

Rakenduste loomise ja programmeerimise alused

Kersti Antoi



TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY



Kursuse korraldus (1)

- 9 õppesessiooni (kokku 36 h)
- iseseisev töö
- Lõpetamisel TTÜ tunnistus
 - võimalik aine arvestada oma koolis
- Edaspidi võimalik jätkukursus
Programmeerimine Pythonis (edasijõudnutele)



Moodle õpikeskkond

moodle.taltech.ee

1. luua kasutajakonto (kui juba olemas, logida sisse)
 - vajalik on toimiv e-posti aadress
 - küsitakse isikukoodi
2. registreeruda kursusele

IDK0017 Rakenduste loomise ja programmeerimise alused (TTÜ tehnoloogiakool) (IDK0017g)

Registreerumise võti: 2023k
(tehnoloogiakool)

Moodle kursusel on:

- lingid materjalidele
- harjutused, ülesanded, testid
- foorumid, küsimustikud
- jooksev info



Kursuse korraldus (2)

- Kursuse lõpetamiseks on vajalik
 - kontakttundidest osavõtt
 - iseseisev töö materjalidega tutvumisel
 - koduülesannete (tähtaegne) esitamine
 - testide sooritamine (parim tulemus min 85%, teste saab teha korduvalt)
 - tagasiside andmine:
jooksvalt ja kursuse lõpus



Arvutiklassid

- TTÜ arvutiklass ICT-401
 - logida sisse kasutajaga **Koolitus1**
parool: Tehnoloogia01
- ühiskasutuses ketas K:
 - teha kaust enda failide salvestamiseks
(oma failid salvestatakse ainult K-kettale)
 - võimalik failide jagamine teiste õppuritega
- Loodud failid võiks iga õppesessiooni lõpus
 - kopeerida mingile oma andmekandjale
 - saata endale e-mailiga
 - kopeerida Moodles alale „privaatsed failid“



Programmeerimisvahendid / -keskkonnad

- [Scratch](http://scratch.mit.edu), graafiline keel
loodud programmeerimise õpetamiseks algajatele
<http://scratch.mit.edu>
 - mõisted
 - tüüpalgoritmid
- Programmeerimiskeel **Python**
 - andmed, muutujad ja loendid
 - põhikäsud
 - funktsioonid
 - moodulid
 - joonistamine (kilpkonnagraafika *Turtle*)
 - tekstifailid



Mõisteid

rakendus — tarkvara kogum antud liiki ülesannete lahendamiseks või tegevuste täitmiseks

modelleerimine — süsteemi struktuuri, käitumise ja oleku kirjeldamine ja mudelite loomine

mudel — reaalse või abstraktse süsteemi lihtsustatud esitus, peegeldab ainult süsteemi olulisi elemente ja tegevusi (antud probleemi seisukohast)

algoritm — täpne ja ühemõtteline eeskiri antud liiki ülesannete lahendamiseks või tegevuste täitmiseks kindla eesmärgi saavutamisel

rakendus, andmed, programm, kasutajaliides, ...

objekt, omadus, meetod, sündmus, valik, kordus, ...

muutuja, loend, massiiv, skoop, operatsioon, omistamine

protseduur, funktsioon, parameeter, moodul, ...



Scratch

- graafiline programmeerimissüsteem [Scratch](http://scratch.mit.edu) on loodud Massachusettsi Tehnoloogiainstituudi (MIT) meedialaboris *Lifelong Kindergarten group* 2007. a <http://scratch.mit.edu>
- mõeldud programmeerimise õpetamiseks algajatele, aga ajaviiteks kasutavad ka paljud profid
- kasutusel paljudes koolides (ka ülikoolides).
- Scratchi edasiarendused:
 - BYOB (Snap! *University of California at Berkeley*) <http://byob.berkeley.edu>



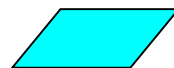
Scratch

- Scratch 1.4 (2009)
http://scratch.mit.edu/scratch_1.4
- Scratch 2.0 (kevad 2013)
 - veebiversioon
 - offline-versioon<http://scratch.mit.edu/scratch2download>
- Scratch 3.0 (kevad 2019)
 - veebiversioon
 - offline-versioon<https://scratch.mit.edu/download>



Objektid

- **Objekt** on teatud osa vaadeldavast süsteemist, mida võib käsitleda antud kontekstis tervikuna ning mille kohta säilitakse andmeid antud rakenduses.
- Objekt võib olla konkreetne ese, isik, graafiline kujund, nähtus, isiku auto, isiku arvuti, arvuti klassis, jmt:



- Igale objektile vastab teatud valik **omadusi** ja **tegevusi** (objekti meetodeid)
- objektidega on võimalik sooritada teatud tegevusi
- Objektid võivad reageerida teatud sündmustele
- Iga objekt on tavaliselt seotud ühe või mitme selle süsteemi objektiga



Objektide omadused ja meetodid

- **Omadused** (atribuudid) on karakteristikud, mis identifitseerivad objekti. Näiteks iseloomustavad objekti olekut, väljanägemist jmt
- **Meetodid** on tegevused (operatsioonid), mida täidab objekt ise või mida täidetakse objektiga

Omaduste ja meetodite valik sõltub rakenduse (süsteemi) ja objekti liigist, eesmärkidest ulatusest jms.



Klassid ja objektid

- **Klass** – abstraktsioon: ühetüübiliste objektide kirjeldus vaadeldavas süsteemis
- Ühte **klassi** kuuluvatel **objektidel** on ühesugune tähendus, ühesugused omadused, meetodid ning seosed teiste objektidega
- Diagrammidel (UML) on esitatud klass ristkülikuna, üldjuhul kolm sektsiooni:
 - **nimi** (kohustuslik)
 - **atribuudid (omadused)**
 - **operatsioonid (metodid)**
- Konkreetne **objekt** on klassi eksemplar (ilming)

Objektiklass

omadused

.....

metodid

.....



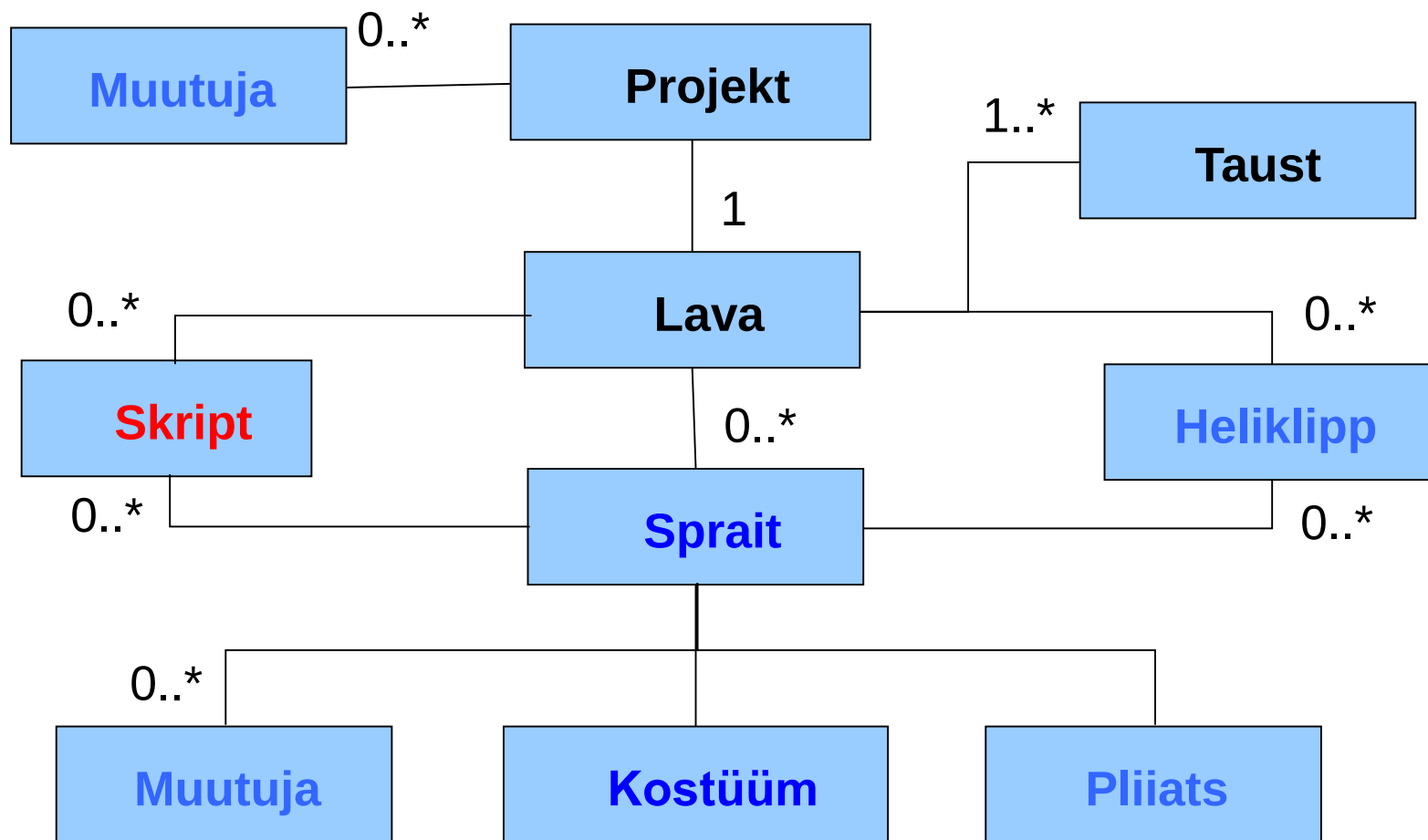
Scratchi põhimõisted (objektid)

