

# Scratch'is toetatavad programmeerimiskontseptsioonid ja -oskused

Interaktiivsete lugude, mängude ja animatsioonide loomise protsessis saab algaja teha Scratch'is endale selgeks mitmed olulised rakenduste loomise ja programmeerimise oskused ja kontseptsioonid.



## Probleemide lahendamise ja projektide disaini oskused



- loogiline ja algoritmiline mõtlemine
- süsteemne lähenemine probleemide lahendamisele
- ideede arendus alates lähtekontseptsioonist kuni projekti lõpplahenduseni
- kasutajaliideste loomise oskused ja kogemused
- silumise ja testimise vilumus
- keskendumisvõime ja visaduse arendamine

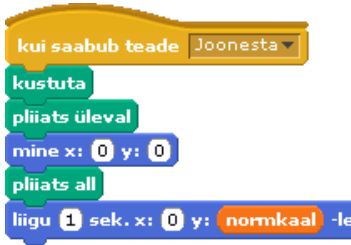


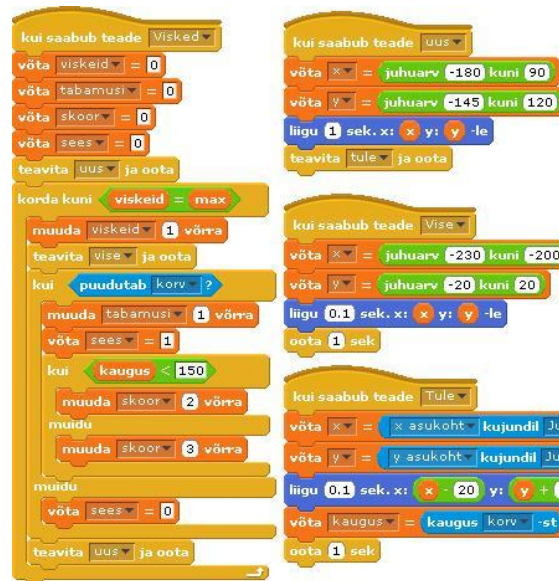
## Fundamentaalsed ideed arvutitest ja programmeerimisest

- programm ütleb arvutile täpselt, samm-sammult, mida teha
- programmide koostamine ei nõua spetsiaalseid teadmisi, vaid selget ja loogilist mõtlemist

## Rakenduste loomise ja programmeerimise põhikontseptsioonid ja põhimõisted

Kontsept	Selgitus	Näide
<b>Kasutajaliides</b>	<p>Kasutajaliides sisaldab vahendeid, mille abil saab kasutaja suhelda programmiga: anda vajalikke korraldusi ja näha nende täitmise tulemusi, muuta algandmeid jms.</p> <p>Scratch'is luuakse liides <b>laval</b>. Selle elementideks võivad olla taustad, aktiivsed ja passiivsed spraidid, käsunupud, klahvid, muutujate monitorid jms.</p> <p>Teistes süsteemides kasutatakse selleks vorme, ohjureid, dialoogibokse jms.</p> <p>Töö programmiga võib toimuda Scratch'i enda keskkonnas, brauseris Java appletiga või Flash'i filmiga.</p>	 <p><a href="#">Näita</a></p>
<b>Programm. Laused ja plokid. Programmiüksused: protseduurid ja skriptid</b>	<p>Programm on <b>lausete</b> (korralduste, käskude) kogum, määramiseks, milliseid <b>tegevusi</b> peab arvuti täitma <b>andmete</b> ja/või <b>objektidega</b>, tagades ka <b>kasutajaliidese</b> töö.</p> <p>Igas keeles on piiratud valik lauseid, mille esitamiseks on kindlad reeglid. Scratch'is kujutavad <b>käsud</b> (laused) endast graafilisi <b>plokke</b>, mis on jagatud otstarbe alusel gruppidesse: <b>Liikumine, Juhtimine</b> jms. Plokiga on lause süntaks määratletud üheselt ja süntaksivigade tekkimine on peaaegu võimatu.</p> <p>Enamikus keeltes võib programm koosneda mitmest üksusest: <b>protseduurid</b>, funktsioonid, skriptid jms.</p>	 <p>Programm koosneb neljast skriptist. Kaks on spraidi Kraps omad, Jukul ja piigal on üks skript.</p> <p>Kui klõpsatakse rohelist lippu, algab Krapsu</p>

