

Programmeerimiskeel

Ivari Horm

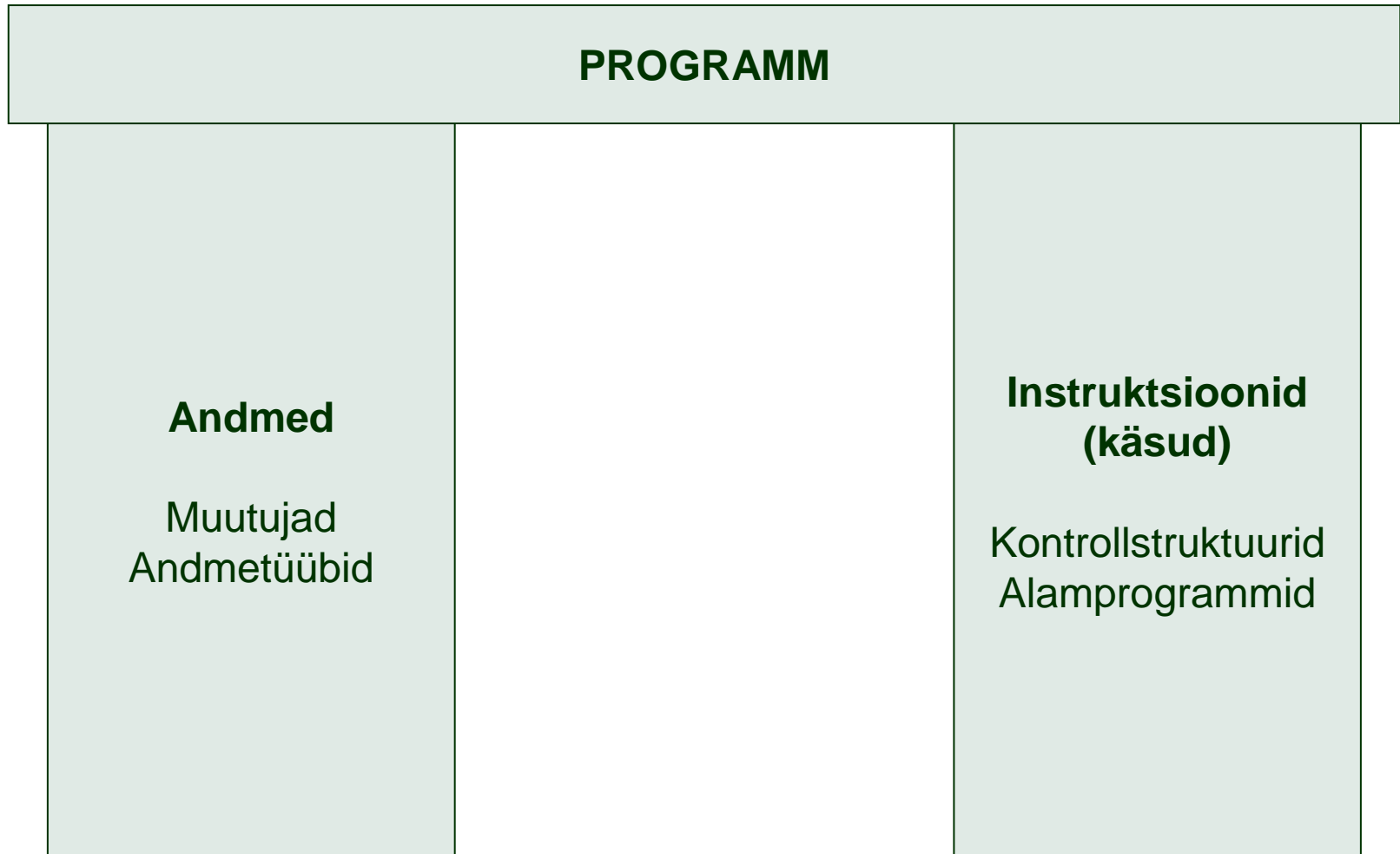
ranger@risk.ee

Programmeerimiskeele elemendid

- Paljud programmeerimiskeeled sarnanevad üksteisele
- Sisaldavad ühesuguseid elemente ja komponente

