WEBQUEST

Programmeerimine 7.-9. klassi matemaatika tundides

Index

[INTRODUCTION](#_heading=h.3dy6vkm) **2**

[Description](#_heading=h.1t3h5sf) **2**

[Tunni väljundid](#_heading=h.4d34og8) **2**

[AJAKAVA](#_heading=h.3rdcrjn) **3**

[ÜLESANDED/TEGEVUSED](#_heading=h.1ksv4uv) **4**

[**Programmeerimiskeel Python**](#_heading=h.44sinio) **4**

[Sissejuhatus ja ettevalmistus matemaatika ülesannete lahendamiseks (kokku 25 min)](#_heading=h.cpt2w5onmzho) 7

[Tegevus 1 - Pythoni tutvustamine ja algoritmid (45 min)](#_heading=h.z337ya) 8

[Tegevus 2 - Matemaatikaülesannete automatiseerimine (45 min)](#_heading=h.3j2qqm3) 8

[Tegevus 3 - Matemaatikaülesannete programmide katsetamine (45min)](#_heading=h.f1w4qta6e2vf) 9

[Kokkuvõte](#_heading=h.o92f2zphjhit) 10

[Protsess](#_heading=h.2xcytpi) **11**

[Vajalik materjal](#_heading=h.1ci93xb) 11

[Hindamine](#_heading=h.147n2zr) **12**

[Kokkuvõte](#_heading=h.32hioqz) **13**

[Omandatud tulemused](#_heading=h.1hmsyys) 13

# INTRODUCTION

# Description

Tunni teemaks on Python programmeerimisülesannete abil õppida matemaatikat ja loogilist mõtlemist. Tutvutakse lähemalt algoritmilise mõtlemisega ja Python programmeerimiskeelega. Tund viiakse läbi programmeerimise õpetaja ja 1-2 programmeerimise kursuse õpilasega.

# Tunni väljundid

* Õppetöö rikastamine programmeerimise ülesannetega.
* Õppeaine teema paremaks ja efektiivsemaks edasiandmiseks programmeerimiskeele Pythoni tutvustamine.

# AJAKAVA

Kestvus: 4\*45 minutit

| 25 min | Sissejuhatus |
| --- | --- |
|  |  |
| 45 min | Tegevus 1 - Pythoni tutvustamine ja algoritmid (45 min) |
| 45 min | Tegevus 2 - Matemaatikaülesannete automatiseerimine (45 min) |
| 45 min  15 min | Tegevus 3- Matemaatikaülesannete programmide katsetamine  Hindamine |
| 5 min | Kokkuvõte |
|  |  |
|  |  |
|  |  |
|  |  |

# ÜLESANDED/TEGEVUSED

# Programmeerimiskeel Python

Mingi ülesande lahendamiseks vajaliku algoritmi kirjapanek plokkskeemina võib küll olla arusaadav inimesele, aga see ei sobi hästi algoritmi arvutile arusaadavaks tegemiseks. Selleks otstarbeks tuleb lahendus esitada programmina (vt Eesti keele seletavast sõnaraamatust). Programm kirjutatakse mingis programmeerimiskeeles. Programmeerimiskeeli on olemas sadu, võib-olla tuhandeid. Nagu loomulikud keeled, on ka mõned programmeerimiskeeled omavahel sarnased, teised jällegi erinevad. Näiteks eesti keel on suhteliselt sarnane soome keelega, vene keel ukraina keelega ja rootsi keel norra keelega. Kuigi eesti keel ja katalaani keel ei ole väga sarnased, on neis mõlemas ikkagi nimisõnad, tegusõnad ja muud paljudele keeltele ühised omadused. Nii on ka erinevates programmeerimiskeeltes palju sarnaseid struktuure. Muu hulgas peab kuidagi olema võimaldatud, et: (https://web.htk.tlu.ee/digitaru/programmeerimine)

kui teatud tingimus on täidetud, siis tehakse ühtmoodi, kui pole täidetud, siis teistmoodi;

mingit tegevust tehakse teatud arv kordi;

mingit tegevust tehakse nii kaua, kui teatud tingimus on täidetud;

mingeid programmiosasid saab suhteliselt iseseisvalt kasutada.

