

# 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  |
| Klass Sprait Scratchis .....                                   | 7  |
| Klassisümbolite erinevad esitusviisid .....                    | 7  |
| Seosed ja klassidiagrammid .....                               | 8  |
| Agregatsioon ja kompositsioon .....                            | 8  |
| Assotsiatsioon .....                                           | 9  |
| Üldistus ( <i>generalization</i> ) .....                       | 9  |
| Hoone põhiklassid .....                                        | 10 |
| Objektid ja klassid süsteemis Scratch.....                     | 11 |
| Klass Sprait ja selle alamklassid .....                        | 11 |
| UML tegevusskeemid .....                                       | 12 |
| Standardkujundid (sümbolid).....                               | 12 |
| Pangaautomaatide süsteem.....                                  | 12 |
| Pangaautomaat .....                                            | 13 |
| Tegevuste jaotus objektide vahel.....                          | 14 |
| Pangaautomaat. PIN koodi kontroll .....                        | 15 |

## Objektorienteeritud lähenemisviisi olemus ja põhimõisted

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 **tarkvaraobjektide** abil.

Tarkvaraobjekt on omavahel seotud andmete ja programmide (protseduuride) kogum. Ka arvuti süsteemi- ja rakendustarkvara põhineb tarkvaraobjektidel, milleks on dokumendid, vormid, aknad, tööriistaribad, ohjurid jmt. Uuemad programmeerimiskeeled on objektorienteeritud ning võimaldavad kasutada ja luua 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, Juku Naaskel, Juku isa suvila, Juku pall, TTÜ, must kolmnurk, konkreetne dokument ...



**Klass** on ühetüübiliste objektide hulk – abstraktsioon ehk üldmõiste:

jalgratas, arvuti, õpilane, suvila, hoone, ülikool, Exceli tööleht, Wordi dokument ...

**Konkreetne objekt** on klassi eksemplar (ilming)



Klassi Jalgratas eksemplarid



Klassi Hukknurk eksemplarid

**Omadused** (atribuudid) on karakteristikud, mis identifitseerivad objekti ja iseloomustavad selle olekut, väljanägemist jms. Kasutatavate omaduste valik sõltub rakenduse liigist, eesmärkidest jms.

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

**Tarkvaraobjekti ratas** atribuudid müügisüsteemis: samad, mis eelmisel rattal, lisaks veel näiteks tarnija, kauplus jms;

**arvutisimulatsioonis** on atribuutideks (pildi) suurus, kiirus, asukoht jms,

**graafikaobjektil**: laius, kõrgus, asukoht ekraanil, täitevvärv, pindala, ümbermõõt ...

**Tegevused** (meetodid, operatsioonid), mida täidab objekt ise või täidetakse objektiga. Valik sõltub süsteemi liigist, eesmärkidest jms.

**Rataste müük**: lisamine, eemaldamine, andmete muutmine jms.

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

**Graafikaobjekt**: teisaldamine, kopeerimine, mõõtmete muutmine, täitevvärv muutmine jms.

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

**Konkreetne ratas:** kokkupõrge, ...

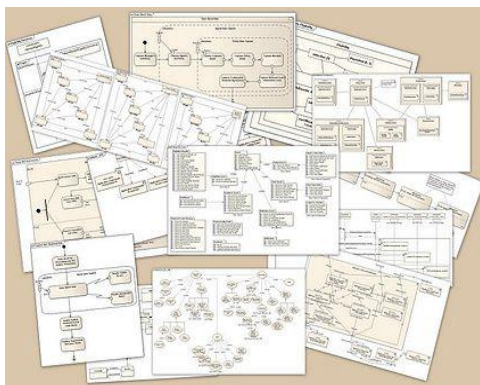
**Simulatsioon:** vajutus tühikule – alusta sõitu, hiireklõps – peatu jms.

**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 erinevate erialade spetsialistide vahel.



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

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

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

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

Rahvusvaheline standard (ISO, OMG)

### Klassid ja objektid

**UMLi** diagrammidel esitatakse klass ristkülikuna. Üldjuhul kolm sektsiooni:

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

Kohustuslik on ainult nimi.

Näites on toodud kooli infosüsteemi klassi **Õpilane** lihtsustatud kirjeldus.

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

| Arvuti                                                                                       | Kõrval on veel ühe klassi kirjelduse näide: klass <b>Arvuti</b> müügisüsteemis.                                                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mark<br>protsessor<br>taktsagedus<br>sisemälu_maht<br>kõvaketta_maht<br>hind<br>/ arv<br>... | <b>NB!</b> Objektiks ei ole üksik arvuti, vaid nõ kaubagrupp: ühesuguste parameetritega arvutid.<br><br>Üheks omaduseks on olemasolevate arvutite arv antud grupis. Tegemist on nn muutuva ehk arvutatava atribuudiga. Sama marki arvutite lisamisel või eemaldamisel muudetakse omadust arv. Arvutatava omaduse tunnus on kaldkriips (/) omaduse nimetuse ees. |
| lisamine()<br>eemaldamine()<br>muutmine()<br>...                                             |                                                                                                                                                                                                                                                                                                                                                                 |

Allpool on toodud näiteks tabel, mis sisaldab andmeid arvutite 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           | 1 083   | 0   |
| Faram   | AMD Athlon64     | 3,0         | 512          | 210           | 499     | 12  |
| Frodo   | AMD Athlon64     | 3,0         | 4072         | 500           | 1 108   | 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   |
|         |                  |             |              |               |         |     |

Veel mõned klasside kirjeldamise näited:

| Raamat                                                        | Fail                                                 | Tööleht                                                                | Kujund                                                                                  |
|---------------------------------------------------------------|------------------------------------------------------|------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| ISBN<br>autor<br>nimetus<br>aasta<br>kirjastus<br>hind<br>... | nimi<br>tüüp<br>maht<br>loodud<br>muudetud<br>...    | nimi<br>ridade arv<br>veergude arv<br>aktiivsus                        | nimi<br>x-koordinaat<br>y-koordinaat<br>laius<br>kõrgus<br>...                          |
| lisa()<br>eemalda()<br>muuda_andmeid()<br>...                 | ava()<br>sulge()<br>eemalda()<br>muuda_nime()<br>... | teisaldamine()<br>kopeerimine()<br>eemaldamine()<br>aktiveeri()<br>... | lisa()<br>teisalda()<br>kopeeri()<br>muuda_laiust()<br>muuda_kõrgust()<br>muuda_värvi() |

Klass Ristkülik süsteemis Geomeetria:

| Ristkülik                                                             |
|-----------------------------------------------------------------------|
| laius<br>kõrgus<br>/ pindala<br>/ ümbermõõt<br>/ diagonaal<br>...     |
| leia_pindala()<br>leia_ümbere()<br>leia_diagonaal()<br>muuda()<br>... |

$$S = a \cdot b$$

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

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

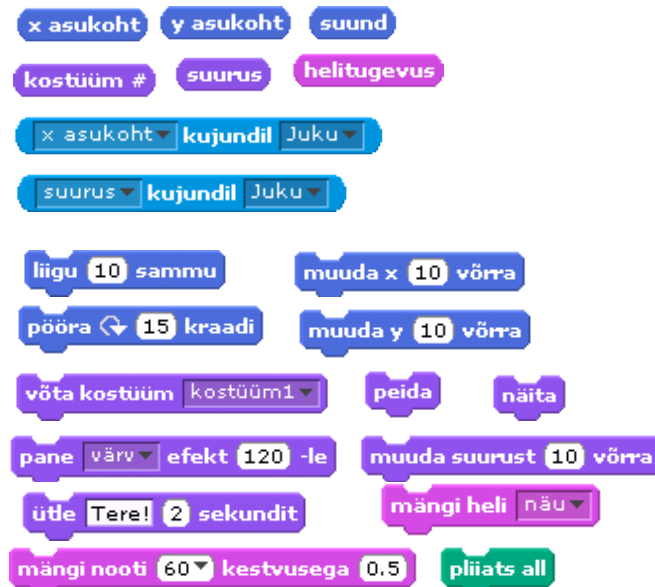
**NB! Sisestatavad** andmed: laius (a) ja kõrgus (b) ning **tuletatavad** omadused: pindala (S), ümbermõõt (P) ja diagonaal (d), millest viimaste ees võib olla / (kaldekriips)

## Klass Sprait Scratchis

Universaalne baasklass – superklass. Selle alusel saab luua alamklasse.



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



## Klassisümbolite erinevad esitusviisid

Antud juhul võib kasutada erinevaid variante: tühjad seksioonid, ärajäetud seksioonid jms.

Sageli kasutataksegi ainult nime (diagrammidel).

| Auto                                    | Auto                                    | Auto                                  |
|-----------------------------------------|-----------------------------------------|---------------------------------------|
| number<br>mark<br>mudel<br>aasta<br>... | number<br>mark<br>mudel<br>aasta<br>... |                                       |
| lisa()<br>muuda()<br>eemalda()<br>...   |                                         | lisa()<br>muuda()<br>eemalda()<br>... |
|                                         | Auto                                    | Auto                                  |
|                                         |                                         |                                       |

## Seosed ja klassidiagrammid

Süsteemi kuuluvatel objektidel on omavahel teatud seosed. Seos (relatsioon, suhe) näidatakse diagrammidel joone abil. Klassidiagramm näitab klasse ja võimalikke seoseid nende objektide vahel. Reeglina on igas seoses ainult kaks klassi. Seosele võib anda nime, mille 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, näit 1, 4, 256

$n_1, n_2, n_3, \dots$  – loend, näit.: 2, 4, 6

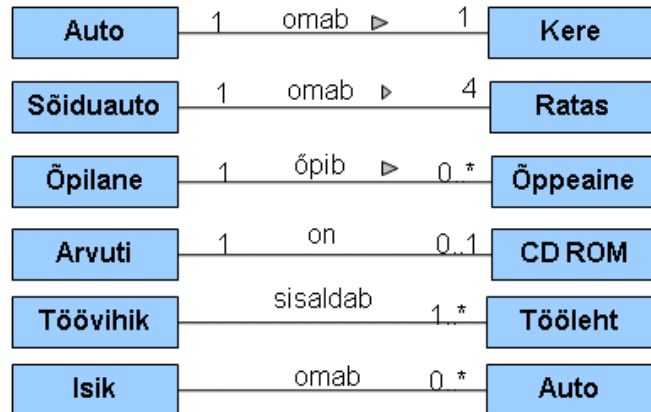
$0..1$  või  $0, 1$  – null või üks

$*$  – suvaline hulk

$1..*$  – üks või enam

$0..*$  – null või enam

Seose nime ja korduse väärtuse 1 võib jätta ära.



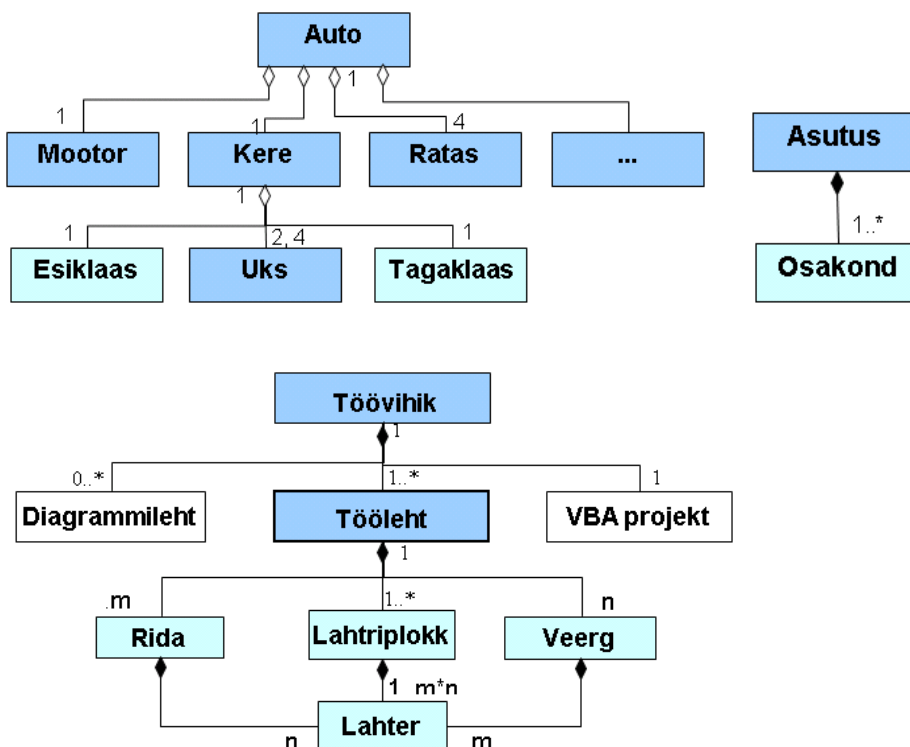
Eristatakse järgmisi seoste liike:

- agregatsioon
- assotsiatsioon
- üldistus

## Agregatsioon ja kompositsioon

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

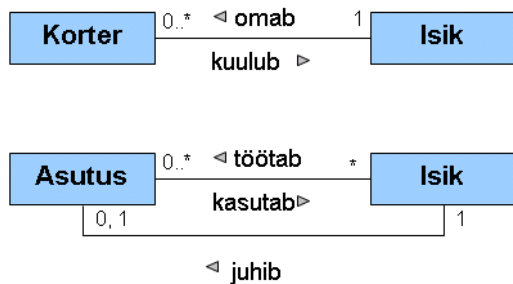
**Kompositsioon** – agregatsiooni erijuht – osa saab eksisteerida ainult tervikus. Tähistus – täidetud romb





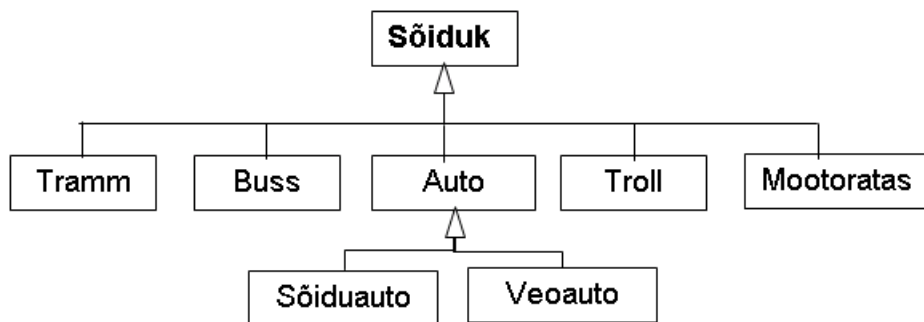
## Assotsiatsioon

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

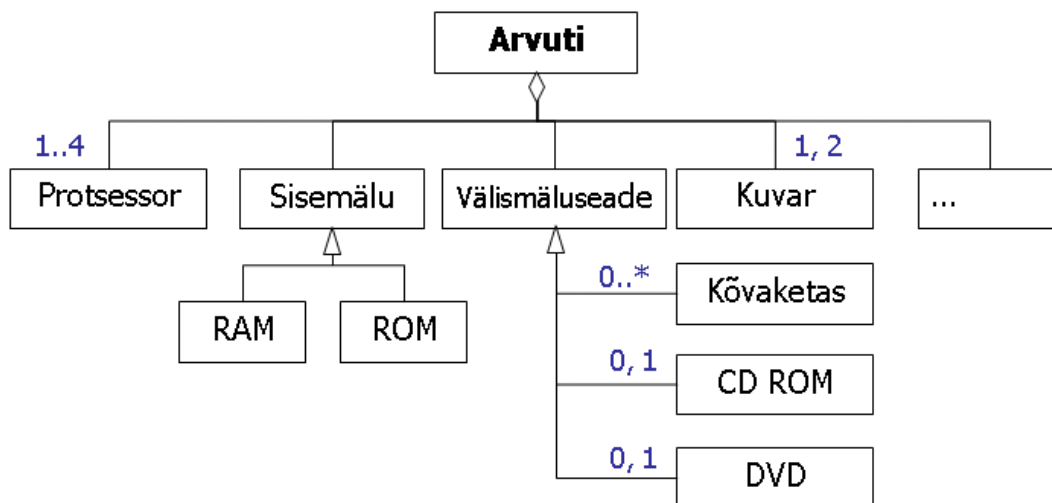


## Üldistus (generalization)

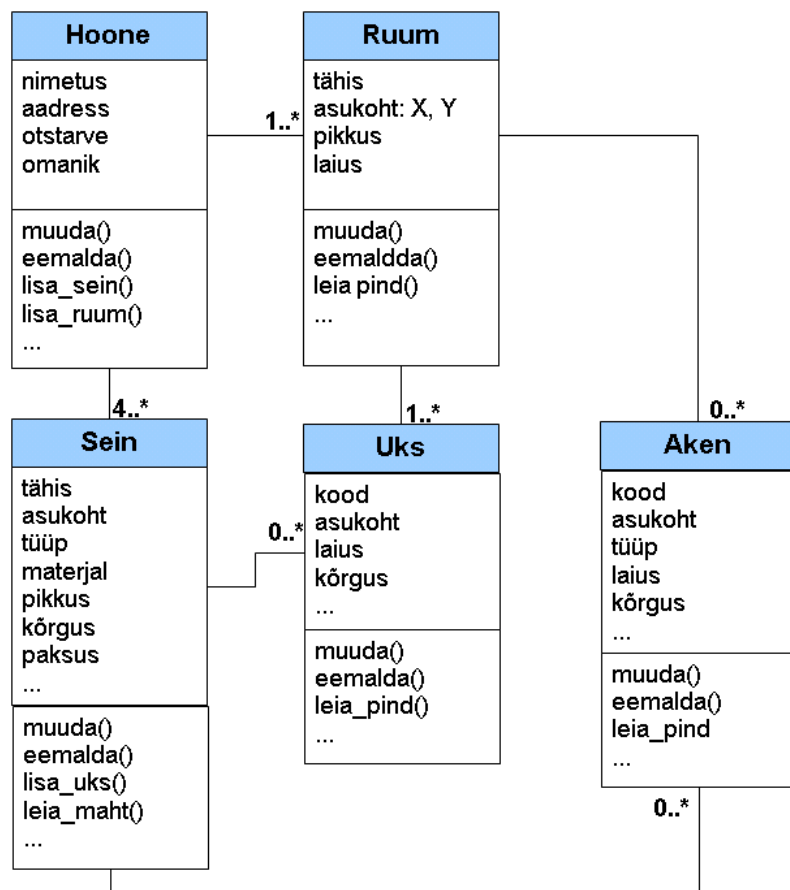
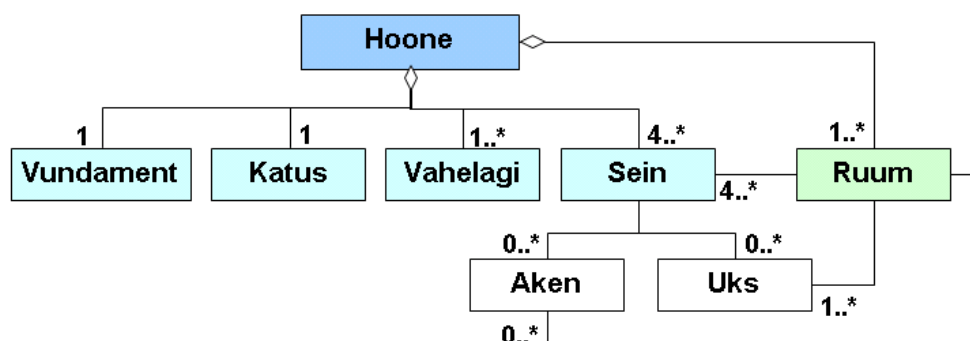
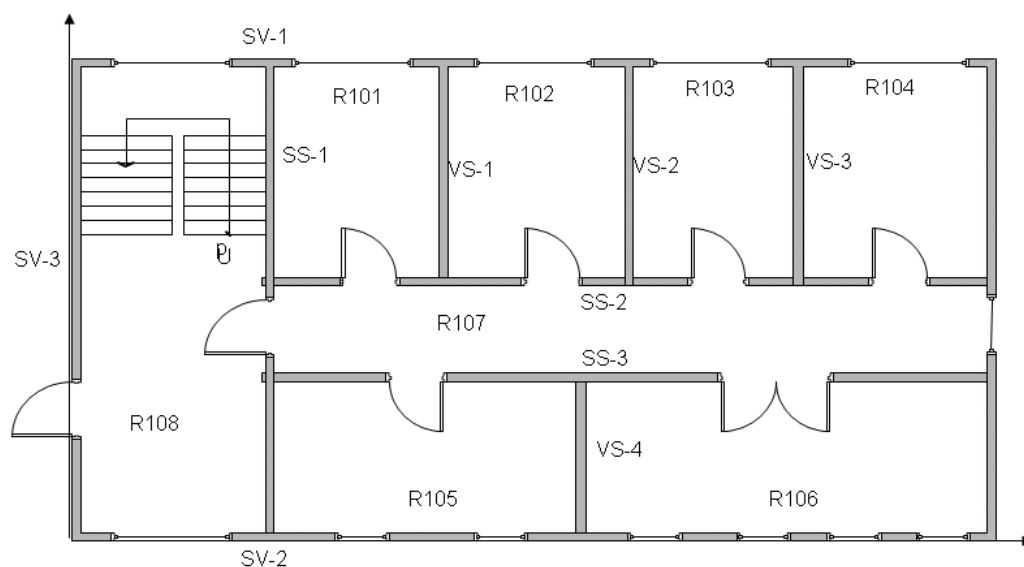
**Üldistus** – esivanema (superklassi) ja järglaste (alamklasside) seos. Järglased pärivad esivanema omadused ning võivad omada täiendavaid omadusi. Kujund  $\triangle$  – esivanema poolel.



Veel mõned klassidiagrammide näited:



## Hoone põhiklassid

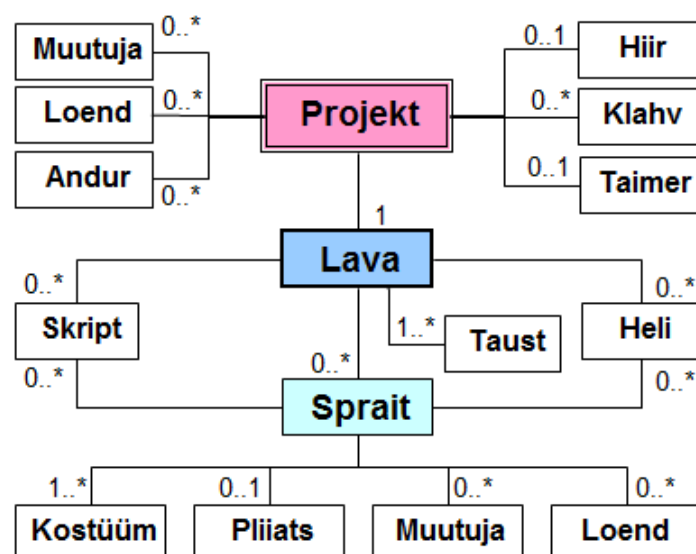
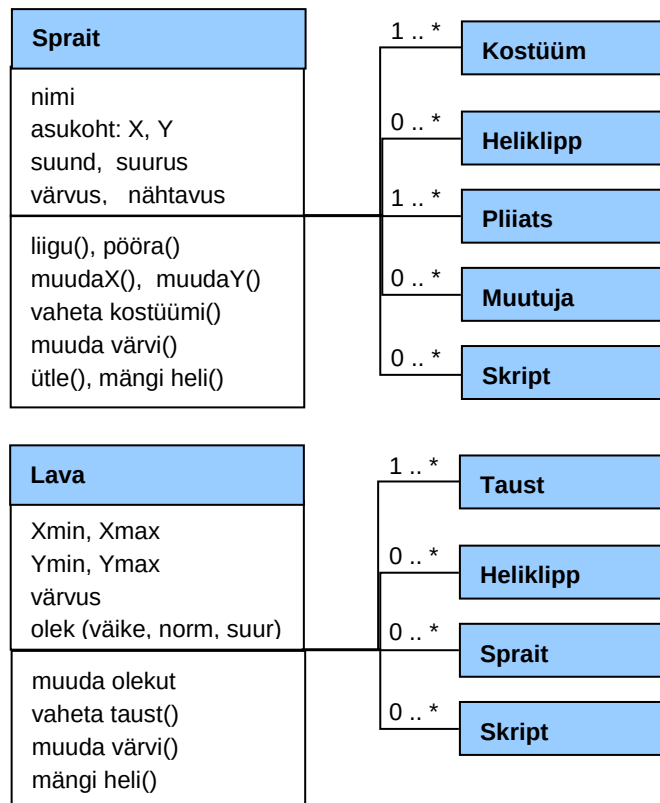


## Objektid ja klassid süsteemis Scratch

Universaalne graafikaobjektide klass on **sprait** (*sprite*). Spraitidel on kindel valik **omadusi**: nimi, asukoht, suurus, jms. Spraitidega saab määrata erinevaid tegevusi: liigutada, pöörata, muuta suurst jms. Seda saab teha kindlate **käskude** ehk **plokkide (meetodite)** abil. Käskudest saab moodustada programmiüksusi – **skripte**. Klassil sprait on mõned **alamklassid** (kostüüm, pliiats).

**Lava** on ekraani piirkond, kus toimub spraitide tegevus.

### Klass Sprait ja selle alamklassid



## UML tegevusskeemid

**Tegevusskeem** ehk **tegevusdiagramm** (*Activity Diagram*) esitab graafiliselt antud protsessi või töö täitmiseks vajalikud tegevused ja nende järjekorra. Kasutatakse äri- ja tööprotsesside ning algoritmide kirjeldamiseks

### Standardkujundid (sümbolid)

- Protsessi algus
- Protsessi lõpp



Tegevus

loe a, b, c

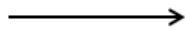
Lahenda ruutvõrrand

kontrolli PIN-koodi

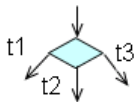
arvuta P, S, V

$D = b^2 - 4ac$

leia hind

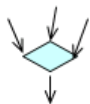


Siire

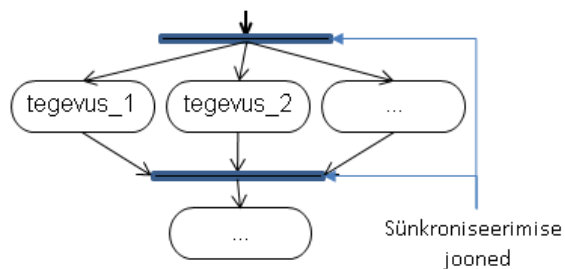


Hargnemine

t1, t2, ... - tingimused

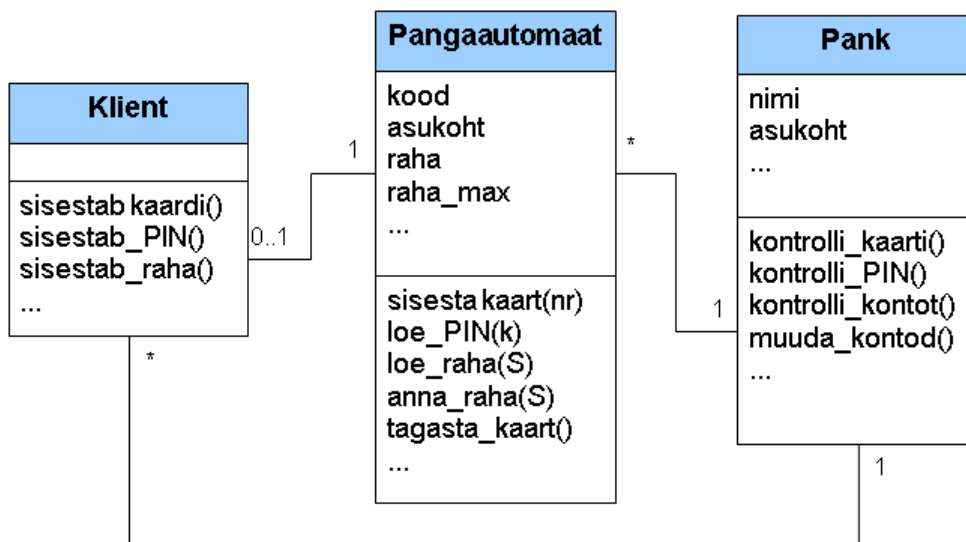


Ühinemine



Tegevusi täidetakse paralleelselt  
Järjekord pole oluline!

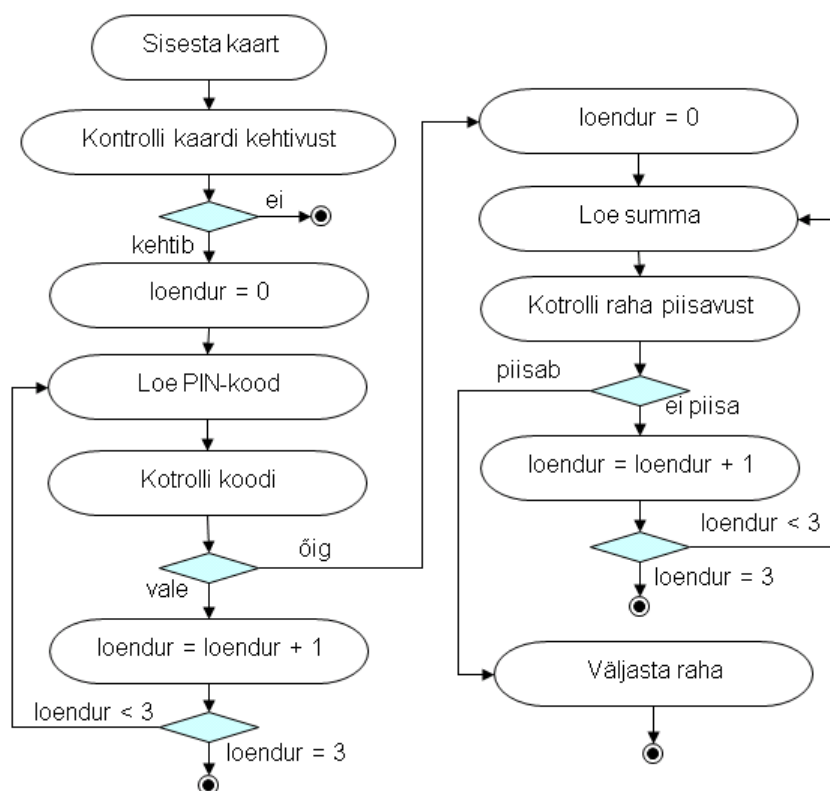
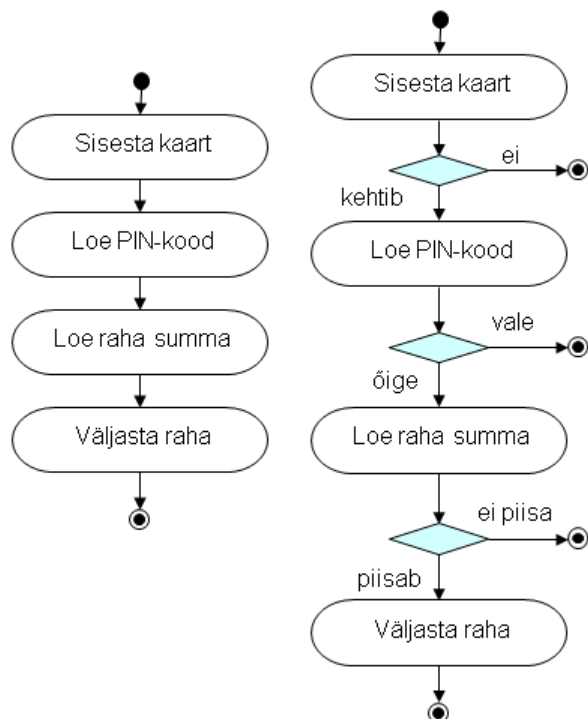
### Pangaautomaatide süsteem



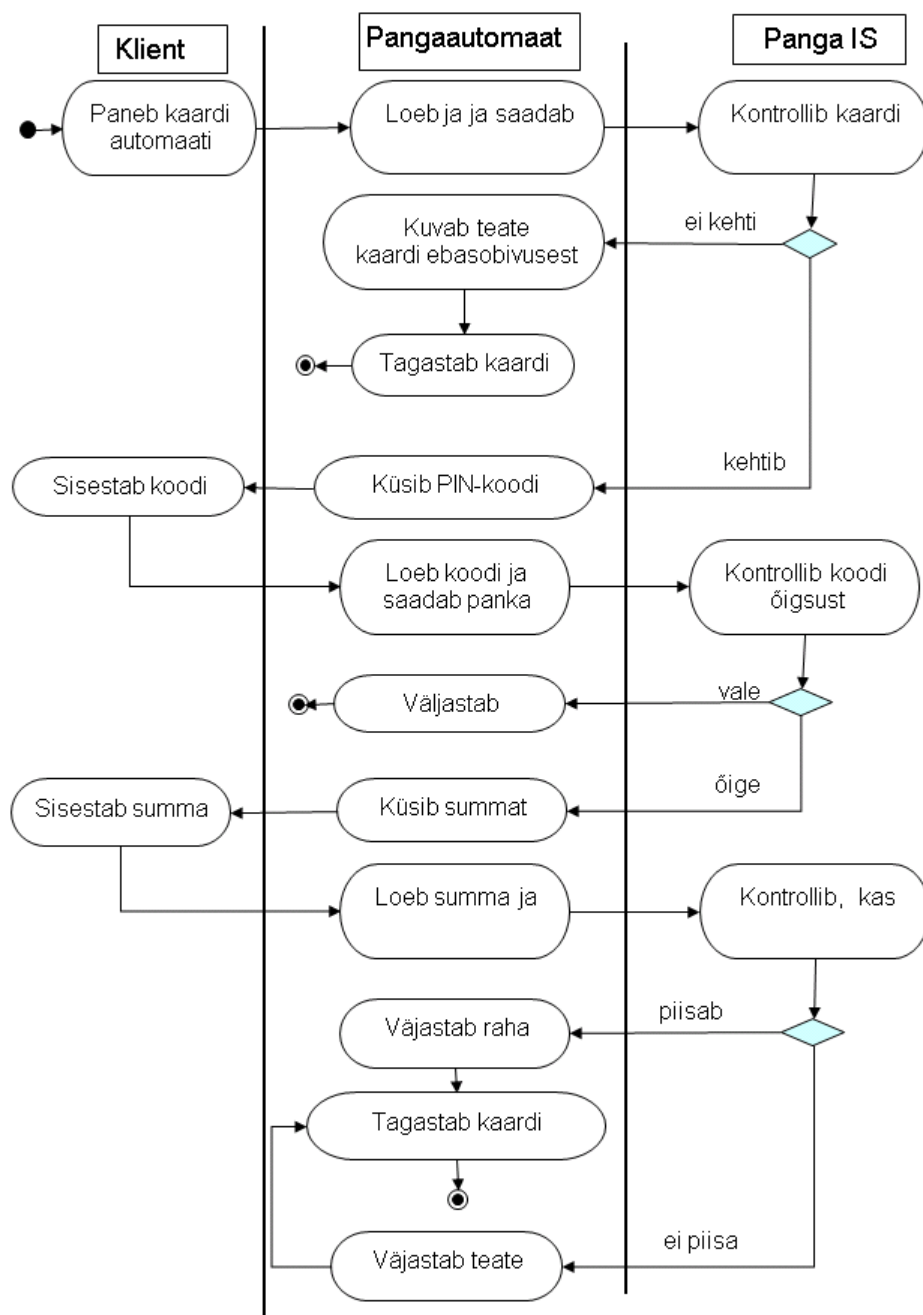
## Pangaautomaat

Klass ja erineva täpsusega tegevusskeemid:

| Pangaautomaat                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------|
| <b>kood</b><br><b>asukoht</b><br><b>raha</b><br><b>raha_max</b><br>...                                                      |
| <b>sisesta kaart(nr)</b><br><b>loe_PIN(k)</b><br><b>loe_raha(S)</b><br><b>anna_raha(S)</b><br><b>tagasta_kaart()</b><br>... |



## Tegevuste jaotus objektide vahel



## Pangaautomaat. PIN koodi kontroll

Lubatud on kolm katset:

