavahel teatud seosed – relatsioonid. Seoseid (relatsioon, suhe) näidatakse diagrammidel joonte abil. Klassidiagramm näitab klasse ja võimalikke seoseid nende objektide vahel. Reeglina on igas seoses ainult kaks klassi. Seosele võib anda nime, mida võib näidata skeemil. Seose mõlema otsa juures võib esitada korduse, mis näitab, mitu antud klassi objekti võib olla seotud teise klassi objektiga. Korduste tüüpvariandid ja nende esitus UMLis on järgmised:

n – kindel arv, näiteks 1, 4, 256

n1, n2, n3, ... – loend, näiteks 2, 4, 6

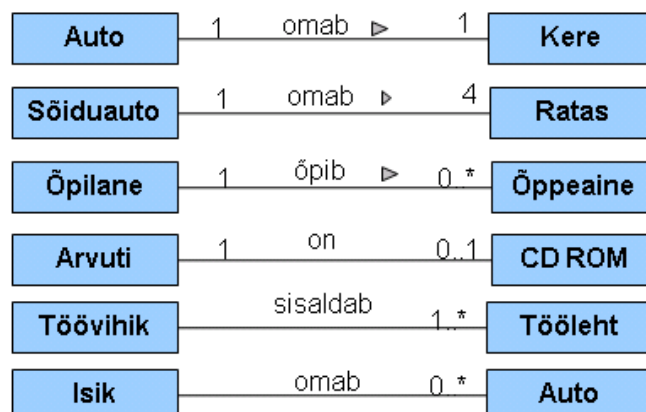
0..1 või 0, 1 – null või üks

* – suvaline hulk

1..* – üks või rohkem

0..* – null või rohkem

Seose nime ja korduse väärtuse 1 võib jätta ära.



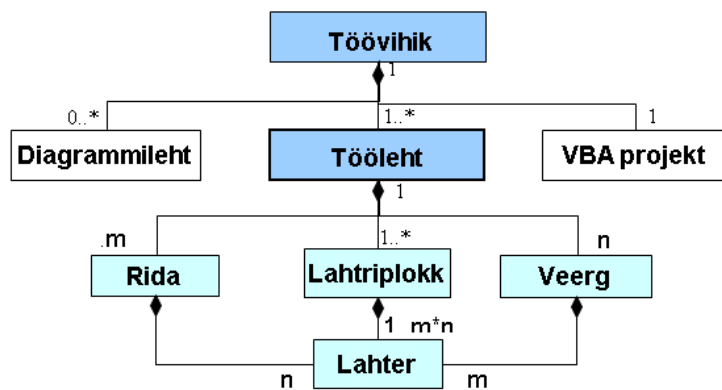
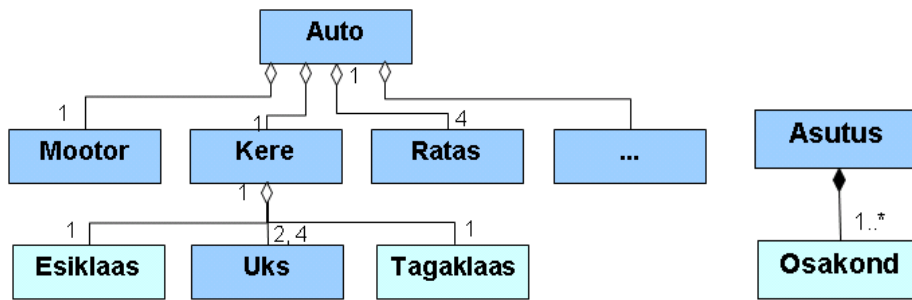
Eristatakse järgmisi seoste liike:

- agregatsioon
- assotsiatsioon
- üldistus

Agregatsioon ja kompositsioon

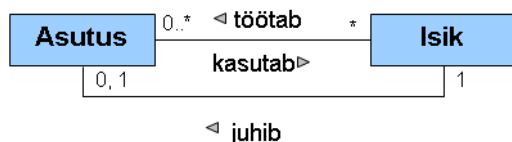
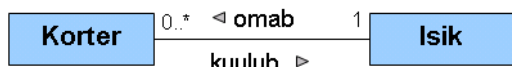
Agregatsioon ehk sisalduvuse seos. Üks objekt sisaldab teisi. Tervik ja osad. Terviku poolel võib olla täitmata romb ◊ (seda ei pea tingimata olema).