- Sprait (*Sprite*)
- Skript (*Script*)
- Lava (*Stage*)
- Kostüüm (*Costume*)
- Heli (*Sound*)
- Sündmus (*Event*)
- Projekt (*Project*)



Scratchi projekti skeem

(seosed objektiklasside vahel)





Objektid Scratchis

Sprait (*Sprite*) –

Scratchi universaalne graafikaobjektide klass

- Spraitide omadused:
nimi, asukoht, suurus, kostüüm, nähtavus jm
- Spraitidele saab määrata tegevusi käskude ehk meetodite abil:
liiguta, muuda suurust, jälg jmt
- Käskudest moodustuvad programmiüksused ehk **skriptid**
- Klassiga **sprait** on seotud klassid:
kostüüm, pliiats, heliklipp, muutuja, ...

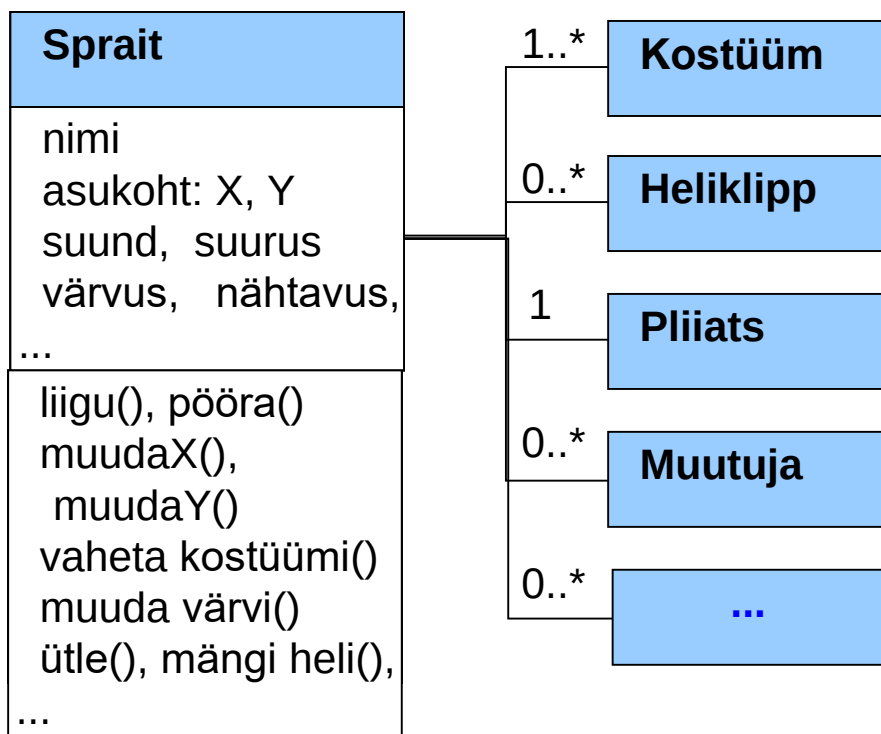
Lava on ekraani piirkond, kus spraidid asuvad ja tegutsevad



Spraidid

Sprait (*Sprite*) on graafikaobjekt, millele programm (skript) määrab tegevusi.

Spraidi põhiomadused ja tegevused:





Kostüümid

Kostüümid (*Costume*) on spraidi kuvandid laval, mis erinevad üksteisest suuremal või vähemal määral.

Kostüüme saab kasutada näiteks animatsioonide loomisel.





