

Programmeerimise alused

Ivari Horm

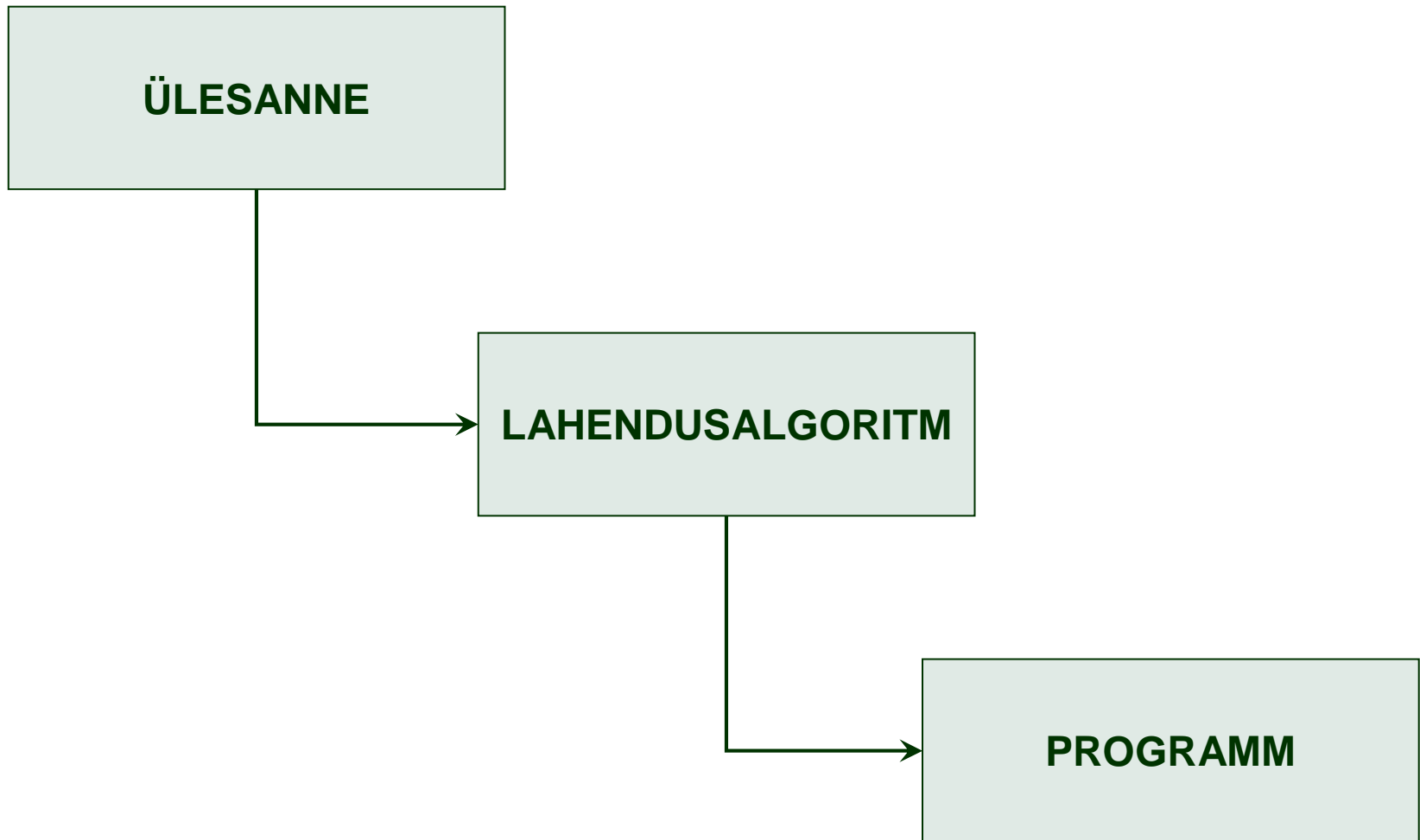
ranger@risk.ee

- Ettevalmistus
- Algoritmide näited
- Programmeerimine
- Programmi täitmine

Ettevalmistus

Ivari Horm

ranger@risk.ee



- Konkreetne ülesanne on vaja formaliseerida
- Välja töötada lahendusalgorithm
- Koostada algoritmi alusel programm