Kompositsioon – agregatsiooni erijuht – osa saab eksisteerida ainult tervikus, mille tähistuseks on täidetud romb ◆. Vt järgnevat skeemi.



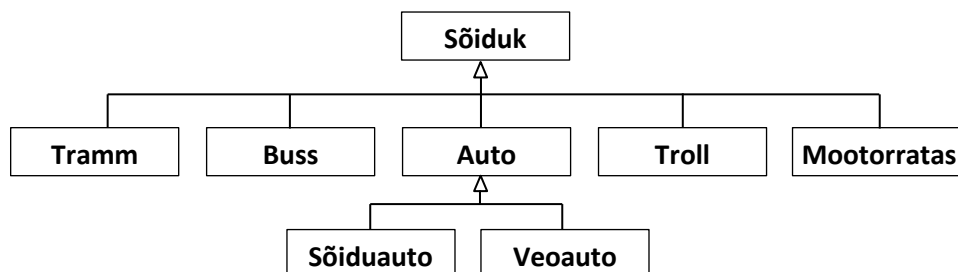
Assotsiatsioon

Assotsiatsioon – suvaline mõtet omav seos (suhe) objektide vahel – omandisuhe, töösuhe jms. Assotsiatsiooni erijuht on agregatsioon.

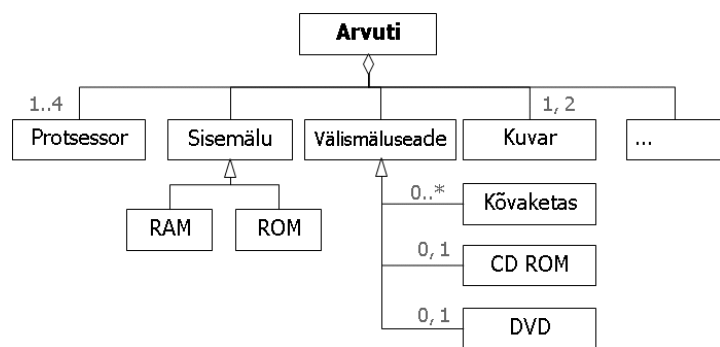


Üldistus (*generalization*)

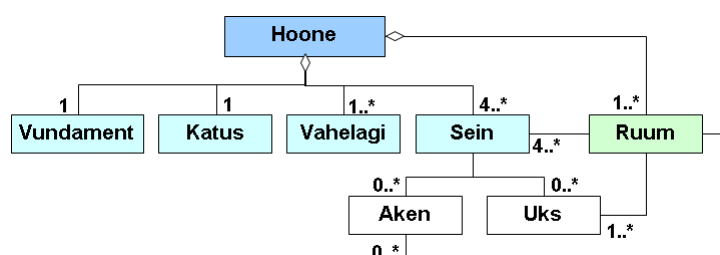
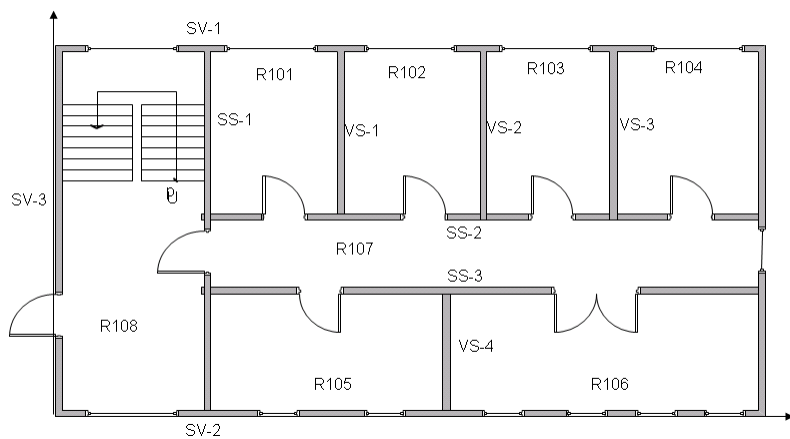
Üldistus – esivanema (ülemklassi) ja järglaste (alamklasside) seos. Järglased pärivad esivanema omadused ning võivad omada täiendavaid omadusi. Kujund \triangle – esivanema poole.