	<p>Scratch`is nimetakse programmiüksusi <b>skriptideks</b>. Iga skript on seotud ühe kindla spraidiga, määrates selle tegevusi.</p> <p>Ühel spraidil võib olla mitu skripti. Skript saab käivitada (öeldakse ka pöörduda, kutsuda välja) teisi skripte, mis kuuluvad samale või teistele spraitidele. Pöördumiseks kasutakse plokkide <b>teavita nimi</b> või <b>teavita nimi ja oota</b>.</p>	<p>esimese skripti täitmine. Peale ”teretamist” käivitab antud skript käsuga <b>teavita start ja oota</b> korraka Juku ja piiga skriptid, mis algavad plokkidega <b>kui saabub teade start</b>, ning ootab kuni nende täitmine jõuab lõpuni. Seejärel jätkub Krapsu skripti täitmine. Kraps liigub lava keskpunkti ja siis käivitatakse sama spraidi teine skript. Peale selle täitmist, jätkatakse esimese skripti täitmist, mis viib Krapsu ette antud punkti.</p> <p><a href="#">Näita</a></p>
<p><b>Objektid (spraidid). Objektide Omadused, meetodid ja sündmused</b></p>	<p>Scratch`is on kesksel kohal <b>graafilised objektid</b>, mida nimetakse <b>spraitideks</b> (sprite) ning nende kostüümid. Objektideks on samuti ka <b>lava</b> ja selle taustad.</p> <p>Kuigi Scratch ei ole formaalselt objektorienteeritud süsteem, on sageli otstarbekas seda käsitleda, tuginedes objektorienteeritud lähenemisviisile.</p> <p>Iga objektiga on seotud teatud valik <b>omadusi</b>: nimi, asukoht laval (x-y), suurus, värvus jms ning <b>meetodeid</b>, mille abil määratakse tegevusi antud tüüpi objektiga: asukoha muutmine, pööramine, suuruse ja värvuse muutmine jms. Käsuplokkid sisuliselt vastavad meetoditele.</p> <p>Objekt võib reageerida <b>sündmustele</b> nagu nt hiireklõps, vajutus etteantud klahvile, kokkupuude teise objektiga...</p>	<p>Skripti täitmine algab, kui klõpsatakse spraiti Kraps (reaktsioon sündmusele). <a href="#">Näita</a></p>  <p>Skripti plokkid muudavad objekti (spraidi) omadusi: suund, asukoht, suurus, värvus, keere.</p>
<p><b>Andmete liigid</b></p>	<p>Scratch`is saab kasutada märk-, graafika- ja heliandmeid.</p> <p><b>Märkandmed</b>: arvud ja tekstid ehk stringid. Neid saab kasutada paljudes plokkides (käskudes) konstantidena, muutujatena ja loendite elementidena. Väärtusi saab leida (tuletada) avaldiste ja funktsioonide abil.</p> <p><b>Graafikaandmed</b> võivad esineda rakendustes kahes variandis:</p> <p><b>Spraidid</b> ja lava <b>taustad</b> – imporditakse või luuakse graafikarektori abil. Käskude abil saab määrata erinevaid tegevusi nendega.</p> <p><b>Pliiatsi</b> käskudega tehtavad <b>joonised</b></p>	 <p>Skriptid demonstreerivad erinevat liiki andmete kasutamist. Märkandmed (arvud ja stringid) on kasutusel konstantide ja muutujatena.</p> <p><a href="#">Näita</a></p>