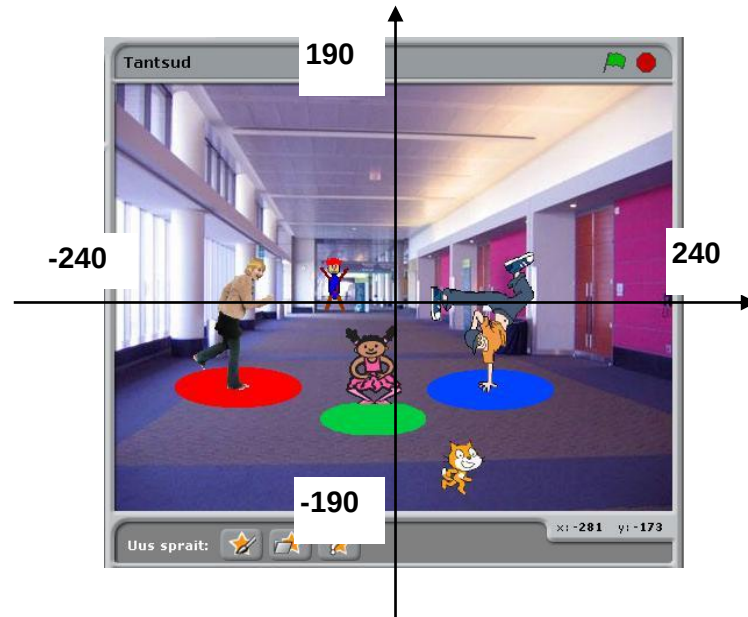
Lava

Lava (*Stage*) on ekraani piirkond, kus spraidid tegutsevad. Laval võib olla üks või mitu tausta (pilti), mida saab vahetada. Lava võib näidata mitmes erinevas suuruses.

Lava koordinaadid

X: -240 kuni 240

Y: -190 kuni 190





Skriptid

- **Skript** – Scratchi programmi põhiüksus; programm võib koosneda ühest või mitmest skriptist
- Skript koosneb **käskudest** (gr. plokkidest ehk spraidi meetoditest), pannakse kokku hiire abil
- Käsud jagunevad gruppidesse:
 - **liikumine**, **välimus**, **heli**, **juhtimine** jms
- Skript on alati kindla **spraidi** (või **lava**) jaoks
- Skript saab käivitada teisi skripte, kui väljastab vastava teate
- Skriptide täitmine võib toimuda paralleelselt



Sündmused

- Võimalik on määrata, et objekt reageerib teatud sündmusele ja täidab etteantud tegevused.
- Sündmusteks võivad olla hiireklõps spraidil või laval, vajutus teatud klahvile, teade teiselt spraidilt jms.





Projektid

- Scratch'i rakendusi nimetatakse projektideks
- Projekti põhikomponentideks on
 - lava
 - spraidid
 - skriptid
- Lisaks
 - kostüümid
 - heliklipid
- Kogu projekt salvestatakse ühes failis
- Projekti saab täita Scratch'i kasutajaliideses, valides erineva suurusega lavade vahel
- Scratchi serveris asuva Scratch rakenduse saab sängitada veebidokumenti



Joonistamine Scratchis

Joonistamisplukkide lisamiseks rakendusele klõpsata Scratch akna all vasakul nurgas Lisa Laiendus (*Add Extension*) – Pliats (*Pen*)

- Kui pliiats on all, jätab sprait liikumisel jälje
- Sprait jätab jäljendi – käsk Stamp
- Joone laiust ja värvi on võimalik määrata ja muuta
- erinevad värvid 1..100, siis korduvad



Muutujad (1)

- **Muutuja** (*Variable*) on mälupea ehk -väli
- Igal muutujal on **nimi, väärtus, skoop** (mõjupiirkond)
 - Nimi on üldjuhul tähtede, numbrite ja allkriipsude jada (algab tähega)
 - muutujal saab igal ajahetkel olla ainult üks väärtus
 - Skoop määrab muutuja mõjupiirkonna:
 - **globaalne** – kättesaadav ehk nähtav kõikidele objektidele
 - **lokaalne** – kättesaadav ainult ühele objektile



Muutujad (2)

- Programm saab täitmise ajal
 - **salvestada** muutujasse (mäluvälja) väärtusi
 - **muuta** mäluväljas olevat väärtust
 - kasutada (**lugeda**) varem salvestatud uute väärtuste leidmiseks
- Väärtuse salvestamist muutujas (mäluväljas) nimetatakse **omistamiseks**
- Muutujale uue väärtuse omistamisel selle eelmine väärtus kaob – asendatakse uuega



Muutujad Scratch'is

- Muutuja loomine toimub rakenduse loomise faasis käsu **Tee muutuja** (*Make variable*) abil.
- Muutuja loomisel määratakse **nimi** ja skoop:
- Muutuja jooksva väärtuse saab lasta kuvada **Laval** vastaval monitoril
- Muutuja (arv)väärtust saab muuta ka liuguri abil, mille saab lasta kuvada muutuja monitori juures