Programmeerimiskeeli saab tinglikult jaotada põlvkondadesse. (Põlvkondadest ja teistest ajaloolistest asjadest on juttu silmaringimaterjalis.) Rõhuv enamik programme kirjutatakse tänapäeval mingis kõrgtasemekeeles. Neist levinumad on näiteks C (ka C++, C#), Java, PHP ja Python. Erinevate keelte kasutuslevik on pidevas muutumises ja seda on huvitav jälgida näiteks siin. (https://web.htk.tlu.ee/digitaru/programmeerimine)

Meie kursuse aluseks on programmeerimiskeel Python. Ühelt poolt on tegemist keelega, milles ka päriselt tööstuses programmeeritakse. Teiselt poolt on see sellistest keeltest sobivaim just esimeseks õpitavaks keeleks, sest lihtsaid programme saab kirjutada võrdlemisi väheste ridade abil. Programmeerimiskeele Python esimene versioon loodi 1989. aasta lõpus. Sellel kursusel kasutame versiooni Python 3. Täpsem versioon (nt 3.4.3 või 3.5.1) pole nii oluline, kuna meie jaoks on nende erinevused vähetähtsad. (https://web.htk.tlu.ee/digitaru/programmeerimine)

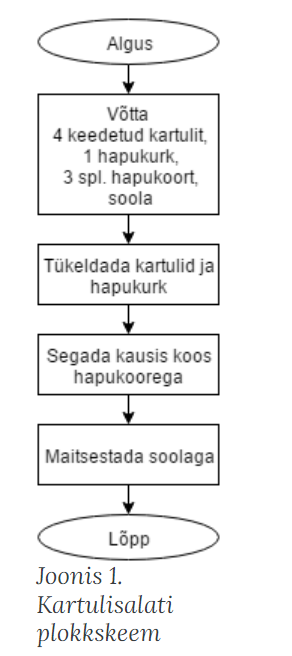
**ALGORITMID**

Kui me tahame teha midagi suuremat, siis võib tulemuse saavutamisel abi olla sellest, kui me kavandatu väiksemateks sammudeks jaotame. Seejuures peaksime muidugi mõtlema, mis järjekorras need sammud tuleb või saab teha. Kas mingeid samme saab teha samaaegselt? Kas midagi tuleb teha korduvalt? Kas mõnda asja tuleb või saab teha ainult teatud tingimustel? Äkki saab hoopis keegi teine meie endi asemel midagi ära teha? (https://web.htk.tlu.ee/digitaru/programmeerimine)

Kui tahta kellelegi teisele selgeks teha, kuidas teatud asi toimuma peaks, siis tuleks selleks kasutada mingit mõlemale poolele arusaadavat viisi. Vast on nii mõnedki kogenud, et täpsed juhised võivad kaasa aidata tulemuse saamisele: “Pööra ringteelt välja teiselt mahasõidult!”, “Keeda 20 minutit!”, “Lisa kaks labidatäit kruusa!”, “Istuge, palun!”. (https://web.htk.tlu.ee/digitaru/programmeerimine)

Kui soovime ümbritsevat maailma mingit moodi käituma suunata, siis hea meetod selleks on kehtestada eeskirjad. Eeskirjad võivad olla keelavad või kohustavad. Keelavad eeskirjad on näiteks sellised: “Ära mine vales kohas üle tänava!”, “Ära söö (joo) nii palju!”, “Ära ületa kiirust!”, “Ära mängi nii palju arvutimänge!”.(https://web.htk.tlu.ee/digitaru/programmeerimine)

Programmeerimises keskendume kohustavatele eeskirjadele – neile, mis kirjeldavad tegevusi, mida peab tegema. Tavaliselt on elus mingi ülesande lahendamiseks vaja teha mitu sammu järjest – sel juhul saame rääkida lahenduseeskirjast ehk algoritmist. (<https://web.htk.tlu.ee/digitaru/programmeerimine>)



## Sissejuhatus ja ettevalmistus matemaatika ülesannete lahendamiseks (kokku 25 min)

Õpetaja teeb õpilastega Pythoni programmeerimiskeele tutvustamiseks mõned lihtsamad programmeerimisnäited ja algoritmid. Materjal asub eelnevatel lehekülgedel 4-8.