	<p><b>Heliandmed.</b> Saab kasutada erinevaid vahendeid helide tekitamiseks (plokid <b>mängi nooti...</b>, <b>mängi trummi...</b> jms), importida ja lindistada heliklippe: kõne, muusika jms.</p>	  <p>Graafikaandmeid esindab sprait Kraps ja pliiatsiga tehtav joon spraidi liikumisel. Heliandmetega on tegemist plokkide <b>mängi heli ...</b> ja <b>mängi trummi...</b> kasutamisel.</p>
<p><b>Andmete organisatsioon</b></p>	<p>Organisatsiooni järgi eristatakse kõikides programmeerimiskeeltes <b>skalaarandmeid</b>: konstandid ja muutujad  <b>struktuurandmeid</b>: tabelid, massiivid, loendid jm  <b>Konstantide</b> väärtused esitatakse vahetult programmis. Ülaltoodud skripti näites on mitu arv- ja stringkonstanti. Need on plokkide väljades asuvad konkreetset arvud: 2, 100, 58, 0.2, 0 jne; ja tekstid: "Tere! Mina olen Kraps!", "Mis on Sinu nimi?" jne.  <b>Muutujad</b> esitatakse programmis nimede abil, nende väärtusi programmis ei näe – väärtused tekivad tavaliselt programmi täitmise ajal. Näiteks muutuja <i>pikkus</i> väärtus tekib sisestamisel käsuga <b>küsi</b>, muutuja <i>normkaal</i> väärtus tekib arvutuste tulemusena (<i>pikkus</i> – 100). Muutujate olemusest ja käsitlemisest vt allpool.  Tabelid, massiivid, <b>loendid</b> kujutavad endast kindla struktuuriga väärtuste kogumeid. Scratch'is saab kasutada <b>ainult loendeid</b>, mis on massiivide erijuht (vt allpool)</p>	
<p><b>Muutujad ja omistamine</b></p>	<p>Muutuja on <b>mälupesa</b>, kuhu programm saab salvestada väärtusi: arve ja stringe ning kasutada (lugeda) neid hiljem näiteks uute väärtuste leidmiseks.  Scratch'is saab muutujaid luua ja kasutada grupis <b>Muutujad</b> olevate plokkide abil. Loomisel saab muutujale anda <b>nime</b>, mida kasutatakse käskudes viitamiseks tema jooksvale väärtusele. Loomisel saab ka määrata, kas muutuja on kättesaadav kõikidele spraitidele (globaalne muutuja) või ainult ühele konkreetsele spraidile (lokaalne muutuja).  Muutujatele väärtuste omistamiseks ja väärtuste muutmiseks kasutatakse plokkide</p> <p><b>võta muutuja = avaldis</b> ja  <b>muuda muutuja avaldis võrra</b></p> <p>Muutuja väärtust saab kuvada laval nn <b>monitori</b> abil . Väärtust saab muuta ka "käsitsi" <b>liuguriga</b>.</p>	<p>Programm imiteerib korvpalli viskeid</p>  <p>Näited: <a href="#">Korvpall</a> ja <a href="#">Ideaal</a></p>

**Loendid (massiivid)**

Scratch'i loend vastab enam-vähem teistes keeltes kasutatavatele dünaamilistele ühemõõtmelistele massiividele. Taoline massiiv kujutab endast järjestatud mälupeade (elementide) kogumit. Elementidele saab viidata massiivi nime ja indeksite abil.

Scratch'is saab luua ja kasutada loendeid grupi **Muutujad** plokkidega. Viimaste abil saab lisada elemente loendi lõppu ja vahele, asendada ja eemaldada elemente, viidata elementidele jms.



Skript keskmise hinna leidmiseks



[Näita](#)

**Avaldised, tehted ja funktsioonid**

Avaldise abil saab anda eeskirja vajaliku väärtuse leidmiseks. Avaldiste loomiseks kasutatakse Scratch'is grupi **Tehted** plokkide.