- Kirjeldab üheselt ära, mis on antud (lähtetingimused)
- Näitab, mida on vaja leida (tulemus)
- Töötab välja probleemi lahendamise (matemaatilise) eeskirja

- Leida ristküliku küljed, kui on teada selle ristküliku pindala ning üks külg on teisest n ühiku võrra lühem.

Formaliseerimine

- Tähistame ristküliku külgede pikkuse tähtedega a ja b .
- Tähistame ristküliku pindala tähega S .
- Ristküliku üldine pindala leidmise valem on $S=a*b$.
- Et $b=a-n$, siis antud juhul $S=a*(a-n) \Rightarrow S=a^2-na$.
- a leidmiseks tuleb seega lahendada ruutvõrrand $a^2-na-S=0$.
- Seejures $b=a-n$.

- Kirjeldab ära tegevused, mis on vaja teostada ülesande täitmiseks

- Enne algoritmi kirjeldamist tuleb määrata ära meetod, mida probleemi lahendamisel kasutatakse
- Ülesanne võib nõuda oma meetodi väljatöötamist!

- Koostada algoritm ristküliku külgede pikkuste leidmiseks, kui on teada selle ristküliku pindala ning üks külg on teisest n ühiku võrra lühem.

Algoritmimine

Tuleb lahendada ruutvõrrand $a^2 - na - S = 0$:

1. Sisestada n ja S
2. Leida $D = (-n)^2 + 4S$
3. Leida $a = \frac{n \pm \sqrt{D}}{2}$
4. Leida $b = a - n$
5. Väljastada a ja b

- **Antud ülesande piires universaalne**
Algoritmiga peab saama lahendada kõiki sarnaseid ülesandeid mistahes sisendandmete korral
- **Resultatiivne**
Programmi töö peab alati lõppema mingi tulemusega (otsitav väärtus, veateade).
Programm ei tohi “rippuma jääda”

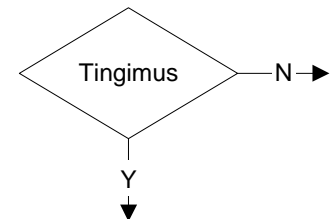
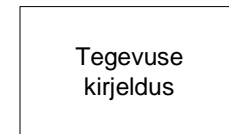
Algoritm peab võimaldama

- Teda realiseerida arvutis kirjeldatavate käskudega
- Teostada andmete sisestamist ja väljastamist, et programm ei töötaks “musta kastina”
- Tükeldada keerulisemat ülesannet väiksemateks, mida on võimalik eraldi programmeerida

- Algoritm ei ole mingis programmeerimiskeeles kirjutatud käskude jada
- Algoritm ei tohi sõltuda kasutatavast programmeerimiskeelest
- Algoritm ei tohi sisaldada ühegi programmeerimiskeele elemente

- Kujutab lahendus algoritmi graafiliselt
- Võimaldab paremini aru saada operatsioonide iseloomust, mida on vaja algoritmi täitmiseks teostada

- Programmi algus/lõpp
- Tegevus
- Tingimus
- Ühendus (nool näitab tegevuste järjekorda)