**Õpilased harjutavad:**

• probleemide lahendamist;

• järjepidevust;

• täpsust;

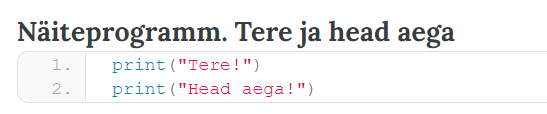
• põhjendamist

Iga õppeülesanne toetab matemaatika õppekava konkreetset teemat, kuid sõltuvalt koostatud programmist on võimalik ülesandeid lõimida ka teiste ainevaldkondadega: keel, loodus, keskkond vms.

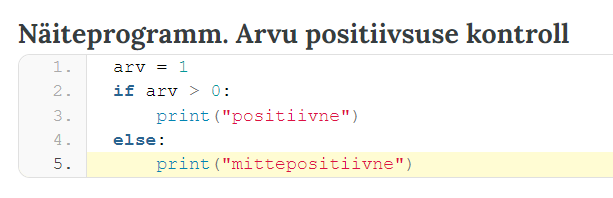
## Tegevus 1 - Pythoni tutvustamine ja algoritmid (45 min)

Antud ülesandes saame kasutada ainetevahelist lõimingut – informaatika ja matemaatika.

Materjal õpilastele tutvustamiseks asub lehekülgedel 4-8. Lahendatakse ka mõned näiteülesanded:   
1) Koostage programm, kus väljastatakse kasutajale Tere! ja järgmisel real Head aega!

  
Joonis 2. Esimene näiteprogramm. Allikas: autori erakogu

2) Koosta programm, kus kontrollitakse etteantud arvu, kas see on positiivne või negatiivne.

  
Joonis 3. Teine näiteprogramm. Allikas: autori erakogu

## Tegevus 2 - Matemaatikaülesannete automatiseerimine (45 min)

1. Koosta programm, mis kontrollib kasutaja poolt sisestatud arvu, kas see on positiivne või negatiivne. Positiivse arvu korral väljastatakse ekraanile teade “ Teie poolt sisestatud arv on positiivne” ja negatiivse arvu korral vastupidiselt eelmisele vastusele.
2. Koosta programm, kus kasutaja sisestab enda ema, isa ja enda vanuse ning programm leiab arvude summa. Kasutajale väljastatakse tema ja tema vanemate vanuste summa.
3. Koosta programm, kus kasutajalt küsitakse nime, keda ta soovib telekas tervitada. Televiisor (ehk arvuti) väljastab kasutajatele (ehk televiisori vaatajatele) nimelise tervitse koos tervitaja enda nimega, näiteks: Tere Mari! Sind tervitab Guido.

## 

## 

## 

## Tegevus 3 - Matemaatikaülesannete programmide katsetamine (45min)

Igat programmi antakse klassikaaslastele kontrollimiseks, kas programm töötab ja kas on arusaadav kõikidele klassikaaslastele sinu programmi töö ja selleks kogutud andmete vajalikkus. Ülesandeid võib katsetada ka tahvli peal läbi, kas siis läbi online Pythoni või siis läbi kirjutades või programmi Thonny´it kasutades.

## 

## Kokkuvõte

Õpilased omandavad järgmised oskused:

• probleemide lahendamine;

• järjepidevus;

• täpsust;

• põhjendamine.

# Protsess

## Vajalik materjal

1. Õpetaja poolt ettevalmistatud

* Esitlus programmeerimiskeele tutvustamiseks.
* Esitlus matemaatika ülesannetest, mida saaks automatiseerida.

1. Õpilase poolt vajalik

* Julgus eksida ja proovida
* Eelteadmised programmeerimisest pole vajalik

# Hindamine

**Matemaatika ülesannete automatiseerimine**

* **Hindamissüsteem:** Hinnatakse kas arvestatud “A” või mittearvestatud “MA”.
* **Hindamise kirjeldus:** Õpilased hindavad ennast ise, vastates küsimusele, kas nende koostatud programm töötab ja annab õigeid vastuseid. Õpilased esitavad valminud töö õpetajale ja õpetaja kontrollib lahenduskäike ja hindab õpilaste sooritusi “A” (alates 50%) või “MA” (alla 50%).

# Kokkuvõte

## Omandatud tulemused

Õpilased oskavad algeliselt kasutada Python programmeerimiskeelt.

Nad oskavad matemaatikaülesandeid automatiseerida.

Õpilased kinnistavad seni õpitut ehk matemaatikatunnis õpitud lahendusi.