Avaldis koosneb operandidest ja tehetest.

Operandideks võivad olla konstandid, muutujad, funktsioonid ja loendite elemendid.

Sõltuvalt tehetest võib avaldised jagada järgmistesse rühmadesse:

- arvavaldised: +, -, \*, /
- stringavaldised: ühenda, eralda
- võrdlused: <, =, >
- loogikaavaldised: **ja**, **või**, **mitte**

Avaldistes saab kasutada mitmeid funktsioone: **abs** (absoluutväärtus), **sqrt** (ruutjuur), **sin**, **cos**, **asin**, **log**, **ln** ...

Sisuliselt on funktsioonideks ka eraldi esitaud plokid; **mod** (jääk), **ümarda** ja **juhuarv**.

**NB!** Sulgusid tehete järjekorra määramiseks Scratchis kasutada ei saa. Iga tehete ploki võib lugeda sulgudes oleva avaldise liikmeks

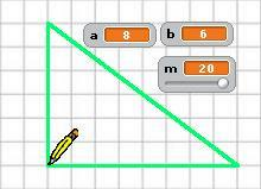
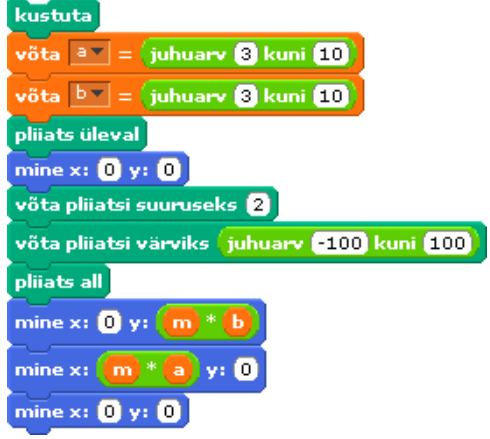
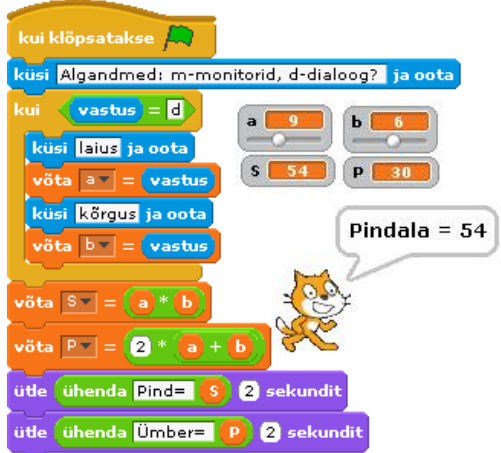









$$c = \frac{a+13}{\log |b|} \quad c \text{ ümardakse kahe kohani peale koma}$$







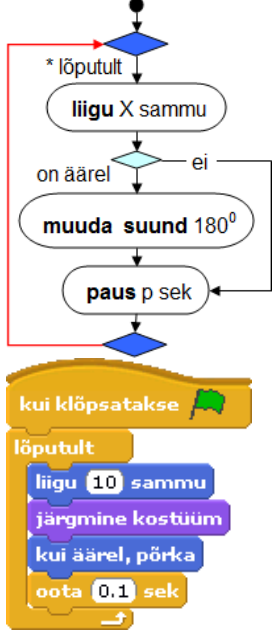
palli asukohta laval muudetakse juhuarvude abil

Näide [Ideaal](#)