- **Süntaks**
Kuidas antud keeles programmikoodi kirjutada tuleb
- **Semantika**
Mida korrektse süntaksiga programmikoodi tulemusena arvuti teeb
- **Pragmaatika**
Kirjeldab keele kasutatavust, rakendusvaldkondi, efektiivsust

- Kokkuleppeline **väljamõeldud** “programmeerimiskeel” erinevate programmeerimiskeele elementide kirjeldamiseks
- Metakeel **ei ole seotud** ühegi konkreetse programmeerimiskeelega
- Metakeel **sarnaneb** paljudele programmeerimiskeeltele



- Mälupesa või mitmest pesast koosnev plokk, millele on antud nimi
- Nime abil on mälupesa lihtne programmikoodis kasutada
- Programmeerija ei pea täpselt teadma, kus pesa mälus asub



- Näitab, mis tüüpi andmeid muutujas hoida saab

Andmetüübid	
Täisarv	2, 5, 1000
Murdarv	3.14, 2.0
Täht	'a', 'B'
Tekst	"Hello"

- Iga programmeerimiskeel sisaldab elemente, et muutujates olevate andmetega tehteid teha ning andmeid muutujasse salvestada ja sealt kätte saada

Omistamine

```
ristkyliku_kylg1 = 12;
ristkyliku_kylg2 = 10;

pindala = ristkyliku_kylg1 * ristkyliku_kylg2;

print pindala;
```


- Programm on arvutile arusaadavate käskude jada
- Käskude täitmine toimub järgemööda, mingeid eelistusi ei tehta
- Programmi käskude täitmise järjekorra muutmiseks kasutatakse kontrollstruktuure

- Programmis kirjeldatud käskude grupp, mida keele süntaksi seisukohast saab vaadelda ühe käsuna
- Süntaksis lubatud ühe käsu saab asendada ploki

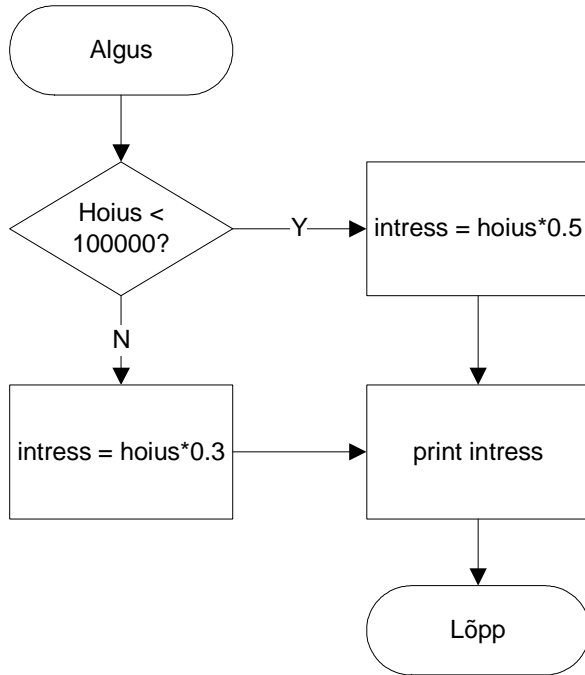
Ploki näide

```
begin
  ristkyliku_kylg1 = 12;
  ristkyliku_kylg2 = 10;

  pindala = ristkyliku_kylg1 * ristkyliku_kylg2;
end

print pindala;
```

- Programmeerimiskeele elemendid, mille abil saab muuta programmi täitmise järjekorda
- **Tsüklid**
Mõne programmi lõigu korduvaks täitmiseks
- **Hargnemised**
Arvuti poolt järgmisena täidetava programmi lõigu valimine etteantud võimaluste vahel, kontrollides enne valiku tegemist nn. “valikutingimust”



Hargnemise näide

```

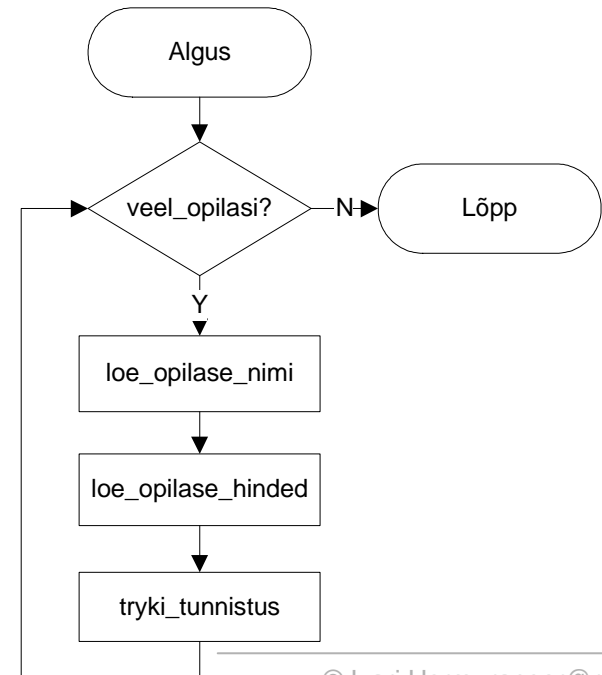
if (hoius < 100000) then
    intress = hoius * 0.05;
else
    intress = hoius * 0.03;
end if

print intress;
  
```

Tsükli näide

```

while(on_veel_opilasi) then
    loe_opilase_nimi;
    loe_opilase_hinded;
    tryki_tunnistus;
end while
  
```



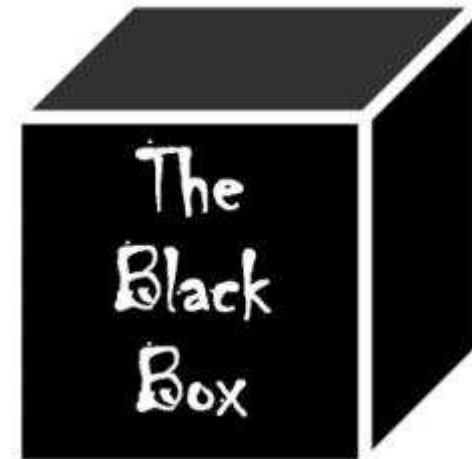
- Suuri programme on praktiliselt võimatu kirjutada ühtse tervikuna
- Alamprogramm sisaldab käske mingi kindla tegevuse sooritamiseks, mis on grupeeritud üheks tervikuks ning millele on antud nimi

Alamprogrammi näide

```
sub burn_man( )  
    get_material;  
    build_podium;  
    build_man;  
    put_man_onto_podium;  
    get_crowd;  
    fire_up_man;  
    be_happy;  
end sub  
  
[...]  
  
call burn_man( );
```



- Säästavad ühe ja sama programmi jupi mitmekordsest kirjutamisest
- Kui alamprogramm on valmis tehtud, piisab ainult tema nime teadmisesest, mis tema sees toimub, pole oluline
- Must kast



Programmi täitmine

Ivari Horm

ranger@risk.ee

- Arvuti sisselülitamisel käivitub operatsioonisüsteem
- Programmide täitmiseks tuleb anda operatsioonisüsteemile korraldus antud programm käivitada

- Arvuti käivitamisel laetakse operatsioonisüsteem “automaatselt” arvuti mällu
- Programmi käivitamisel laeb operatsioonisüsteem programmikoodi arvuti mällu
- Protsessor hakkab käsk-käsu järel programmis olevaid instruksioone täitma

- Kui programmis on kirjeldatud mingi muutuja, siis selleni jõudmisel reserveerib operatsioonisüsteem vajaliku hulga mälu
- Muutuja kustutamisel mälu **vabastatakse**

- Kõik programmi käsud asuvad arvuti mälus
- Kõik programmis sisalduvad muutujad asuvad mälus

- Töö lõppedes antakse programmi poolt hõivatud mälu teiste programmide ja andmete kasutusse
- Mälu **vabastatakse**, programmis olnud muutujate väärtused lähevad kaduma

- Programmi täitmisel ei saa ületada käsutuses oleva mälu mahtu
- Igale programmis esinevale suurusele eraldatakse automaatselt mälu
- Muutujale väärtuse andmisel kopeeritakse väärtus muutujaga seotud mäluväljale

- Muutujat saab programmis kasutada siis, kui talle on antud väärtus
- Iga andmetüübi jaoks on keeles kehtestatud minimaalne ja maksimaalne väärtus

- **Programmeerimine**
Rein Jürgenson, Tallinn 1989
- **Machine Code Introduction**
<http://c2.com/cgi/wiki?MachineCode>
- **Programmeerimise alused**
<http://www.aprote.ee/~a.truu/tk/prog.html>