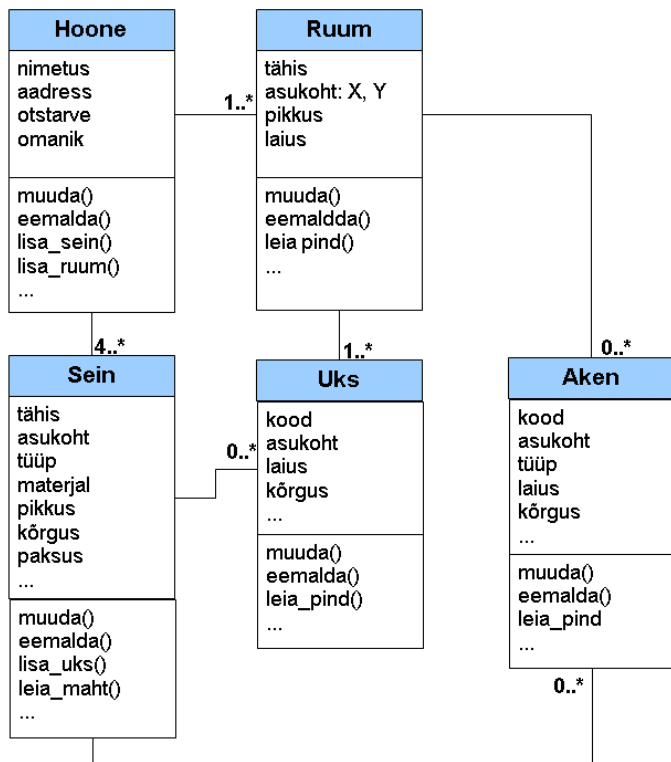


Veel mõned klassidiagrammide näited:



Hoone põhiklassid:



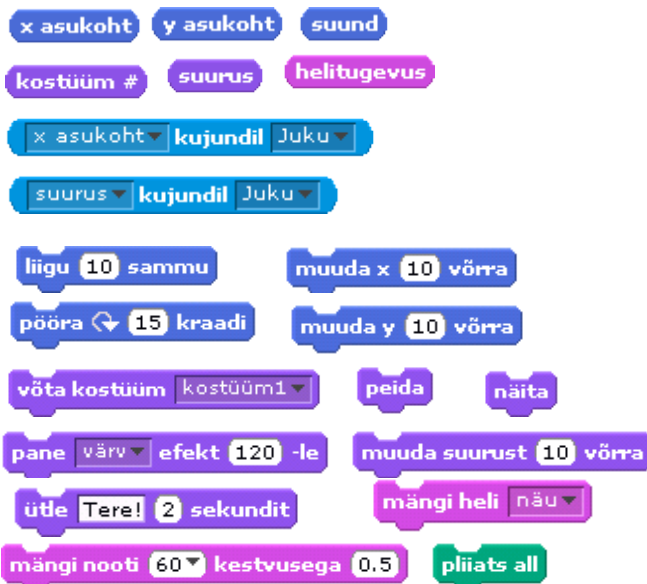


Objektid ja klassid süsteemis Scratch

Klass Sprait Scratchis



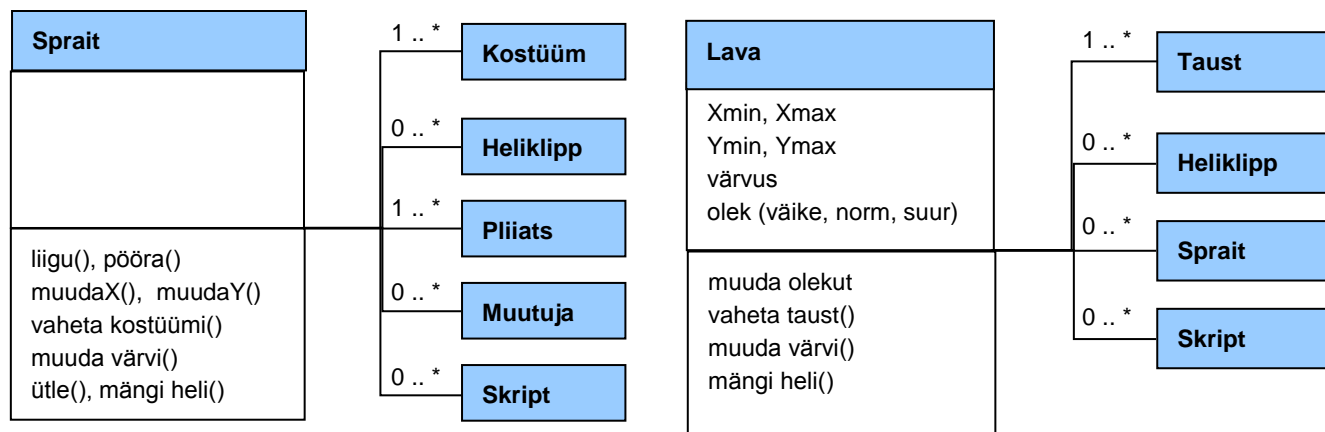
| Sprait |
|---|
| nimi asukoht: x,y suund suurus värvus nähtavus |
| liigu() pööra() muudaX() muudaY() vaheta kostüümi() muuda värvi() ütle() mängi heli() ... |