Algoritmide näiteid

Ivari Horm

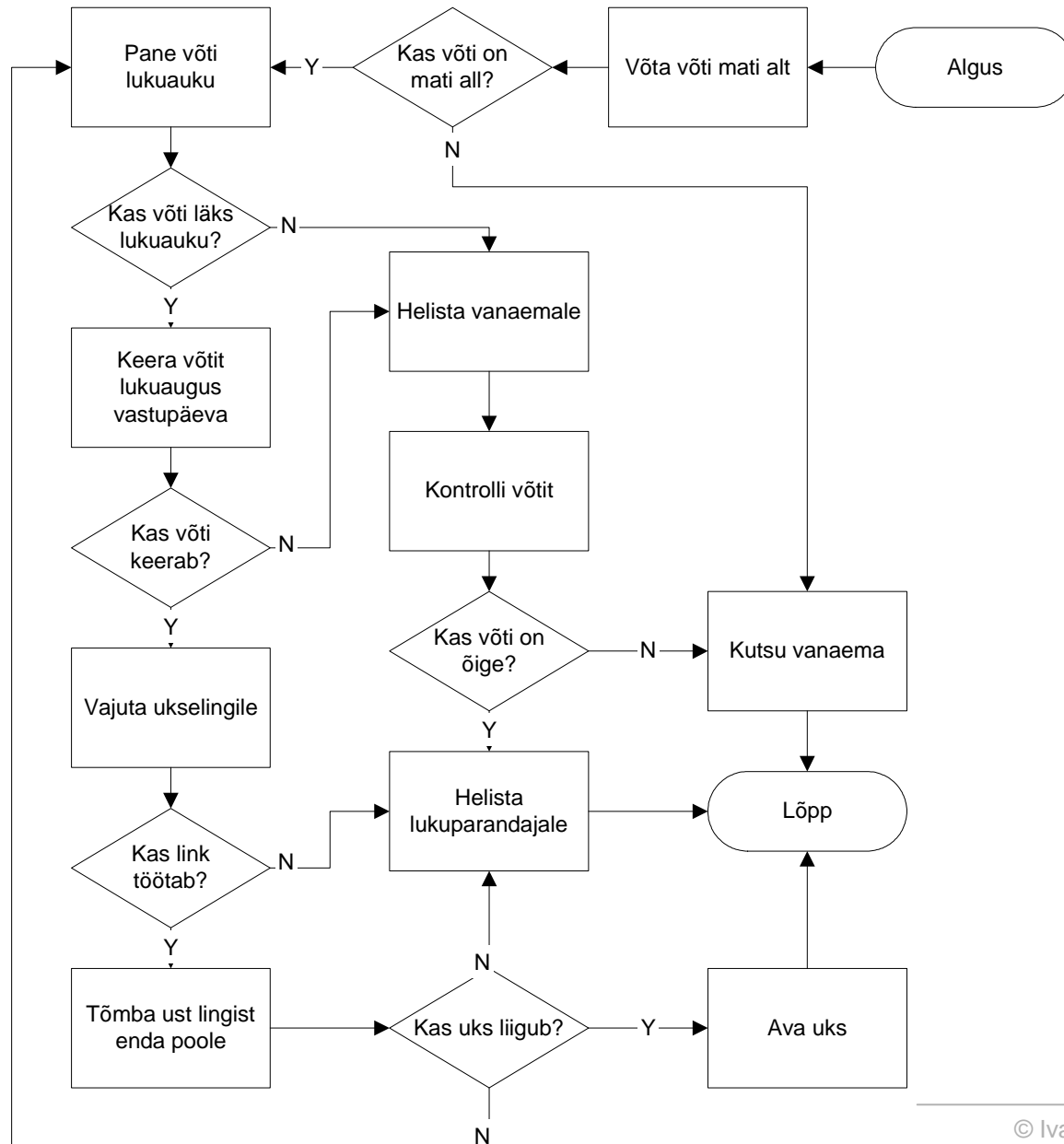
ranger@risk.ee

Ukse avamise algoritmi leidmine



- **Lähtetingimused**
 - Uks avaneb väljapoole
 - Uks on suletud ja lukustatud
 - Uksel on vaid üks lukk
 - Võti on mati all
 - Teine võti on vanaema käes
- **Tulemus**
 - Ukse avanemine

Ukse avamise algoritmi plokskeem



Malelauamustri koostamise algoritm



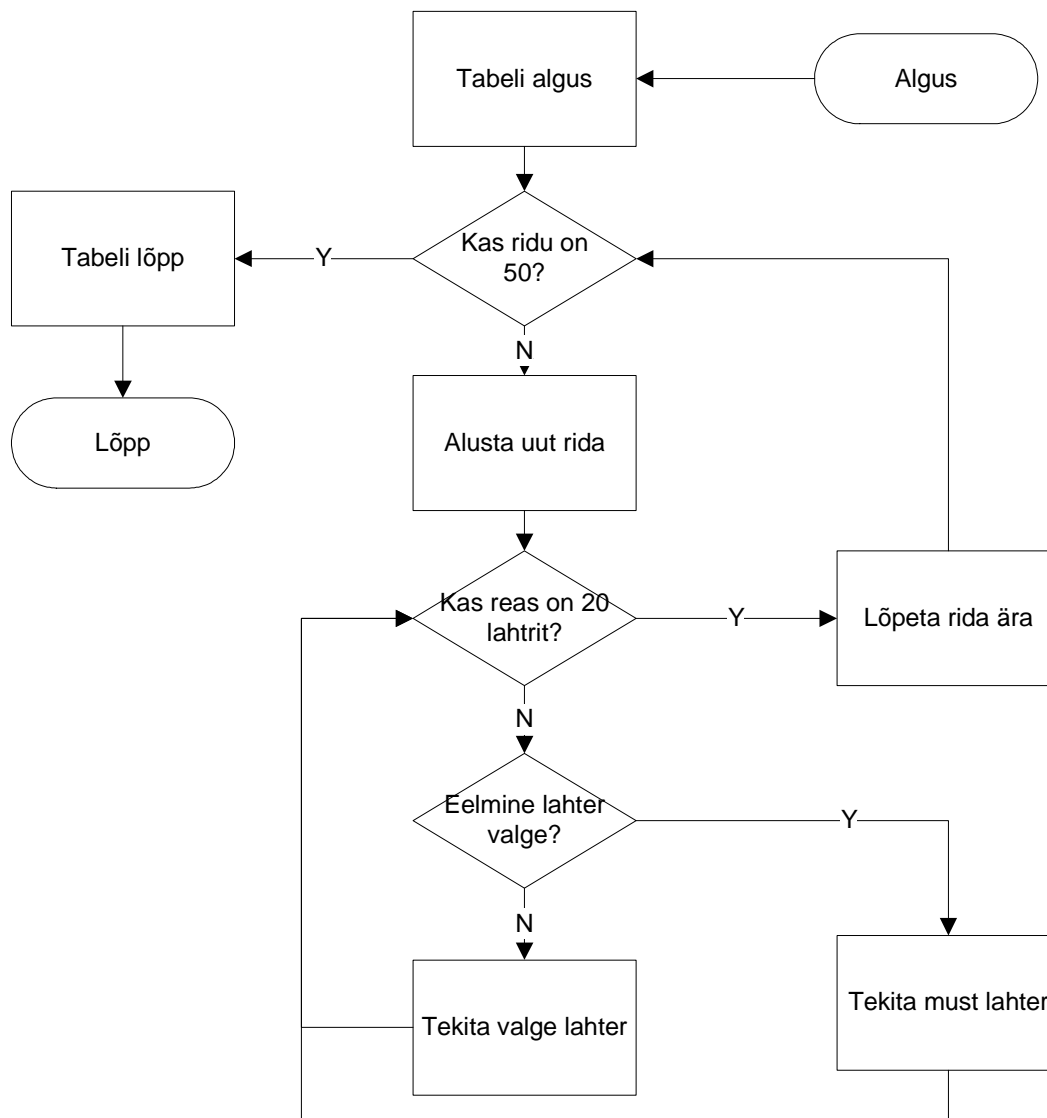
- **Lähtetingimused**

- Mustri suurus on 50x20
- Ruudud värvitakse sarnaselt malelauale

- **Tulemus**

- 50x20 ruudust koosnev malelauamuster

Malelauamustri algoritmi plokskeem



- Antud algoritm ei ole universaalne
- Võimaldab tekitada vaid 50x20 mustrit
- Algoritmi peaks olema lisatud suhtlemine välismaailmaga, et küsida ridade ja veergude arvu
- Sellisel juhul tuleb lisada ka kontroll, et sisendandmed õiged on (ridade ja veergude arv > 0)!

Hapupiima joomise algoritm

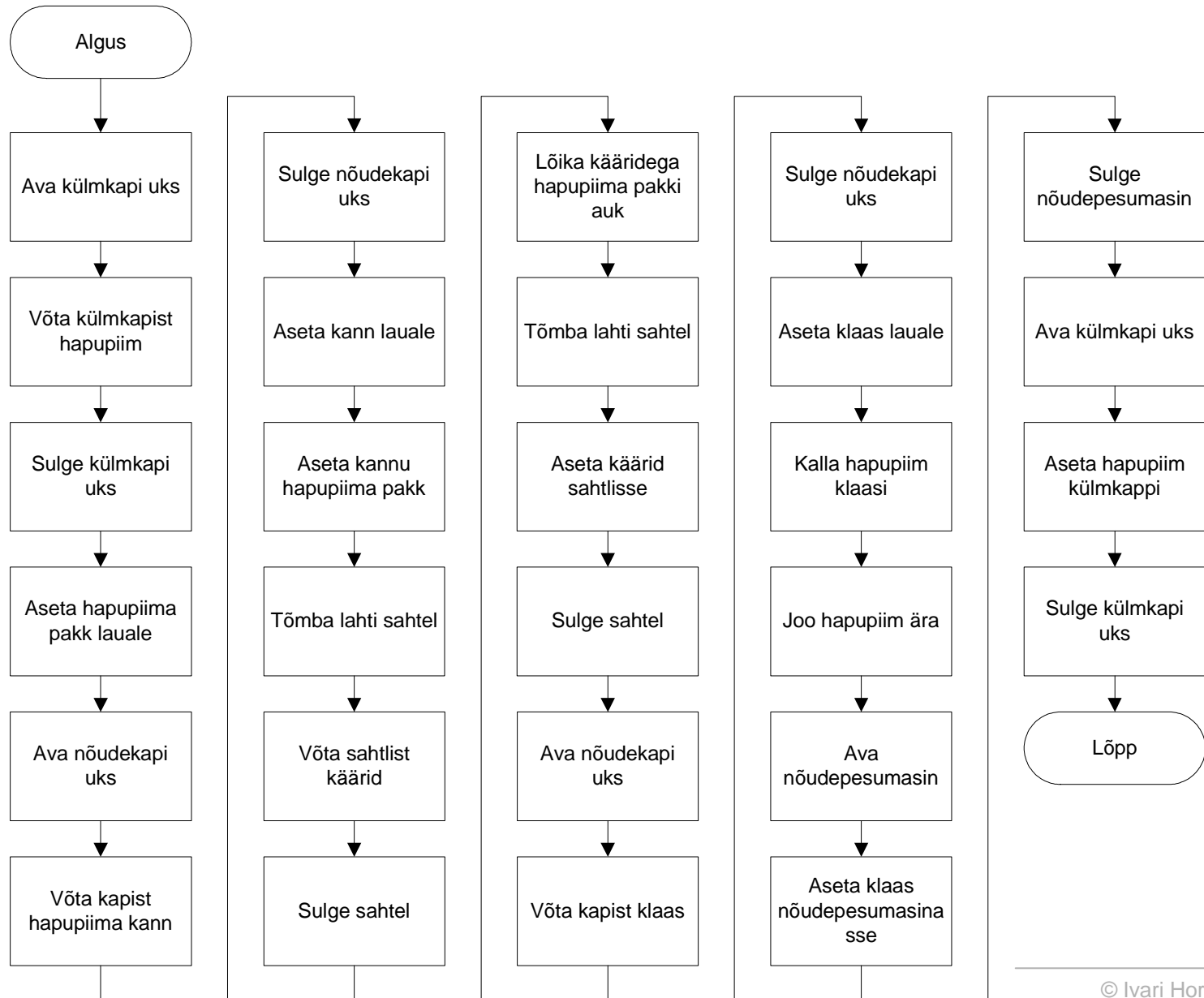


- Koostada hapupiima joomise algoritm
- Panna kirja tegevuse tulemus
- Alamtegevuste täitmise õnnestumist kontrollida ei ole vaja

▪ Lähtetingimused

- Hapupiim on kinnises kilepakis
- Kilepakk on külmkapis
- Kann hapupiima jaoks on nõudekapis
- Klaas on nõudekapis
- Käärid on sahtlis
- Mustad nõud tuleb panna nõudepesumasinasse
- Kõik võetud asjad tuleb oma kohale tagasi asetada

Hapupiima joomise algoritmi plokskeem



Programmeerimine

Ivari Horm

ranger@risk.ee

- Käskude jada, mida arvuti peab ülesande lahendamiseks täitma

- Masinkood
- Sümbolkeel
 - Madalataseme keeled
 - Kõrgkeeled

- Programm sisaldab vahetult protsessori käske
- Käsud on numbrilisel kujul
- Töötatakse vahetult arvuti mäluaadressidega
- Kõigis teistes keeltes kirjutatud programmid teisendatakse täitmiseks alati masinkoodi

Masinkood

8020	78
8021	A9 80
8023	8D 15 03
8026	A9 2D
8028	8D 14 03
802B	58
802C	60
802D	EE 20 D0
8030	4C 31 EA

- Käskude andmiseks kasutatakse numbrite asemel käsku kirjeldavaid lühendeid
- Madala taseme keeled on seotud riistvaraga
- Kõrgtaseme keeled on riistvarast sõltumatud

- *Assembler, assembly language*
- Madalatasemeline keel riistvara- ja suure jõudlusega programmide kirjutamiseks
- Valmis programm tuleb masinkoodi tõlkida e. **assembleerida**

Assembler

```
Start:      .org $8020
SEI
LDA         #$80
STA         $0315
LDA         #$2D
STA         $0314
CLI
RTS
INC         $D020
JMP         $EA31
```

- Koosnevad lausetest
- Kirjeldavad lahendusalgorithmi inimese mõtlemisele sarnaselt
- Saab kasutada muutujaid, tsükleid, kordusi, alamprogramme, funktsioone jpm.

- Kõrgkeeltes kirjutatud programmid tuleb masinkoodi transleerida.
- Selleks kasutatakse **translaatoreid**, mis võivad olla nii riistkui tarkvaralised.

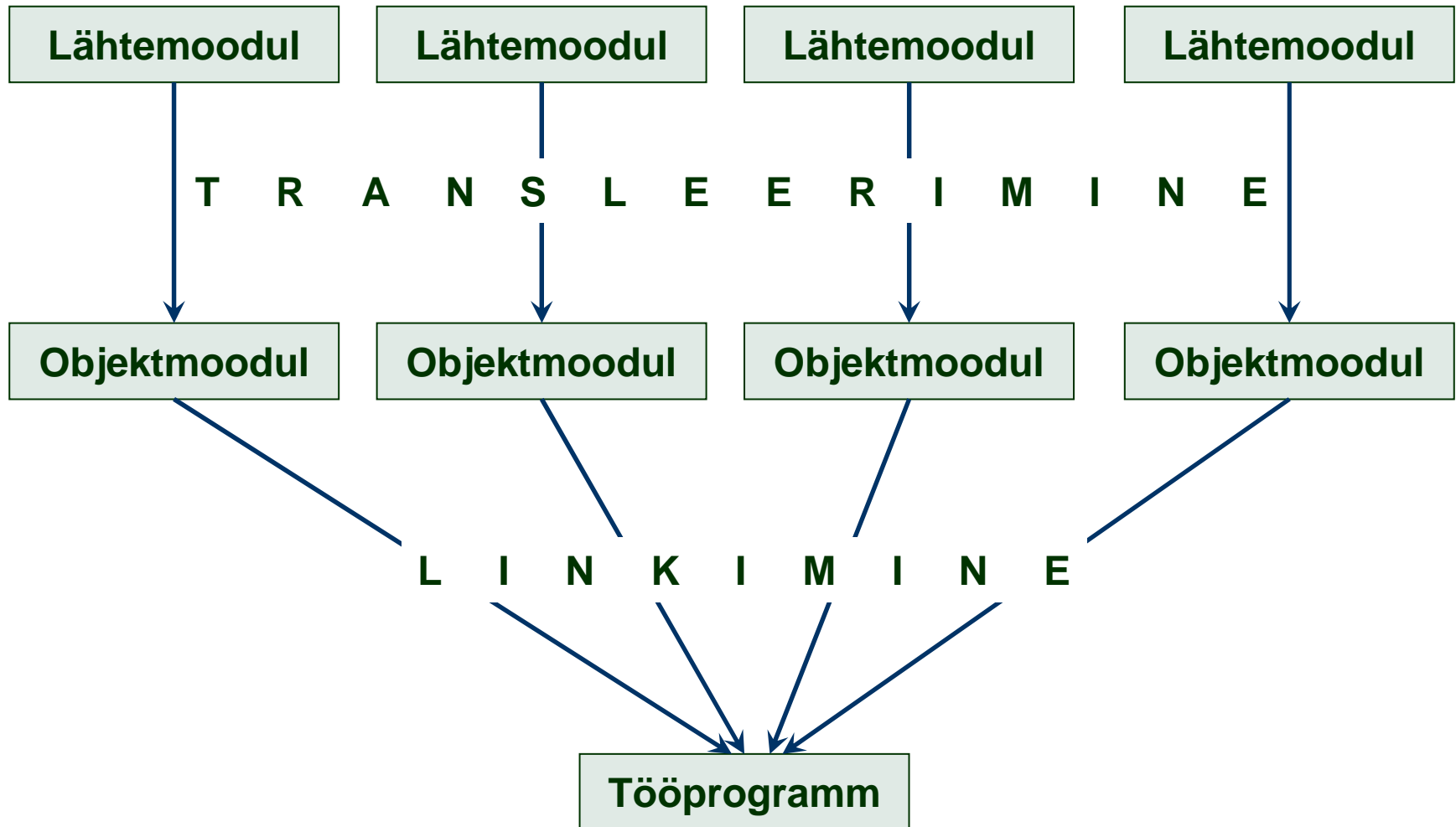


- **Kompilaatorid**
Transleerimine peab toimuma enne programmi täitmist
- **Interpretaatorid**
Transleerimine toimub programmi täitmise ajal

- Lähteprogramm transleeritakse vahekujule
- Sellisele kujule viidud käske nimetatakse **objektkoodiks**
- Sarnaneb masinkoodile

- Objektkoodi sisaldavaid programmiosi nimetatakse **objektmooduliteks**
- Enne programmi täitmist tuleb objektmoodulid kokku **linkida**
- Objektkood sisaldab linkimiseks vajalikku infot

Mitmetasemeline kompileerimine



Mitmetasemeline kompileerimine

- Võimaldab programmi osi eraldi transleerida
- Kasulik suurte programmide puhul
- Muudatuse tegemisel ei ole vaja kogu koodi uuesti kompileerida