<p><b>Joonistamine, joonestamine</b></p>	<p>Enamikus programmeerimiskeeltes on olemas vahendid, mis võimaldavad luua programmil täitmise ajal jooniseid. Scratchis on peamised vahendid selleks gruppides <b>Pliiats</b> ja <b>Liikumine</b>. Kõrvalolev skript joonistab täisnurkse kolmnurga, mille külgede pikkused <b>a</b> ja <b>b</b> tekitatakse juhuarvude abil. Täisnurk pannakse lava keskele punkti (0,0). Muutuja <b>m</b> abil määratakse mastaap: punktide arv ühes pikkuse ühikus.</p> 	 <p>Demo <a href="#">Rist Ring</a></p>
<p><b>Märkandmete sisestamine ja väljastamine</b></p>	<p>Ülesannete lahendamisel on sageli programmil vaja saada algandmeid ja peegeldada tulemusi. Esimesel juhul räägitakse andmete <b>sisestamisest</b> (lugemisest), teisel juhul – andmete <b>väljastamisest</b> (kirjutamine, kuvamine). Scratch'is saab andmete sisestamiseks kasutada muutujate liuguritega monitore (ainult arvud) ja plokki <b>küsi</b>, mis võimaldab sisestada arve ja tekste dialoogirežiimis. Tulemusi saab väljastada muutujate ja/või loendite <b>monitoride</b> või plokkidega <b>ütle</b>.</p>	 <p>Näide <a href="#">Ideaal</a></p>
<p><b>Protsesside juhtimine</b></p>	<p>Väga tähtsal kohal ülesannete lahendamisel on tegevuste täitmise järjekorra määramine. Just selles seisnebki programmeerimise kunst. Vajalikud tegevused ja nende järjekord ei sõltu üldiselt kasutatavast keelest, vaid tulenevad ülesande olemusest ja selle lahendamise <b>algoritmist</b>. Võib eristada nelja liiki protsesse: <b>jada</b> ehk järjestikune protsess, <b>kordus</b> ehk tsükliline protsess, <b>valik</b> ehk hargnev protsess ning <b>paralleelne protsess</b>. Programmeerimiskeeltes on olemas vahendid erinevat liiki protsesside kirjeldamiseks.</p>	
<p><b>Jada</b> ehk järjestikune protsess</p>	<p>Programmi koostamisel peab arvestama, et plokkide poolt määratud tegevusi täidetakse kindlas järjekorras. Lihtsamal juhul on tegemist järjestikuse protsessiga, kus plokkide täidetakse järjest ülevalt alla.</p>	

<p><b>Kordus</b> ehk tsükliline protsess</p>	<p>Korduste kirjeldamiseks on Scratch'is, nagu ka teistes keeltes, mitu lauset (plokki): <b>lõputult</b>, <b>korda n</b>, <b>korda kuni tingimus</b>, <b>lõputult kui tingimus</b>.</p> <p>Ploki <b>lõputult</b> sees olevaid plokkide täidetakse põhimõtteliselt lõpumatult. Skripti töö saab katkestada näiteks punase nupuga. Korduse sees võib olla üks või mitu plokki <b>kui tingimus</b>, mis tingimuse täitumisel katkestab vastavate plokkide abil skripti (plokk <b>peata skript</b>) või terve programmi töö (<b>peata kõik</b>).</p> <p>Käsu <b>korda n</b> täitmisel korratakse ploki sees olevaid plokkide <b>n</b> korda. Kordamiste arv (<b>n</b>) võib olla antud konstandi, muutuja või avaldise abil.</p> <p>Käsu <b>korda kuni tingimus</b> täitmisel korratakse ploki sees olevaid plokkide seni kuni tingimus saab tõeseks.</p> <p>Käsu <b>korda kui tingimus</b> täitmisel korratakse ploki sees olevaid plokkide nii kaua, kui tingimus on tõene</p>	 <p>Esimese ploki toimetel "jalutab" sprait lõpumatult laval edasi-tagasi. Katkestada saab punase nupu abil. Teise ploki täitmisel liigub sprait vasakult paremale. Liikumine (kordamine) lõpeb, kui spraidi x-koordinaat saab suuremaks 200st,</p>  <p>Objekt (näiteks kiisu) "pöristab" trummi ja teeb ringi (360°) laval.</p> <p><a href="#">Näita</a></p>  
<p><b>Valik</b> ehk hargnev protsess</p>	<p>Plokkidega <b>kui tingimus</b> ja <b>kui tingimus muidu</b> saab kirjeldada valikuid ehk hargnevaid protsesse.</p> <p>Lause <b>kui ...</b> korral (valik ühest), <b>kui tingimus</b> on tõene täidetakse ploki sees olevad plokkid, vastupidisel juhul jäetakse need vahele.</p> <p>Lause <b>kui ... muidu</b> korral (kahendvalik), <b>kui tingimus</b> on tõene täidetakse esimeses harus olevad plokkid, vastupidisel juhul teises harus olevad plokkid.</p>	  <p>Esimeses ploki kontrollitakse tingumust, kas x-koordinaat on suurem 200-st, kui jah, siis täidetakse sisemised plokkid. Teises skripti fragmendis kuvatakse teade, sõltuvalt sisestatud vanusest. Näide <a href="#">Ideaal</a></p>

<p><b>Harud</b> ehk paralleelsed protsessid</p>	<p>Kahte või enamat skripti (protsessi) saab täita samaaegselt ehk paralleelselt. Paralleelset täitmist saab määrata mitmel erineval viisil.</p> <p>Näiteks kõik skriptid, mille esimeseks plokiks on rohelise lipuga plokk, käivitatakse samaaegselt ja täidetakse paralleelselt, kui klõpsatakse rohelist lippu.</p> <p>Paralleelselt täidetakse ka skripte, mis algavad plokiga <b>kui saabub teade nimi</b>, milles on kasutusel sama <b>nimi</b>.</p>	 <p>Üks objekt liigub pidevalt mööda kolmnurkset trajektoori, teine – mööda ringikujulist.</p> <p>Kui saabub teade "korras", üks objekt teeb 3 hüpet, teine vahetab 8 korda kostüümi (näiteks tantsib)</p> <p>Projekt <a href="#">Tantsud</a></p>
<p><b>Sündmused</b></p>	<p>Objektid võivad reageerida teatud sündmustele: vajutus mingile klahvile, objekti klõpsamine hiirega, kokkupuude teise objektiga või lava servaga jms. Skriptides võib ette näha reaktsioone kindlatele sündmustele. Peamisteks vahenditeks sündmuste haldamisele on nn päiseplokkid:</p> <p><b>kui vajutakse klahvi ja kui klõpsatakse spraiti</b></p> <p>Skriptide sees leiab sageli kasutatust plokk <b>kui puudutab</b> {sprait   serv   kursor}</p> <p>Kui antud sprait puudutab teist spraiti, lava serva või hiirekursorit, tagastab plokk väärtuse <b>tõene</b>.</p> <p>Põhimõtteliselt on sündmusega tegemist ka rohelise lipu kasutamisel ja skriptide käivitamisel plokkide <b>teavita</b> ja <b>kui saabub teade</b> abil (vt allpool).</p>	 <p>Kui klõpsatakse rohelist lippu, võetakse muutujate väärtuseks 0 ja pall algseisu</p> <p>Kui vajutakse tühikuklahvi, muudetakse juhuarvudega palli asukohta ja suurendatakse löökide arvu ühe võrra. Kui pall puudutab väravat, lisatakse 1 muutujale <b>tabas</b>. Proovige!</p>
<p><b>Dünaamiline interaktsioon</b></p>	<p>Objekte saab mõjutada nõrealajas hiire kursoriga (hiire liigutamisega), heliga jm., kasutades grupi <b>Andurid</b> plokkide <b>hiire x</b>, <b>hiire y</b>, <b>helitugevus</b> jm</p>	 <p>Kui klõpsatakse lippu, liigub objekt horisontaalsuunas koos hiirekursoriga ja muudab keerme suurst, vastavalt y-i väärtusele. Proovige!</p>

<p><b>Koostöö spraitide ja skriptide vahel</b></p>	<p>Kui programm koosneb mitmest skriptis tekkitab sageli vajadus nende töö koordineerimiseks ja sünkroniseerimiseks. Üks sprait võib pöörduda teiste poole, need omakorda järgmiste poole jne. Skriptide koostöö korraldamisel kasutatakse plokkide</p> <ul style="list-style-type: none"> <li><b>teavita teade</b> ja</li> <li><b>teavita teade ja oota</b> ning</li> <li><b>kui saabub teade</b></li> </ul> <p>Siin vaadeldav programm koosneb neljast skriptist. Rohelise lipuga skript on <b>peaskript</b>. Sellest algab programmi töö ning ta käivitab teisi. Kui kasutaja sisestab tähe "d", käivitatakse skript <b>Loe</b> ning peale selle töö lõppu pannakse tööle skript <b>Arvuta</b>. Kui sisestati d-st erinev väärtus, käivitatakse kohe <b>Arvuta</b>.</p>	 <p>Ootamata <b>Arvuta</b> töö lõppu pannakse tööle ka skript <b>Joonesta</b> (ei ole siin näidatud, on analoogiline skriptiga punktis <b>Joonestamine</b>). <a href="#">Näide</a>.</p>
<p><b>Algoritmimine</b></p>	<p>Algoritmi all mõistetakse eeskirja, mis määrab, milliseid tegevusi ja millises järjekorras tuleb täita antud ülesande lahendamiseks või töö tegemiseks. Vajalikud tegevused ja nende täitmise järjekord ei sõltu üldiselt konkreetsest keelest vaid ülesandest või mudelist. Tegevused algoritmis esitatakse üldisemal kujul. Viimasel ajal kasutatakse algoritmide esitamiseks sageli modelleerimiskeelt UML.</p> <p>Scratch'i skripte võib vaadelda kui ühte algoritmide esitusviisi.</p>	 <p>Demod: <a href="#">Jalka</a>, <a href="#">Ruutvõrand</a>, <a href="#">Maks_Vek</a></p>



## Modelleerimine

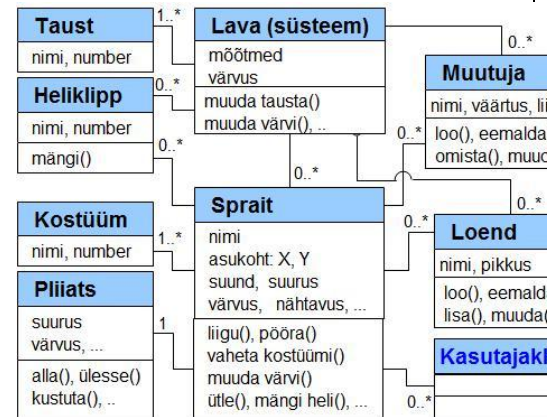
Scratch'i vahendeid saab kasutada objekt-orienteeritud modelleerimise põhimõistete selgitamiseks ja illustreerimiseks.

Scratch'i spraidid kuuluvad ühte universaalsesse **klassi** – **Sprait**. Spraitidel kindel valik **omadusi**: nimi, asukoht, suurus, ... ja **meetodeid**: liigu(), pööra(), muudaX(), ... Viimased esitatakse käskude ehk plokkide abil

Käskudest (baasmeetoditest) moodustatud skripte võib käsitleda kasutaja **alamklassidena**. Alamklassidena on käsitletava ka spraidiga seotud kostüümid, pliats, muutujad, loendid.

**Lava** on vaadeldav nõ peaklassina – klass **Sprait** on selle alamklass. Kõik spraidid asuvad laval.

Klassi **Lava** võib vaadelda süsteemi või projekti nõ esindajana.



Näide. Projekt [Tantsud](#).