Sprait (*sprite*) on universaalne graafikaobjektide klass. Spraitidel on kindel valik **omadusi**: nimi, asukoht, suurus, jms. Spraitidega saab **käskude** ehk **plokkide (meetodite)** abil määrata erinevaid tegevusi: liigutada, pöörata, muuta suurust jms. Käskudest saab moodustada programmiüksuseid – **skripte**.

Lava on ekraani piirkond, kus toimub spraitide tegevus.

Klass Sprait ja sellega seotud klassid:



UML tegevuskeemid

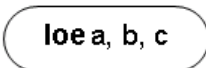
Tegevuskeem ehk **tegevusdiagramm** (*Activity Diagram*) esitab graafiliselt antud protsessi või töö täitmiseks vajalikke tegevusi ja nende järjekorda. Tegevuskeemi kasutatakse äri- ja tööprotsesside ning algoritmide kirjeldamiseks.

Standardkujundid (sümbolid)

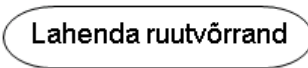
- Protsessi algus
- Protsessi lõpp



Tegevus



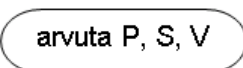
loe a, b, c



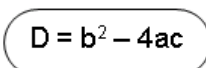
Lahenda ruutvõrrand



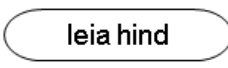
kontrolli PIN-koodi



arvuta P, S, V



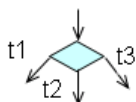
$D = b^2 - 4ac$



leia hind

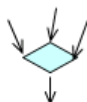


Siire

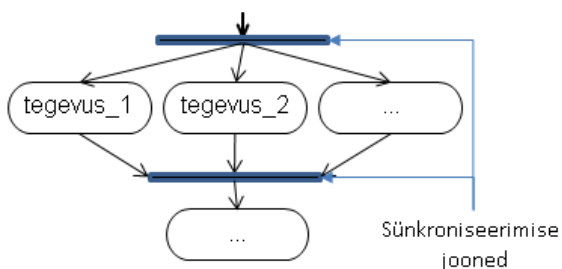


Hargnemine

t1, t2, ... - tingimused

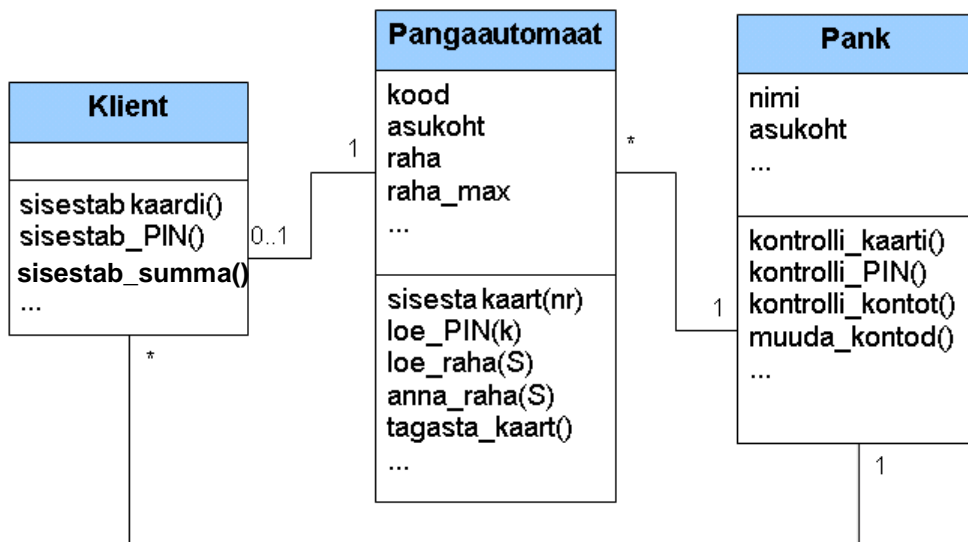


Ühinemine



Tegevusi täidetakse paralleelselt
Järjekord pole oluline!

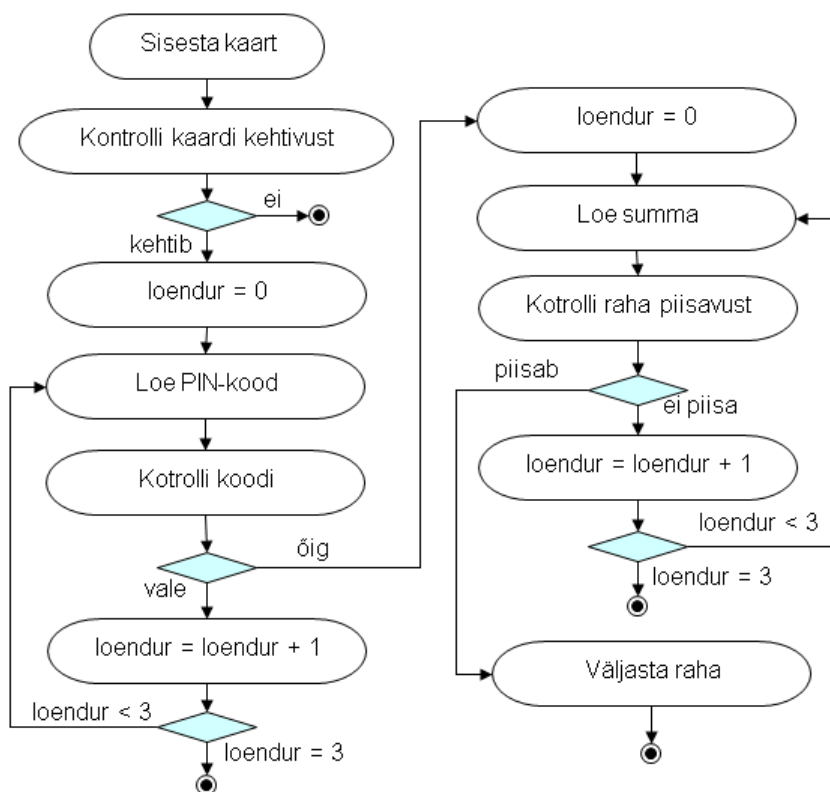
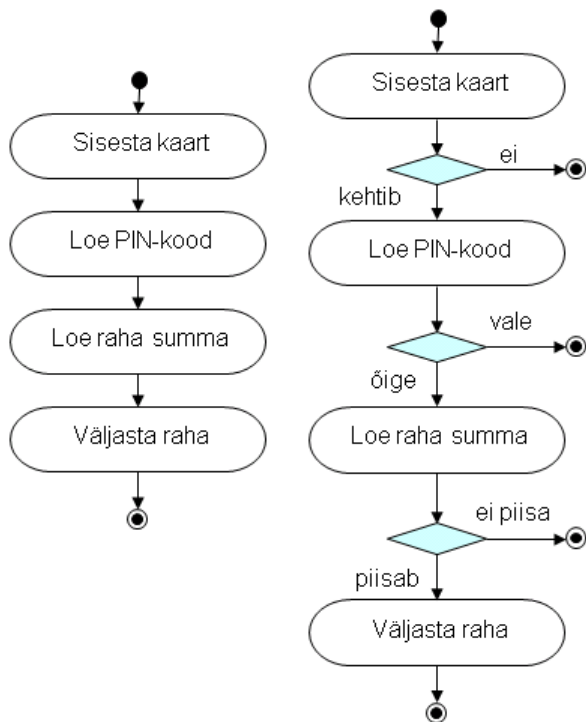
Pangaautomaatide süsteem



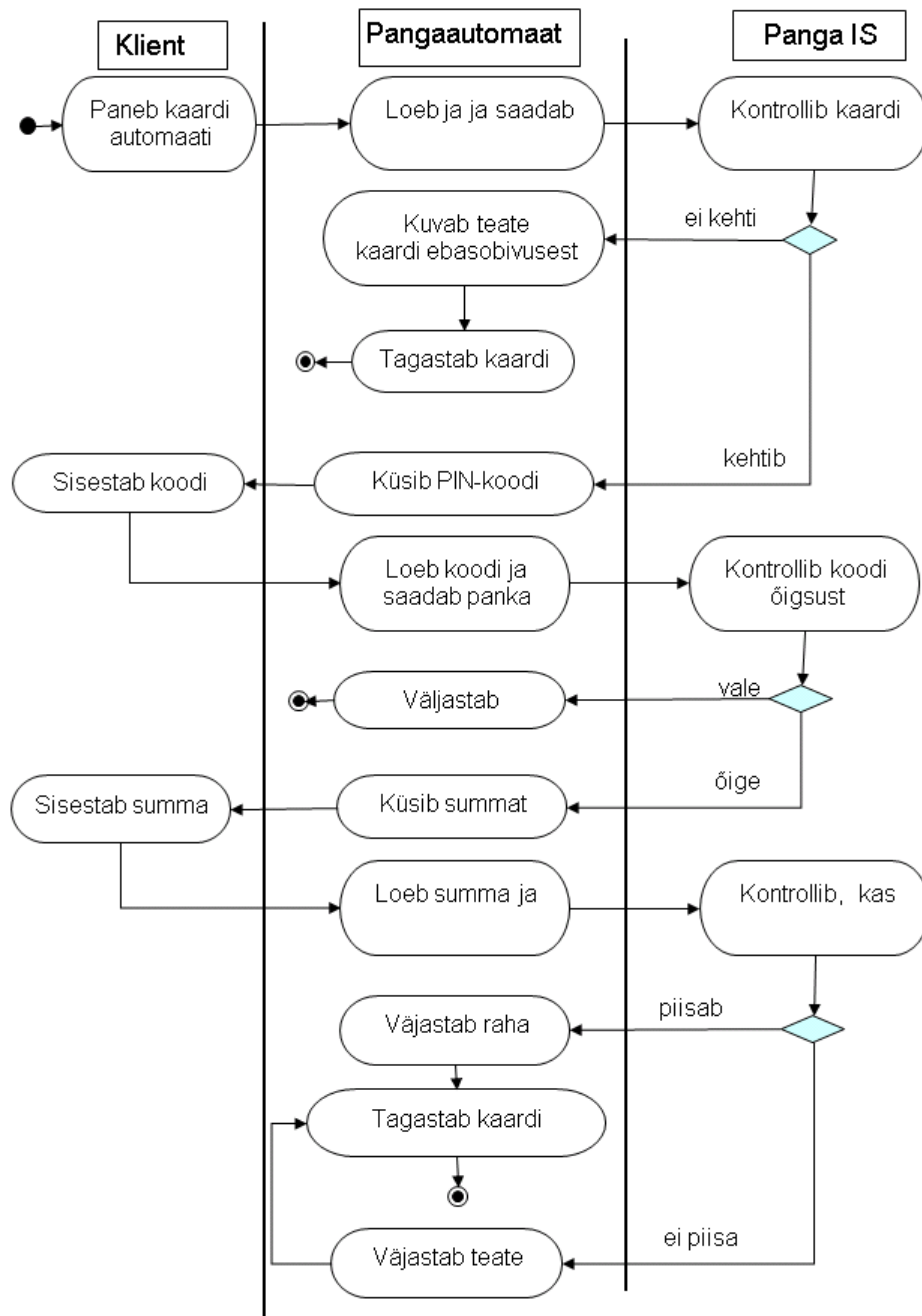
Pangaautomaat

Klass ja erineva täpsusega tegevusskeemid:

| Pangaautomaat |
|--|
| kood asukoht raha raha_max ... |
| sisesta kaart(nr) loe_PIN(k) loe_raha(S) anna_raha(S) tagasta_kaart() ... |



Tegevuste jaotus objektide vahel



Pangaautomaat. PIN koodi kontroll

Lubatud on kolm katset:

