

# PROGRAMME KÕIGILE

TARTU  
1983

TARTU RIIKLIK ÜLIKOOL

Arvutuskeskus

# F O R T R A N

(informatsioonivahetus ja silumine)

Programme kõigile

Koostanud Ü. Kaasik

TARTU 1983

Kinnitatud matemaatikateaduskonna nõukogus  
24. märtsil 1983. a.

Käesolev väljaanne kujutab endast TRÜ Arvutuskeskuses kõige laiemat tarbijaskonda leidnud algoritmikeele FORTRAN kasutajatele määratud metoodilise juhendi teist osa, olles seega sarjas "Programme kõigile" 1982.a. ilmunud vihiku "FORTRAN (keele põhikonstruktsioonid)" vahetu järg. Seost arvestades pole seal kirjeldatud mõistete semantikat siin enam uuesti selgitatud - piirduetakse vaid vastavate viidetega täheühendi FKP abil. Nii näiteks kirjutus "vt. FKP 2" viitab nimetatud vihiku leheküljele 2, kus kirjeldatud tähistusviis leiab kasutamist ka käesolevas vihikus.

FORTRAN.  
(Обмен информации и отладка).  
Программы для всех.  
Составитель Джо. К а а з и к.  
На эстонском языке.  
Тартуский государственный университет.  
ЭССР, 202400, г.Тарту, ул.Школьная, 18.  
Vastutav toimetaja J. Tapfer.  
Paljundamisele antud 11.05.1983.  
Formaat 60x84/16.  
Rotaatoripaber.  
Masinakiri. Rotaprint.  
Tingtrükipoognaid 5,58.  
Arvestuspoognaid 4,06. Trükipoognaid 6,0.  
Trükiarv 300.  
Tell. nr. 527.  
Hind 15 kop.  
TRÜ trükikoda. ENSV, 202400 Tartu, Palsoni t. 14.

## 1. SISEND-VÄLJUNDOOPERATSIOONID

### 1.1. Informatsioonivahetuse üldpõhimõtted

Informatsioonivahetus sisemälu ja välisseadmete vahel on FORTRANis kohaldatud arvuti EC-1022 üldise operatsioonisüsteemiga, kusjuures kasutatavateks välisseadmeteks on (perfo)kaartsisend ja -väljund, (magnet)lintmälu, (magnet)ketasmälu, trükiseade ning (operaatoripuldi) kirjutusmasin. Neist ainult lint- ja ketasmälu võimaldavad informatsioonivahetust mõlemas suunas, ülejäänud välisseadmete korral on vahetuse suund üheselt määratud (iga katse vastassuunaliseks informatsioonivahetuseks neil seadmel toob kaasa programmi töö katkestamise).

Vahetatav informatsioon tuleb väliskanduritele vormistada operatsioonisüsteemi OS nõudeid rahuldavate andmekogumitena, mida järgnevas nimetame failideks. Failide tegeliku vormistamise üksikasjadega tutvumiseks võib kasutada sarja "Programme kõigile" väljaannet "Operatsioonisüsteemist OS arvutile EC-1022". Enamasti pole aga FORTRAN-programmi koostajal neid üksikasju tarvis tunda - piisab üldpildist, mille kohaselt fail kujutab terviklike alajaotuste, nn. kirjetejada, kusjuures iga kirje on tõlgendatav sümbolite jadana.

Vahetu füüsilise tähenduse omandab kirje üksnes kaartsendi või -väljundi ja trükiseadme korral, kus kirjeks osutub vastavalt perfokaart ja trükirida. Sellega on aga nende seadmete jaoks ühtlasi määratud ka kirje maksimaalpikkused: perfokaardil 80 ja trükireas 121 sümbolit.

Informatsiooni saab faili ning sisemälu vahel vahetada üksnes kirjete kaupa. Seega faili ja informatsioonivahetuse seisukohalt on kirje lahutamatu tervik - kirje üksiksümbolitega opereerimine osutub võimalikuks ainult sisemälus.

Sõltuvalt sellest, kuidas on korraldatud juurdepääs kirjetele, jagunevad failid kahte klassi - jadafailid ja otsefailid. Jadafail on moodustatav igal välisseadmel, kirjed salvestatakse niisugusesse faili nende laekumise järjekorras või siis loetakse sealt nende salvestusjärjekorras. Otsefail on moodustatav ainult ketasmälus, kirjeid võib sellisesse faili salvestada ja sealt lugeda suvalises järjekorras, tuleb ainult näidata vastava kirje järjekorranumber failis.

Mõnevõrra erandlikul viisil on organiseeritud väljastamine trükiseadmele. Nimelt ei kuulu trükireana (kui kirjena) väljastamiseks määratud sümbolite jada esimene sümbol tegelikult üldse väljastamisele, vaid seda sümbolit tõlgendatakse hoopis trükirea juhtsümbolina. Kui juhtsümboliks on +, siis toimub vastava rea trükk ilma paberi edasilükketa; kui juhtsümboliks on 1, siis minnakse üle uuele leheküljele; kui juhtsümboliks on tühi või null, siis lükatakse paberit enne trükki vastavalt ühe või kahe rea võrra edasi. Kulgi kirje võib trükiseadme korral sisaldada kuni 121 sümbolit, kuulub neist väljastamisele seega siiski ainult 120 viimast.

Operatsioonisüsteemi OS kasutamisel toimub programmis vajalike andmekogumite seostamine välisseadmetega mitte vahetult programmis, vaid lahendamistellimuses, tööde juhtimise keele juhtlausete abil. Vajalikud juhtseadmed näidatakse juhtlausetes sealjuures nende tüübinumbrite või rühmanimede kaudu - vastavuse korraldamine konkreetsete füüsiliste seadmetega jääb täielikult operatsioonisüsteemi hooleks.

Andmekogumite eristamiseks FORTRAN-programmis kinnistatakse igale failile kindel failinumber ning kõik pöördumised faili poole toimuvad failinumbri kaudu. TRÜ arvutuskeskuses kasutatava operatsioonisüsteemi korral saab ühes programmis tegemist olla kuni kahekümne failiga ja nende failinumbriid võib valida naturaalarvude 1,...,20 hulgast.

Lahendamistellimuses tuleb iga programmis kasutatud failinumbri jaoks vastava juhtlausetega määrata sobivat tüüpi välisseade (lähemalt on juhtlausete koostamisega seotud küsimusi vaadeldud käesoleva väljaande viimases peatükis). Sealjuures on põhimõtteliselt võimalik luua ühel seadmel ka mitu sama failinumbriga jadafaili, kuid segadustest hoidumiseks on seda võimalust soovitatav vältida - järgnevas käsitluses igatahes vastavaid näiteid ei tooda.

TRÜ arvutuskeskuse standardsete juhtprotseduuride kasutamise korral (vt. p. 2.3) pole kahe failinumbri jaoks vastavaid juhtlauset tarvis lisada: protseduurid ise määravad failinumbrile 5 vastavusse kaartsisendi ja failinumbrile 6 trükiseadme. Neid standardseid määranguid saab uute juhtlausete lisamisega muidugi muuta, kuid kogu järgnevas käsitluses eeldatakse, et seda pole tehtud.

## 1.2. Informatsioonivahetus jadafailidega

Informatsioonivahetust teostavaks direktiiviks on vastavalt vahetuse suunale kas sisestamis- või väljastamisdirektiiv. Informatsiooni (vähemalt ühe kirje) lugemiseks jadafailist sisemällu tuleb kasutada sisestamisdirektiivi, mille üldkuju on

READ (n,f,e) l

kus n tähendab failinumbrit (esitatakse kas märgita konstandi või täisarvulise lihtmuutuja kujul), f kas vastava formaadikirjelduse märgendit (vt. p. 4.3), seda kirjeldust sisaldava massiivi nime, sümbolit \* või siis nimistu nime (vt. p. 4.5), e ekstraväljundi kirjeldust ja l informatsioonivahetusest osa võtvate objektide loetelu.

Objektide loetelu l annab nende muutujate nimestiku ja järjekorra, kuhu sisestatav informatsioon tuleb salvestada. Loetelu koosneb elementidest, mis on eraldatud üksteisest komadega. Elementiks võib olla kas muutuja (liht- või indeksitega), massiivi identifikaator või nn. tsüklikiline element. Viimane paikneb sulgudes ja peab sisaldama vähemalt ühe tsüklikloendajast sõltuva indeksitega muutuja ning selle tsüklikloendaja muutumiseeskirja (mis kirjutatakse täpselt samal kujul nagu tsükliidirektiivis - vt. FKP 58). Näiteks objektide loetelu tsüklikiline element

(A(K),K = 1,13,2)

on samaväärne loeteluga

$A(1), A(3), A(5), A(7), A(9), A(11), A(13)$

Tsükliline element võib aga omakorda sisaldada teisi elemente ja terveid loetelusid, mis tsükliloendaja muutumisel kuuluvad kordamisele. Näiteks tsükliline element

$(A(I), B, C, I = 1, 3)$

on samaväärne loeteluga

$A(1), B, C, A(2), B, C, A(3), B, C$

element

$((F(I, K), K = 1, 3), I = 1, 4)$

aga loeteluga

$F(1, 1), F(1, 2), F(1, 3), F(2, 1), F(2, 2), F(2, 3),$   
 $F(3, 1), F(3, 2), F(3, 3), F(4, 1), F(4, 2), F(4, 3)$

Viimast moodust kasutatakse siis, kui massiivi on tarvis sisestada ridade kaupa. Veergude kaupa sisestamiseks jätkub

massiivi identifikaatori lülitamisest loetelusse. Nii näiteks kirjeldusega

DIMENSION F(4,3)

määratud massiivi identifikaatori F esinemine objektide loetelu elemendina on samaväärne loeteluga

F(1,1),F(2,1),F(3,1),F(4,1),F(1,2),F(2,2),  
F(3,2),F(4,2),F(1,3),F(2,3),F(3,3),F(4,3)

Indeksitega muutuja esinemisel objektide loetelus võib indeksi kohal paikneda üldiselt suvaline aritmeetiline avaldis, keelatud on vaid niisuguste funktsioonide kasutamine, mis ise välisseadmete poole pöörduvad.

Ekstraväljundi kirjeldus e annab väljumismärgendid juhtudeks, kui sisestamisel esines tõrge või väliskanduril lõppes informatsioon (s.t. sisestamisdirektiivi täitmist nõutakse veel pärast faili viimase kirje sisestamist). Esimese juhu jaoks kirjeldab e veaväljundi kujul

ERR = m

kus m on sama mooduli selle direktiivi märgend, kuhu tõrke esinemisel tuleb anda juhtimine (vea põhjustanud kirje jääb sealjuures sisestamata). Teise juhu tarvis võib e kirjeldada veel lõpuväljundi kujul

END = m

kus m on selle direktiivi märgend samast moodulist, kuhu antakse juhtimine juba lõppenud faili poole pöördumise korral.

Kirjeldus e võib sisaldada kas nii vea- kui ka lõpuväljundi suvalises järjekorras, ainult ühe neist või mitte ühtegi väljundit (vastava ekstraväljundi puudumisel kutsutakse tõrge või faili lõppemine esile programmi tööd katkestamise). Seega on näiteks järgmised viis sisestamisdirektiivi kõik lubatava kujuga:

```
READ (5,1Ø,ERR = 2Ø,END = 15) S
READ (5,1Ø,END = 15,ERR = 2Ø) S
READ (5,1Ø,END = 15) A
READ (5,1Ø,ERR = 2Ø) B
READ (5,1Ø) C
```

Sisestamisdirektiivi üldkujus võib puududa kas formaadikirjelduse märgend f, ekstraväljundi kirjeldus e, objektide loetelu l või ka kõik kolm. Seega sisestamisdirektiivi võib kasutada veel järgmisel kuuel lühendatud kujul:

READ (n,e) l
READ (n,f,e)
READ (n,e)
READ (n) l
READ (n,f)
READ (n)

Esimene ja neljas nendest tähendavad, et loeteluga 1 määratud baitidesse sisestatakse üks kirje, kusjuures informatsiooni ei teisendata ega redigeerita. See nn. formaadita sisestamine on kasutatav vaid siis, kui vastav kirje on väliskanduril sobivalt vormistatud, näiteks viidud sinna formaadita väljastamise teel (vt. lk. 11).

Teine ja viies lihtsustatud sisestamisdirektiiv leiavad kasutamist siis, kui soovitakse sisestada redigeerimistekste: objektiks on nüüd formaadikirjeldus (vastav näide vt. lk. 28). Viies variant leiab kasutamist ka sisestamisel niimista järgi (vt. p. 1.5).

Kolmas ja kuues lihtsustatud variant on nn. tühisisestused: loetakse üks kirje, kuid saadavat informatsiooni sisemällu ei kirjutata. Niisugune sisestamisviis on vajalik jadafaali ühe kirje vahelejätmiseks.

Tõrke või faili lõpu korral väljutakse esimesel kolmel juhul direktiivis antud ekstraväljundi järgi, viimasel kolmel juhul aga lõpetatakse siis programmi täitmine ja väljastatakse trükiseadmel vastav teade.

Sisestamiseks perfokaartidelt võib kasutada veel sisestamisdirektiivi kaht lihtsustatud kuju

READ f,l
READ f

mis TRÜ arvutuskeskuse juhtprotseduuride standardsete määrangute korral osutuvad samaväärseteks vastavalt direktiividega READ (5,f) l ja READ (5,f).

Informatsiooni viimiseks sisemälust välisseadmele tuleb kasutada väljastamisdirektiivi, mille üldkuju on

WRITE (n,f,e) l
-----------------

kus metasümbolite n, f ja l tähendus langeb kokku samade sümbolite tähendusega sisestamisdirektiivi korral, e aga tähendab ainult veaväljundit (et väljastamisvigu avastatakse harva, siis jäetakse see väljund enamasti kirjutamata). Objektide loetelu määrab nüüd nende muutujate nimestiku ja järjekorra, mille väärtused kuuluvad väljastamisele.

Väljastamisdirektiivi võib kasutada ka järgmisel kahel lühendatud kujul:

WRITE (n) l
-------------

WRITE (n,f)
-------------

Esimene neist on määratud ühe kirje formaadita väljastamiseks, teine redigeerimistekstide väljastamiseks. Formaadita väljastamist kasutatakse peamiselt selleks, et ajutiselt väliskandurile (ketas- või lintmällu) hoiule viidud informatsiooni hiljem formaadita sisestamise teel võimalikult kiiresti sisemällu tagasi tuua (vt. ka p. 1.7).

Väljastamiseks trükiseadmele võib TRÜ arvutuskeskuses kasutada väljastamisdirektiivi kaht lihtsustatud kuju

PRINT f,l
-----------

PRINT f
---------

mis on samaväärsed vastavalt direktiiviga WRITE (6,f) I ja WRITE (6,f). Sellise lihtsustatud kujuga direktiivi nimetame edaspidi trükkimisdirektiiviks.

Informatsiooni vahetamisel jadafailidega lint- või keetasmälus tuleb peale sisestamis- ja väljastamisdirektiivi sageli kasutada veel järgmist kolme abidirektiivi.

Abidirektiividest kõige sagedamini vajalikuks osutuv failitagurdamise direktiiv kirjutatakse kujul

REWIND n

kus n on failinumber. Selle direktiivi toimetatakse fail algseisu, nii et järjekordne sisestamis- või väljastamisdirektiiv pöördub faili esimese kirje poole. Kui viimane vaadeldava faili poole pöördumise direktiiv oli väljastamisdirektiiv, siis failitagurdamise direktiivi mõjul toimub ühtlasi faili lõpetamine. Kui programm pole aga vaadeldava faili poole veel üldse pöördunud, siis failitagurdamise direktiivi ignoreeritakse, s.t. ta on sel juhul samaväärne tühi-direktiiviga.

Kirjetagurdamise direktiiv kirjutatakse kujul

BACKSPACE n

kus n on failinumber. Direktiivi toimetatakse fail ühe kirje võrra tagasi (teda ei tohi kasutada nimistu järgi väljastatud faili korral). Kui faili n poole pole veel pöördunud või fail on algseisus, siis vaadeldavat direktiivi igno-

reeritakse. Kui viimaseks pöördumiseks vaadeldava faili poole oli väljastamine, siis kaasneb ka kirjetagurdamise direktiiviga ühtlasi faili lõpetamine.

Faili- ja kirjetagurdamise direktiive saab täita üksnes lint- või ketasmälus paiknevate jadafailide korral. Nii näiteks pole kirjetagurdamise direktiiviga võimalik trükiseadmes paberit ühe rea võrra tagasi tõmmata või kaartsisendis viimati sisestatud perfokaarti uuesti sisestamisvalmis seada. Kui programmi täitmise ajal selgub, et üks nendest direktiividest pöördub selleks mitte sobiva välisseadme poole, siis programmi täitmine katkestatakse ning väljastatakse vastav veateade (sellist viga pole transleerimisel võimalik avastada, sest failide seostamine konkreetsete välisseadmetega toimub ju alles lahendamistellimuses). Katkestamist ei toimu vaid siis, kui direktiiv kuuluks mõnel ülalnimetatud põhjusel niikuinii ignoreerimisele.

Kolmandaks jadafailide korral kasutatavaks abidirektiiviks on faililõpetamise direktiiv, mis kirjutatakse kujul

ENDFILE n ,

kus n on failinumber. Direktiiv väljastab väliskandurile erilise faili lõputunnuse, mis kujutab endast operatsioonisüsteemi poolt määratud struktuuriga kirjet. Kui vaadeldavale direktiivile järgneb kirjetagurdamise direktiiv, siis see muudab kättesaadavaks lõputunnuse, mitte aga viimase kirje: faili viimasena väljastatud kirje kättesaadavaks tegemine nõuab nüüd kaht kirjetagurdamise direktiivi.

### 1.3. Formaadikirjeldus

Informatsioonivahetusel toimub peale andmete füüsilise ülekandmise enamasti ka nende teisendamine (ühest esitusviisist teise) ja redigeerimine (pealkirjadega varustamine, tabeliks vormistamine jms.). Nende operatsioonide sisu määratakse kindlaks vastava formaadikirjeldusega, mis ühtlasi täpsustab, kuidas on informatsioon väliskanduril kujutatud või kuidas ta sinna tuleb viia. Formaadikirjeldus määrab kogu vahetatava informatsiooni jaotuse kirjeteks ja samuti ka iga kirje kui sümboleite jada jaotamise üksikkomponentideks, nn. väljadeks. Erinevalt kõigist teistest kirjeldustest võib formaadikirjeldus paikneda mooduli mistahes kohas. Formaadikirjelduste eristamiseks tuleb nad varustada märgenditega.

Formaadikirjeldus algab reserveeritud sõnaga FORMAT, millele sulgudes järgneb nn. väljakirjeldajate loetelu (iga kahe väljakirjeldaja vahel on eraldajaks kas koma või kaldkriips, viimane tähistab sealjuures kirje lõppu). Väljakirjeldaja määrab väliskanduril teatud arvu positsioone (perfokaardil veerge, magnetlindil või -kettal sümboleid, trüki-seadmel täheruume) ja nendelt positsioonidelt loetava või sinna kirjutatava informatsiooni teisendamise viisi (erandiks on väljakirjeldaja nimega T, mis määrab üksnes positsiooni kirjes, millest alates toimib järgmine väljakirjeldaja). Tuleb jälgida, et ühte kirjesse kuuluvad väljakirjeldajad ei määraks kirje maksimaalpikkusest rohkem positsioone; maksimaalsest väiksema positsioonide arvu korral jääb kirje lõpp lihtsalt lugemata või sinna kirjutatakse tühihikud.

Peale positsioonide arvu  $n$  (välja laiuse) sisaldab väljakirjeldaja veel teisenduse nime: tähe I, F, E, D, Q, G, L, A, Z, H, X või T. Olenevalt informatsioonivahetuse suunast tähendab teisendus kas näiteks arvu teisendamist vastavaks sümbolite jadaks või sümbolite jada tõlgendamist arvuna ja selle teisendamist kahendsüsteemi. Vaatlemegi iga teisendust mõlemas suunas veidi lähemalt.

Väljakirjeldaja kujul  $In$  tähendab sisestamisel, et väliskanduri järjekordsel  $n$  positsioonil paiknevaid sümboleid tõlgendatakse ühe täisarvuna, mis teisendatakse kahendsüsteemi ja salvestatakse objektide loetelus näidatud muutuja jaoks reserveeritud baitidesse. Nende  $n$  sümboli hulgas võib enne esimest numbrit olla märk, kõik tühikud sümbolite hulgas loetakse nullideks. Järgmises tabelis on toodud mõned näited vaadeldava väljakirjeldaja toimest sisestamisel (käesoleva punkti tabelites on number null parema loetavuse huvides jäetud läbi kriipsutamata - üksnes väljakirjeldajates, kui vahetult FORTRAN-programmi kuuluvates konstruktsioonides kasutatakse kirjutusviisi  $\emptyset$ ):

Sümbolid väliskanduril	Väljakirjeldaja	Saadav arv
12345	I5	12345
-142	I6	-1420
2222	I8	202020
+99	I4	99

Väljastamisel võetakse väljakirjeldaja  $In$  toimel järjekordse (objektide loeteluga määratud) muutuja väärtus, tei-

sendatakse see sümbolitè jadaks ning kantakse väliskanduri järjekordsele n positsioonile. Vabad positsioonid arvu ees täidetakse tühikutega, esimese numbriga ette pannakse negatiivse arvu korral miinus, positiivse arvu korral plussi väliskandurile ei kanta. Kui tühikutest erinevad sümbolid ei mahu väljale ära, siis kantakse väliskandurile n täärni koodi. Järgmises tabelis on toodud mõned näited vaadeldava väljakirjeldaja toimest väljastamisel:

Väljastatav arv	Väljakirjeldaja	Esitus väliskanduril
12345	I5	12345
-199	I6	LL-199
15	I1	*
-4865	I4	****

Reaalrude teisendamiseks määratud väljakirjeldajates kirjutatakse teisenduse nime (F, E, D või Q) taha kaks täisarvu, väljakirjeldaja kuju on seega kas Fn.m, En.m, Dn.m või Qn.m, kus naturaalarv n tähendab välja laiust, m aga murdosa numbrite arvu.

Kõik need väljakirjeldajad tähendavad sisestamisel, et väliskanduri järgmise n sümboli jada tõlgendatakse reaalarvuna, mis teisendatakse ujukoma arvuks ja salvestatakse objektide loetelus määratud baitidesse. Arv võib väliskanduril olla kujutatud kas tavalise kümnendarvuna või poollogaritmilisel kujul. Erinevalt reaalarvuliste konstantide kirjutamise reegleist (vt. FKP 8) võib siin järgu märgi olemasolu korral täht E või D poollogaritmilises kujus ka puududa.

Kui reaalarvu kujutis väliskanduril sisaldab kümnendpunkti, siis väljakirjeldajas antud täisarvu  $m$  ei arvestata. Kümnendpunkti puudumisel loetakse arvu või selle mantissi viimased  $m$  kohta murdosaks, s.t. kümnendpunkt lisatakse parameetri  $m$  järgi. Tühikud väliskanduril loetakse nullideks. Järgmises tabelis on toodud mõned näited reaalarvude teisendamisest sisestamisel:

Sümbolid väliskanduril	Väljakirjeldaja	Saadav arv
$\llcorner\llcorner 23.23$	F7.3	$0,2323 \cdot 10^2$
$-\llcorner 123\llcorner 567$	D9.2	$-0,1230567 \cdot 10^5$
$-\llcorner 123\llcorner 567$	Q9.2	$-0,1230567 \cdot 10^5$
$+88.2E-3$	F8.7	$0,882 \cdot 10^{-1}$
$\llcorner\llcorner\llcorner\llcorner$	E5.3	$0,0 \cdot 10^0$
$\llcorner-954 + \llcorner 5$	D8.4	$-0,954 \cdot 10^4$
$\llcorner+.5D1\llcorner$	E7.3	$0,5 \cdot 10^{10}$
$-33333Q-15\llcorner$	E11.1	$-0,33333 \cdot 10^{-11}$
$-33333E-15\llcorner$	Q11.5	$-0,33333 \cdot 10^{-15}$

Väljastamisel teisendatakse järjekordne reaalarv väljakirjeldaja  $F_n.m$  toimele  $n$ -sümboliliseks kümnendarvuks, mille murdosa sisaldab  $m$  numbrikohta. Ülearused positsioonid arvu ees täidetakse tühikutega, positiivse arvu ette märki ei panda (kui ära ei mahu ainult täisosa null, siis jäetakse see väljastamata). Kui arv väljale ära ei mahu, siis väljastatakse  $n$  täärni koodi. Kui arvu murdosa sisaldab üle  $m$  numbriga, siis toimub ümardamine. Järgmises tabelis on toodud mõned näited reaalarvude väljastamisest teisendusega F:

Väljastatav arv	Väljakirjeldaja	Esitus väliskanduril
$0,666666 \cdot 10^2$	F6.3	66.667
$-0,123456 \cdot 10^3$	F11.5	└-123.45600
$-0,1234 \cdot 10^0$	F6.4	-.1234
$-0,1234 \cdot 10^3$	F6.2	* * * * *

Väljakirjeldaja En.m, Dn.m või Qn.m toimel teisendatakse väljastatav reaalarv poollogaritmilisele kujule. Saadavas n sümboli jadas on kõigepealt arvu märk (väljastatakse ainult negatiivse arvu korral), seejärel mantissi täisososa null, kümnendpunkt ja mantissi murdosa m numbrit (vajaduse korral toimub ühtlasi ümardamine) ning lõpuks veel järk kujul: täht E, D või Q, järgu märk ja kaks numbrikohta (mittenegatiivse järgu korral väljastatakse märgi + asemel tühi). Kui n positsiooni sellega ei täitu, siis kirjutatakse arvu ette tühi-  
tühi, kui aga kõik vajalikud sümbolid n kohale ära ei mahu, siis väljastatakse n täärni koodi (kui ära ei mahu ainult täisososa null, siis jäetakse see väljastamata). Järgmises tabelis on toodud mõned näited reaalarvude väljastamisest teisendustega E, D ja Q:

Väljastatav arv	Väljakirjeldaja	Esitus väliskanduril
$0,23697 \cdot 10^9$	E9.4	.2370E└09
$-0,5523 \cdot 10^{-1}$	D1└.2	└-0.55D - 01
$0,12345 \cdot 10^{15}$	E12.4	└└0.1235E└15
$-0,77777 \cdot 10^2$	D1└.5	* * * * *
$-0,77777 \cdot 10^2$	Q12.5	└0.77777Q└02

Väljakirjeldaja Gn.m on kasutatav nii tõeväärtuste, täis- kui ka reaalarvude korral - konkreetne tõlgendus sõltub vastava muutuja tüübist objektide loetelust. Reaalarvude sisestamine selle väljakirjeldajaga toimub täpselt samuti nagu teisenduste F, E, D ja Q korral. Väljastatav reaalarv teisendatakse väljakirjeldaja Gn.m toimel sõltuvalt selle arvu suuruselt kas tavaliseks kümnendarvuks (nagu Fn.m toime) või poollogaritmilisele kujule (nagu En.m toime). Viimane moodus realiseerub nimelt siis, kui väljastatava arvu järk on negatiivne või suurem kui m. Väljastamisel tavalise kümnendarvuna saadakse kokku m numbrit, neist järguga võrdne arv täisosas. Arvu lõppu jääb sel juhul alati neli tühikut. Järgmises tabelis on toodud mõned näited reaalarvude väljastamisest teisendusega G:

Väljastatav arv	Väljakirjeldaja	Esitus väliskanduril
$0,55555 \cdot 10^{-2}$	G13.4	000055556E-02
$0,55555 \cdot 10^2$	G13.4	000055.560000
$-0,1234 \cdot 10^6$	G10.5	*****
$0,1234 \cdot 10^6$	G10.5	.12340E+06

Reaalarvude teisendamisel on veel võimalik muuta nende mastaabi, s.t. korrutada või jagada teguriga  $10^k$ , kus täisarv  $k$  on nn. mastaabitegur, mis peab rahuldama tingimust  $-127 \leq k \leq 127$ . Mastaabitegur  $k$  kirjutatakse vastava väljakirjeldaja ette koos tähega P (näiteks 5PF10.3, -1PG13.2). Positiivse mastaabiteguri ette märki ei kirjutata, mastaabitegureid 1 või -1 ei tohi kirjutamata jätta.

Mingi väljakirjeldaja ette kirjutatud mastaabitegur mõ-  
jub ka väljakirjelduste loetelu järgmistele elementidele (ai-  
nult neile, kus teisenduse nimeks on kas F, E, G, D või Q)  
kuni järgmise mastaabitegurini või siis loetelu lõpuni. See-  
ga järjestikuste väljakirjeldajate ühise mastaabiteguri võib  
kirjutada ainult esimese ette neist, sinna aga, kust alates  
mastaabitegurit enam tarvis ei ole, tuleb panna tegur null  
(näiteks kujul  $\emptyset PF1\emptyset.3$ ).

Tavaliste kümnendarvude sisestamisel jagatakse nad mas-  
taabiteguri toimel teguriga  $10^k$ , poollogaritmilisel kujul  
antud arvud aga sisestamisel ei muutu. Mõned näited mastaa-  
biteguri toimest sisestamisel on toodud järgmises tabelis:

Sümbolid väliskanduril	Väljakirjeldaja	Saadav arv
$\square\square\square-149$	2FE7.2	$-0,149 \cdot 10^{-1}$
$\square\square 4.7E-1$	5FE8.2	$0,47 \cdot 10^0$
$+\square 4.6.656$	-2FD9.2	$0,436656 \cdot 10^5$
$-777.44$	-3PG7.1	$-0,77744 \cdot 10^6$
$\square+0.1D-3$	3PQ8.5	$0,1 \cdot 10^{-3}$

Väljastamisel teisendusega F korrutatakse arv mastaabi-  
teguri toimel teguriga  $10^k$ . Tavalise kümnendarvu väljastami-  
sel teisendusega G seda korrutamist ei toimu. Väljastamisel  
poollogaritmilisele kujule (teisendused E, D, Q) arvu suurus  
mastaabitegurist ei muutu, küll aga väliskuju. Nimelt korru-  
tatakse mantiss teguriga  $10^k$ , järku aga vähendatakse k võr-  
ra. Mõned näited mastaabiteguri toimest väljastamisel on  
toodud järgmises tabelis:

Väljastatav arv	Väljakirjeldaja	Esitus väliskanduril
$0,18042 \cdot 10^2$	2PF1Ø.3	LL1804.200
$0,18042 \cdot 10^2$	2PG1Ø.3	LL18.0LLLL
$0,3456 \cdot 10^2$	-1PD12.4	LL0.0346DL03
$-0,3456 \cdot 10^2$	4PD12.4	L-3456.0D-02
$-0,12345 \cdot 10^3$	-3PE11.3	L-0.000EL06
$0,12345 \cdot 10^6$	2PG 9.3	L12.3EL04
$+0,12345 \cdot 10^5$	-2PQ1Ø.4	0.0012QL07

Loogiliste konstantide kujutamiseks väliskandureil kasutatakse tähti T ja F. Sisestamisel väljakirjeldajaga Ln loetakse väliskanduri järjekordsel n positsioonil paiknevad sümbolid: kui esimene tühikust erinev sümbol on T, siis omistatakse vastavale loogilisele muutujale väärtus "tõene"; kui aga esimene tühikust erinev sümbol on F või kogu väli on täidetud tühikutega, siis omistatakse muutujale väärtus "väär". Seega võib tõeväärtusi väliskanduril esitada näiteks sõnadega TOENE ja FALE.

Tõeväärtuste väljastamisel kantakse väljakirjeldaja Ln toimel väliskanduri esimesele n-1 positsioonile tühikud ja viimasele positsioonile väljastatavast väärtusest sõltuvalt kas täht T või F.

Teksti vahetamiseks väliskanduri ja sisemälu vahel kasutatakse väljakirjeldajat An. Sisestamisel loetakse selle väljakirjeldaja toimel väliskandurilt järjekordsed n sümboleid ja kasutatakse neid sõltuvalt objektide loetelu vastava muutuja pikkusest p. Kui  $n > p$ , siis kirjutatakse muutuja

väärtuseks viimased p sisestatud sümboleist (esimesed n-p sümboolit lähevad kaotsi; kui aga  $n < p$ , siis omistatavat tekstikonstanti täiendatakse paremalt p-n tühikuga.

Väljastamisel tõlgendatakse järjekordse muutuja väärtust tekstikonstandina ja kantakse see väliskandurile: kui  $n > p$ , siis tekstikonstanti täiendatakse vasakult n-p tühikuga; kui aga  $n < p$ , siis väljastatakse vaid esimesed n sümboolit. Mõned näited teisenduste L ja A kasutamisest sisestamisel ning väljastamisel on toodud järgmises kahes tabelis (kus teisenduse A korral eeldatakse, et vastav muutuja on 8-baidiline):

Sümbolid väliskanduril	Väljakirjeldaja	Saadav väärtus
LLL	I3	"väär"
LLTROE	I6	"tõene"
L SISESTAMINE	A12	ESTAMINE
SISEND	A6	SISENDLL

Muutuja väärtus	Väljakirjeldaja	Esitus väliskanduril
"tõene"	I5	LLLLL T
"väär"	I1	F
ABCDEFGHI	A3	ABC
ABCDEFGHI	A10	LL ABCDEFGH

Väljakirjeldaja Zn on määratud kuuteistkümnendarvude ülekanamiseks. Sisestamisel loetakse väliskandurilt järjekordsed n sümboolit, tõlgendatakse neid kuuteistkümnendnumbritena (tühik samastatakse nulliga) ja kirjutatakse vastava muutuja jaoks reserveeritud baitidesse - ühte baiti kaks

numbrit. Vajaduse korral täiendatakse arvu vasakult nullidega või jäetakse üleaarused numbrid vasakult salvestamata. Väljastamisel asendatakse puuduvad numbrid vasakult tühikutega või lõigatakse liigsed numbrid vasakult ära.

Järgmised kaks teisendust nimedega H ja X on määratud informatsiooni vahetamiseks formaadikirjelduse ja väliskanduri vahel: objektide loetelus nendele väljakirjeldajatele muutujaid ei vasta. Väljakirjeldaja nX tähendab sisestamisel, et väliskanduri järjekordsed n positsiooni jäetakse lihtsalt vahele. Väljastamisel kantakse selle väljakirjeldaja toimel väliskanduri järjekordsesse n positsiooni tühiku koodid.

Apostroofide vahel esitatud või kujul nH algav väljakirjeldaja on tegelikult tekstikonstant, mis väljastamisel kantakse vahetult väliskandurile. Näiteks väljakirjeldaja 8HLAENDUS või 'LAENDUS' toimel kantakse väliskanduri järjekordsesse kaheksasse positsiooni tekst LAENDUS. Sisestamisel loetakse sellise väljakirjeldaja toimel väliskandurilt n sümbolit ja kirjutatakse need formaadikirjeldusse senise n sümboli asemele. Näiteks väljakirjeldaja 13HAAAAAAAAAAAAA (või 'ABCDEFGHIJKLM') muutub pärast vastava sisestamisdirektiivi täitmist väliskanduril oleva teksti LAENDIT\_POLE toimel väljakirjeldajaks 13HLAENDIT\_POLE (või vastavalt väljakirjeldajaks 'LAENDIT\_POLE').

Väljakirjeldaja kujul Tn otseselt informatsioonivahetust ei juhi, ta määrab vaid kirjes positsiooni, millest alates toimivad järgmised väljakirjeldajad. Näiteks väljakirjeldaja T40 tähendab, et loetelu järgmise väljakirjeldajaga määratud väli algab kirje 40-ndast positsioonist. Väljastamisel tru-

kiseadmele tuleb siinjuures arvestada, et kirje algab juht-sümboliga, seega väljakirjeldajate paari T4Ø, 'A' toimel saadakse täht A trükirea 39-ndale, mitte aga 40-ndale kohale.

Informatsiooni vahetamisel juhtub sageli, et vahetamisele tulevad sama struktuuriga muutujaterühmad. Vastavate väljakirjeldajate rühmade korduva kirjutamise vältimiseks võib iga sellise rühma võtta sulgudesse ja kirjutada sulgude ette kordumiste arv. Kui rühm koosneb ühestainsast väljakirjeldajast, siis jätetakse sulud ära ja kordumiste arv kirjutatakse lihtsalt väljakirjeldaja ette (mastaabiteguri olemasolu korral selle ja ülejäänud väljakirjeldaja vahele). Näiteks formaadikirjelduste

17 FORMAT (I6,I6,I6,-2PE9.3,E9.3)

18 FORMAT (E7.2,F9.3,I4,F9.3,I4,F8.2)

(17 ja 18 on kirjelduste märgendid) asemel võib kasutada lühemat kirjutusviisi:

17 FORMAT (3I6,-2P2E9.3)

18 FORMAT (E7.2,2(F9.3,I4),F8.2)

Tuleb arvestada, et sulgudesse võetav väljakirjeldajate rühm tohib omakorda sisaldada sulgudes oleva rühma, kuid selline sisaldamine ei tohi olla enam kui kahekordne. Näiteks formaadikirjeldus

10 FORMAT (I3,2(X4,2(I5,2(I2,I1))),A8)

osutub juba vigaseks, küll aga on lubatav formaadikirjeldus

```
10 FORMAT (I3,2(X4,2(I5,2I2),A8))
```

Ka korduvaid tekstikonstante võib ühendada rühmadesse, näiteks formaadikirjelduse ja trükkimisdirektiivi

```
14 FORMAT (1H+,12Ø(1H-))  
PRINT 14
```

toimel väljastatakse trükiseadmél (ilma paberi edasilükketa) üle terve lehe ulatuv joon.

Tekstikonstantide mitmesuguste rühmade kasutamine formaadikirjeldustes osutub eriti vajalikuks just tabelite jms. trükiste vormistamisel. Näiteks programmitkendi

```
PRINT 21  
PRINT 22  
PRINT 23  
PRINT 21  
PRINT 22  
21 FORMAT (10X,55(1H-))  
22 FORMAT (10X,7(1HI,8X))  
23 FORMAT (1H+,8X,3(4X,4HNIMI,5X,5HSUMMA))
```

toimel saadakse järgmine trükipilt:

```
-----  
I NIMI I SUMMA I NIMI I SUMMA I NIMI I SUMMA I  
-----  
I I I I I I I I
```

#### 1.4. Formaadiga informatsioonivahetus

Formaadikirjelduse igale väljakirjeldajale (peale teisenduste nimedega T, X ja H ning apostroofide vahel oleva tekstikonstandi) vastab seda kirjeldust kasutava sisestamisvõi väljastamisdirektiivi objektide loetelus reeglina üks muutuja. Erandi moodustab kompleksarvuline muutuja, mis vastab formaadikirjelduse kahele järjestikusele väljakirjeldajale (lubatavad on sealjuures vaid nimed F, E, D, Q ja G).

Formaadikirjelduste kasutamisel tuleb hoolitseda selle eest, et iga väljakirjeldaja oleks kooskõlas talle objektide loetelus vastava muutuja tüübiga. Igasuguse muutujaga peale kompleksarvulise sobivad teisendused A ja Z ning igasugusega G, kuid ülejäänud teisendused nõuavad alati kindlat tüüpi muutujat. Näiteks programmitõlgend

```
WRITE (8,40) A,B,C  
40 FORMAT (L2,X3,F8.2,',',I7)
```

osutub korrektseks vaid siis, kui A on loogiline, B reaalarvuline ja C täisarvuline muutuja.

Üksusest vastavust väljakirjeldajate ja objektide loetelu vahel (nagu äsjatoodud näites) üldiselt ei tarvitse olla: vastavuse puudumisel on määravaks objektide loetelu. Nimelt kui objektide loetelu osutub väljakirjeldajate loetelust lähemaks, siis viimaseid väljakirjeldajaid lihtsalt ignoreeritakse. Kui aga objektide loetelu osutub pikemaks, siis hakatakse väljakirjeldajaid kordama alates formaadikirjelduse

kõige viimasest avavast sulust (kordamisele kuuluvate rühmade puudumisel on selleks kogu loetelu alustav sulg). Seoses sellega tuleb formaadikirjelduses vahel eraldada väljakirjeldajate rühmi kordamiste arvuga üks (mis jäetakse sulgude ette kirjutamata). Tuleb aga arvestada, et iga kord, kui väljakirjeldajate loetelus jõutakse formaadikirjelduse lõpusuluni, toimub ühtlasi üleminek uuele kirjele (sellise ülemineku tunnuseks on muidu kaldkriips eraldaja koma asemel). Näiteks kui formaadikirjeldust

```
15 FORMAT (E7.2,2(F9.3,I4),2HX=,(F8.2,I2))
```

kasutava direktiivi objektide loetelu sisaldab kümme elementi, siis esimest kirjet iseloomustavad väljakirjeldajad

```
E7.2,F9.3,I4,F9.3,I4,2HX=,F8.2,I2 ,
```

teist kirjet väljakirjeldajad

```
F8.2,I2
```

ja kolmandat väljakirjeldaja

```
F8.2 :
```

Kui programm sisaldab näiteks formaadikirjelduse ja väljastamisdirektiivi:

```
10 FORMAT (8H0X JA Y=,F8.2,F10.4/(8X,F8.2,F10.4))
```

```
.....
```

```
WRITE (6,10)(X(I),Y(I),I = 1,3)
```

siis selle direktiivi toimel (eeldades, et muutujatel on vastavad väärtused) saadakse trükiseadmes kõigepealt tühi rida ja edasi järgmine kirjapilt:

```
XJJAY = 100.000001.0025  
00000 0 200.000001.0035  
00000 0 300.000001.0045
```

(tühikute asemel on siin kirjutatud 0 vaid positsioonide äranäitamiseks; meenutame ühtlasi, et iga rea esimene sümbol on juhtsümbol, seega number 0 ja hiljem esimene tühik jääb trükkimata). Märkime veel, et näites toodud väljastamisdirektiivi võib asendada samaväärse trükkimisdirektiiviga

```
PRINT 10,(X(I),Y(I),I=1,3)
```

Kui formaadiga informatsioonivahetuse korral objektide loetelu üldse puudub, siis kestab informatsioonivahetus seni, kuni formaadikirjelduses jõutakse niisuguse väljakirjedajani, millele peaks vastama loetelu järjekordne element. Näiteks programmilõigu

```
37 FORMAT (15HSOBIMATU,LAHEND,I5)  
READ (5,37)
```

toimel loetakse kaartsisendilt üks kaart ja kasutatakse selle esimesed viisteist sümbolit formaadikirjelduse 37 esimese väljakirjedajaga määratud tekstikonstandi muutmiseks - kirje järgmised sümbolid jäävad kasutamata.

Kui viimase näite formaadikirjeldus asendada järgmisega:

```
37 FORMAT (I5/15HSOBIMATU, LAHEND)
```

siis sama direktiiv teostab ainult ühe kaardi tühisestuse.

Formaadikirjelduse osatähtsust objektide loetelu olemas-  
olu korral illustreerib järgmine kaheosaline näide, milles  
eeldame, et kaartsisendil paiknev fail (standardse faili-  
numbriga 5) koosneb kahest perfokaardist:

1. kaart: LKKKKK,008001400-203.20E-1E,077.L,0+14

2. kaart: L,0001.0017, -1111E-01++L,1435.050378,7

Selle faili sisestamine programmilõiguga

```
COMPLEX C(10)  
REAL R(4)  
READ (5,50) C(3),R  
50 FORMAT (5X,3(2X,F9.3))
```

annab muutujatele C(3), R(1), R(2), R(3) ja R(4) vastavalt  
väärtused (8001.4,20.32), -770.0, 1.0017, 1111.1 ja 1435.05.

Kui aga sama faili sisestamiseks kasutada programmilõiku

```
INTEGER KK(8)  
READ (5,60) KK  
60 FORMAT (3H,MM,2(7X,I5,1X,I5))  
WRITE (6,60) KK
```

siis omandavad muutujad KK(1),...,KK(8) vastavalt väärtused  
140, -203, -77, 14, 170, 1111, 1435 ja 5037, kusjuures trü-

kiseadmél väljastatakse järgmine tekst:

KK            140 -203            -77    14  
                 170 1111            1435 5037

Informatsioonivahetuseks vajaliku väljakirjeldajate loetelu saab esitada ka ilma formaadikirjelduseta. Nimelt võib selle loetelu (koos teda ümbritsevate sulgudega) tekstikons-tandina omistada mingi massiivi esimeste muutujate väärtu-seks. Niisugusel juhul tuleb seda loetelu kasutavas sises-tamis- või väljastamisdirektiivis formaadikirjelduse märgen-di f kohale kirjutada vastava massiivi identifikaator.

Väljakirjeldajate loetelu omistamine massiivile võib toimuda kas sisestamisdirektiiviga väliskandurilt või alg-väärtustamiskirjeldusega. Esimest moodust kasutatakse peami-selt siis, kui programmis esineva informatsioonivahetusega kaasnev redigeerimisviis peab sõltuma algandmetest ja seda ei saa järelikult vahetult programmis esitada. Illustreerime seda võimalust järgmise tingliku näitega.

Olgu perfokaardile (alates esimesest positsioonist) perforeeritud 22-sümboliline tekst:

(E12.2,F8.2,I7,2E20.3) ,

mida soovitakse kasutada tulemuste väljatrükkimisel. FORTRAN-programmi vastavad osad võib siis kirjutada näiteks kujul:

```
DIMENSION F(8)
10 FORMAT (6A4)
READ (5,10)(F(I),I=1,6)
PRINT F,A,B,K,C,D
```

Ilma eraldi perfokaarti kasutamata võib väljakirjeldajate sama loeteluga väljastamise teostada ka järgmiselt:

```
DOUBLE PRECISION G(3)
.....
DATA G/22H(E12.2,F8.2,I7,2E20.3)/
.....
PRINT G,A,B,K,C,D
```

kuid praktikas on sellist moodust tarvis peamiselt siis, kui programmi täitmise käigus osutub vajalikuks formaadikirjelduse osaline muutmine. Niiisuguseks muutmiseks tuleb formaadikirjeldust sisaldavas massiivis teha töö käigus mõningaid formaalseid omistamisi. Näiteks programmilõigu

```
INTEGER S(7) / '(1H□,T20,2HA(,I1,3H)□=,F8.3) ' /
REAL A(5)
.....
DO 1 I=1,5
WRITE (6,S)I,A(I)
1 S(2)= S(2) + 2
```

toimel (eeldades, et muutujad A(1), ..., A(5) on punktiiriga asendatud osas omandanud vastavad väärtused) saadakse trüki-seadmes järgmine kirjapilt:

```
A(1) = 125.417
A(2) = -1.030
A(3) = 40.532
A(4) = 0.001
A(5) = -450.333
```

## 1.5. Informatsioonivahetus nimistu või loetelu järgi

Jadafailide korral on formaadiga ja formaadita informatsioonivahetuse kõrval lubatud veel selline täiendav moodus, milles nii objektide loetelu kui ka formaadikirjelduse asemel erilise nimistukirjelduse abil teatatakse informatsioonivahetusest osa võtvate muutujate ning massiivide loetelu.

Nimistukirjeldus algab reserveeritud sõnaga NAMELIST, millele järgneb kirjeldatavate nimistute loetelu (nimistute vahele selles loetelus mingit eraldajat ei panda). Iga nimistu kohta antakse kaldkriipsude vahel, kõigepealt nimistu nimi (moodulis lokaliseeritud identifikaator) ja selle järel nimistusse kuuluvate lihtmuutujate või massiivide loetelu (komadega eraldatult).

Üks ning sama muutuja võib vajaduse korral kuuluda mitmesse nimistusse, kuid sama nimega nimistut ei tohi kirjeldada üle ühe korra (ei samas ega mõnes teises nimistukirjelduses). Nimistukirjeldus võib paikneda mooduli mistahes kohas, kuid enne esimest sisestamis- või väljastamisdirektiivi, mis kasutab mõnd kirjeldatavat nimistut.

Ühe direktiivi toimel vahetatakse nimistu järgi alati formaalselt üks kirje, kuigi süsteem võib informatsiooni vahetuse korral automaatselt jaotada mitmeks füüsiliseks kirjeks (kui ei mahu näiteks perfokaardile või trükireale). Programmeerijal pole seda asjaolu aga eriti vaja arvestada - ta võib lihtsalt lugeda, et kirje pikkus pole niisuguse informatsioonivahetuse korral piiratud (samuti nagu formaadita sisestamise või väljastamise korral).

Sisestamisel nimistu järgi peavad andmed väliskanduril olema spetsiaalselt vormistatud. Iga kirje esimeseks sümboliks peab olema tühik. Korraga sisestatava kirjeterühma esimese kirje teiseks sümboliks peab olema &, selle järel nimistu nimi ja siis veel üks tühik. Rühma viimane kirje peab lõppema sümbolitega &END (mille järel võib olla tühikuid).

Kirjete sisu kujutab endast andmeelementide loetelu (komadega eraldatult). Iga andmeelement kujutatakse väliskanduril võrdusena, kus kas vasakul on muutuja nimi ja paremal konstant, või vasakul on massiivi nimi ja paremal konstantide loetelu nagu algväärtustamiskirjelduses (vt. PKP 36; võib ühendada korduvaid konstante kujul  $n * k$ ). Tühikud arvu lõpul loetakse nullideks, tõeväärtused võib esitada kujul T ja F.

Andmeelementides kasutatavad nimed peavad kõik esinema vastava nimistu kirjelduses (keelatud on isegi asendamine ekvivalentse muutuja nimega). Oluline pole aga nimede järjekord, samuti ei tarvitse andmetes sisalduda kõik nimistus toodud nimed. Kui nimistus on antud massiivi nimi, siis võib andmetes kasutada ka selle massiivi indeksitega muutujate nimesid (konstantidega indeksite kohal).

Sisestamist nimistu järgi on sobiv kasutada näiteks nn. hõredate maatriksite täitmisel, kus vaid üksikud elemendid erinevad nullist. Näiteks kui kaartsisendil paiknevad perfokaardid

1. kaart:  $\begin{matrix} \square & \&NL1 & \square \\ \square & K(1,2)=3, & A(2,1)=4.2, & K(3,1)=\square-2, & A(4,2)=.1, \end{matrix}$

2. kaart:  $\begin{matrix} \square & K(4,3)=1\square, & K(2,4)=\square7, & A(3,4)=-7.5, & \&END \end{matrix}$

siis programmilõigu

```

DIMENSION A(4,6),B(12),K(4,4)
DATA A,K/24*0.0,16*0/
NAMBLIST /NL1/A,B,K/NL2/F,K
READ (5,NL1)
. . . . .

```

toimel saame maatriksitele A ja K järgmised väärtused:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 4,2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -7,05 & 0 & 0 \\ 0 & 0,1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad K = \begin{pmatrix} 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 7 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 \end{pmatrix}.$$

Väljastamisel nimistu järgi saadakse informatsioon väliskanduril niisugusel kujul, et seda võib vajaduse korral sama nimistu järgi sisestada (muidugi kui välisseade seda võimaldab!). Massiivid väljastatakse sealjuures samas järjekorras, nagu vastavad indeksitega muutujad sisemälus paiknevad (vt. PKP 21). Näiteks kui viimatitoodud programmilõigule lisada veel direktiiv

```
WRITE (6,NL2) ,
```

siis saadakse järgmine trükis (igast trükireast on siin esitatud vaid 60 esimest sümbolit):

```

&NL2
F= 6.50000000 ,K= 0, 0, -2,
    0, 0, 0, 10, 0
&END

```

Nimistu järgi informatsioonivahetusega mõneti sarnane on ka informatsioonivahetus loetelu järgi: nimistu aset täidab objektide loetelu, mis ühtlasi kaudselt määrab ka formaadi.

Sisestamine loetelu järgi toimub direktiiviga

```
READ (n,*) 1
```

kus n tähendab ikka failinumbrit. Selle direktiivi toimel sisestatakse formaalselt üks kirje (mis võib jälle jaotuda mitmeks füüsiliseks kirjeks), tõlgendatakse selle koostisosi objektide loetelu l elementidega määratud tüüpi konstantide-na ja omistatakse need konstandid loetelu elementideks olevate muutujate väärtusteks. Sisestatav kirje peab sel juhul kujutama endast tühikute või komadega eraldatud konstantide jada (perfokaartidelt sisestamisel on eraldajaks ka kaardi lõpp). Kaks järjestikust koma tähendavad konstandi puudumist, s.t. loetelu järjekordne element jääb väärtustamata. Konstantide arv kirjes tohib olla väiksem elementide arvust objektide loetelus (siis peab viimase konstandi järel paiknema sümbol / ), kuid ei tohi olla sellest suurem.

Väljastamine loetelu järgi toimub direktiiviga

```
WRITE (n,*) 1
```

mis väljastab loetelus antud muutujate väärtustest koosneva kirje (see on sisestatav sama struktuuriga loetelu järgi).

Tuleb arvestada, et kirjetagurdamise direktiivi ei tohi loetelu järgi informatsioonivahetusel kasutada.

## 1.6. Informatsioonivahetus otsefailidega

Otsefaili iga kirjega seotakse süsteemselt kirjenumber, mis võimaldab ilma mingi ettevalmistuseta pöörduda otsekohe teatud numbriga kirje poole (jadafaili korral tuleks näiteks enne viiendana väljastatud kirje taassisestamist sooritada failitagurdamine ning veel neli tühisestust).

Selleks, et operatsioonisüsteem tõlgendaks mingile failinumbrile vastavat andmekogumit otsefailina, peab programmi esinema vastav kirjeldus - kõik kirjeldamata failid loetakse automaatselt jadafailideks. Loomulikult tuleb otsefaili numbrile vastavaks välisseadmeks lahendamistellimuses määrata ketasmälu. Standardsete määrangute säilitamiseks on soovitatav mitte kirjeldada otsefailidena faile numbritega 5 ja 6. Iga otsefail tuleb FORTRAN-programmis kirjeldada vaid üks kord (põhiprogrammis või mõnes protseduuris), sest sama faili järgnevald kirjeldusi ignoreeritakse. Soovitatav on koondada need kirjeldused mooduli algusesse, kuid igal juhul peavad nad paiknema enne pöördumisi vastavate failide poole.

Failikirjeldus algab reserveeritud sõnaga DEFINE FILE, millele järgneb kirjeldatavate failide loetelu (komadega eraldatult). Iga fail selles loetelus kirjeldatakse kujul

$n(a,p,v,s)$  ,

kus  $n$  on failinumber,  $a$  - kirjete arv failis,  $p$  - kirje maksimumpikkus,  $v$  - vahetustunnus ja  $s$  - seosemuutuja nimi ( $n$ ,  $a$  ja  $p$  esitatakse märgita täisarvuliste konstantidena).

Vahetustunnusena v võib kasutada üht kolmest tähest: E, U või L. Tunnus E tähendab, et kirjeldatava faili korral on lubatud üksnes formaadiga informatsioonivahetus; tunnus U lubab üksnes formaadita, tunnus L aga nii formaadiga kui ka formaadita informatsioonivahetust. Tunnuste E ja L korral tuleb kirje maksimaalpikkus p esitada baitides, tunnuse U korral aga sõnades (sõna on neli järjestikust baiti).

Seosemuutuja on täisarvuline lihtmuutuja, mille väärtuseks pärast iga sisestamist või väljastamist sellesse faili süsteem omistab viimasena töödeldud kirjele järgneva kirje numברי (otsimisdirektiivi korral otsitud kirje numברי).

Näiteks failikirjeldus

DEFINE FILE 8(5Ø,12Ø,L,I8),9(1ØØ,5Ø,U,I9)

teatab, et faille 8 ja 9 tõlgendatakse vastavas programmis otsefailidena. Esimene neist (fail 8) võib sisaldada kuni 50 120-baidilist kirjet, kuhu informatsiooni saab salvestada nii formaadiga kui ilma. Fail 9 võib sisaldada kuni 100 200-baidilist (50-sõnalist) kirjet, kuhu informatsiooni saab salvestada ainult formaadita. Kirjeldatud failide seosemuutujateks on vastavalt I8 ja I9.

Otsefailide korral informatsioonivahetust teostavad direktiivid erinevad jadafailide vastavatest direktiividest (vt. p. 1.2) tegelikult ainult selle poolest, et peale failinumbri tuleb neis tingimata näidata veel ka kirjenumber. Nimelt kui tegemist on otsefailiga, siis tuleb sisestamisdirektiiv esitada kujul

READ (n'k,f,e) l

kus n on failinumber (ikka kas märgita konstant või täisarvuline lihtmuutuja), k - kirjenumbrit määrav täisarvuline aritmeetiline avaldis, f - formaadikirjelduse märgend (või väljakirjeldajate loetelu sisaldava massiivi identifikaator), e - veaväljund (lõpuväljundil pole nüüd mõtet) ja l - objektide loetelu (mis ei tohi sisaldada faili n seosemuutujat).

Elemendid f, e või l võivad konkreetsetes sisestamisdirektiivides ka puududa. Nii näiteks formaadikirjelduse märgendi f puudumine tähendab, et tegemist on formaadita sisestamisega (selline sisestamisviis osutub lubatavaks muidugi vaid siis, kui vastavas failikirjelduses on antud vahetus-tunnus U või L). Objektide loetelu l puudumine tähendab, et sisestatakse redigeerimisteksti, sest tühisestuseks pole otsefailide korral vajadust.

Tuleb silmas pidada, et k määrab esimesena sisestatava kirje numbri. Kui kasutatav formaadikirjeldus nõuab suurema arvu kirjete sisestamist, siis igal üleminekul uuele kirjele suureneb kirjenumber ühe võrra. Näiteks programmilõigu

```
DEFINE FILE 3(200,100,E,I)
DIMENSION A(20)
1 FORMAT (//10F10.2)
READ (3'7,1) A
```

toimel sisestatakse failist 3 kirjed 9 ja 12. Seosemuutuja I omandab sisestamisdirektiivi täitmise lõpul väärtuse 13.

Väljastamisdirektiiv esitatakse otsefaili korral kujul

WRITE (n'k,f) l

kus metasümbolite tähendused on samad mis sisestamisdirektiivi juhul (f või l võib konkreetses direktiivis ka puududa). Näiteks programmilõik

```
DEFINE FILE 9(15Ø,24Ø,L,K)
3 FORMAT (2ØØ12.4)
DIMENSION F(6Ø)
READ (9'1,3) F
WRITE (9'K+1ØØ) F
```

loeb faili 9 esimesest kolmest kirjest kuuskümmend reaalarvu ja väljastab need ilma formaadita sama faili sajaneljandasse kirjesse (massiivi F muutujad on kirjeldatud 4-baidilistena, järelikult mahub massiiv ära ühte 240-baidilisse kirjesse).

Sisestamisel otsefailidest võib programmi töö kiirendamiseks juba eelmisena sisestatud kirje töötlemise ajal anda korralduse järgmisena sisestatava kirje leidmiseks. Seda saab teha nn. otsimisdirektiiviga, mille kuju on

FIND (n'k)

kus n tähendab ikka faili- ja k kirjenumbrit. Tuleb arvestada, et otsimisdirektiivi toimel saab faili n seosemuutuja väärtuseks leitud kirje number k. Enne järjekordset väljastamisdirektiivi pole otsimisdirektiivi mõtet kasutada, sest programmi töö sellest ei kiirene.

## 1.7. Asünkroonne informatsioonivahetus

Senivaadeldud informatsioonivahetusdirektiivid töötavad samuti nagu vaadeldava keele kõik teised direktiivid: alles pärast ühe direktiivi töö lõppemist tuleb täitmisele järgmine. Formaadita informatsioonivahetuseks jadafailidega on aga täiendavalt ette nähtud veel direktiivide asünkroonse täitmise võimalus - informatsioonivahetusega samaaegselt täidetakse programmi edasisi direktiive. Selle võimaluse tõttu on senivaadeldud sisestamis- ja väljastamisdirektiive loomulik nimetada sünkroonne informatsioonivahetuse direktiivideks.

Asünkroonse sisestamise või väljastamise korral tuleb programmis täpselt piiritleda nn. asünkroonsuspiirkond, s.t. programmilõik, millesse kuuluvaid direktiive tohib täita samaaegselt informatsioonivahetusega. See piirkond algab alati informatsioonivahetusdirektiivist ja lõpeb erilise ootamisdirektiiviga. Kui ootamisdirektiivini jõutakse enne informatsioonivahetuse lõppemist, siis jääb programm ootama ning töö jätkub alles pärast informatsioonivahetuse lõppu.

Faililõpetamise, samuti faili- või kirjetagurdamise direktiive asünkroonsuspiirkonnas esineda ei tohi. Neid andmeid, mida asünkroonselt sisestatakse, ei saa asünkroonsuspiirkonnas kasutada, ka ei ole seal võimalik muuta asünkroonselt väljastatavaid andmeid. Asünkroonsuspiirkonnas on keelatud veel sünkroonse informatsioonivahetuse direktiivid, kuid seal võib alustada asünkroonsel informatsioonivahetust mõne teise failiga. Seega erinevate failide korral võivad asünkroonsuspiirkonnad lõikuda.

Asünkroonne sisestamisdirektiiv esitatakse kujul

READ (n, ID=k) l ,

kus n on failinumber, k aga nn. direktiivinumber - täisarvuline avaldis, mille väärtus identifitseerib selle konkreetse direktiivi (ekstraväljundite kirjeldused pole siin lubatud).

Objektide loetelu l määrab sisestamispiirkonna, mis tohib nüüd koosneda vaid ühe massiivi muutujatest. Seetõttu tuleb l esitada ühel järgmisest neljast kujust.

Ainult massiivi nimest m koosnev objektide loetelu tähendab, et sisestamispiirkonnaks on kogu see massiiv. Loetelu kujul  $m(i_1) \dots m(i_2)$  määrab sisestamispiirkonnaks massiivi lõike, alates indeksitega muutujast  $m(i_1)$  kuni indeksitega muutujani  $m(i_2)$  (kaasa arvatud). Kui  $m(i_1)$  tähendab massiivi esimest muutujat või  $m(i_2)$  viimast muutujat, siis võib kasutada vastavalt lihtsustatud loetelusid  $\dots m(i_2)$  või  $m(i_1) \dots$ .

Vaadeldava direktiivi tolmel sisestatakse üks kirje, mis jaotub massiivi tüübis määratud pikkusega väljadeks ja omistatakse need ilma teisenduseta sisestamispiirkonna muutujate väärtusteks. Sisestamine lõpeb kirje või sisestamispiirkonna ammendamisega: esimesel juhul võib viimane muutuja saada pooliku "väärtuse", järgmiste väärtusi aga ei muudeta. Loetelu l puudumisel toimub tühisestus (vt. lk. 10).

Asünkroonne väljastamisdirektiiv esitatakse kujul

WRITE (n, ID=k) l ,

kus ainsa erinevusena sisestamise juhust l määrab nüüd väl-

jastamispiirkonna. Selle direktiivi toimet väljastatakse üks kirje, mille pikkuse määrab väljastamispiirkond. Piirkonna muutujate väärtused kirjutatakse kirjesse järjest ilma mingite teisendusteta.

Nii asünkroonse sisestamis- kui ka väljastamisdirektiivi korral peab programmis tingimata näitama vastava asünkroonsuspiirkonna lõpu. Selleks lisatakse ootamisdirektiiv

WAIT (n,p) l

 ,

kus p tähendab parameetrite loendit. See loend sisaldab alati direktiivinumbril kujul  $ID=k$ , kusjuures k väärtus peab ühtima asünkroonsuspiirkonda alustanud direktiivnumbriga. Lisaks võivad p koosseisu kuuluda veel elemendid kujul  $COND=t_1$  ja  $NUM=t_2$ , kus  $t_1$  ning  $t_2$  on täisarvuliste muutujate nimed. Muutuja  $t_1$  väärtuseks saadakse informatsiooni vahetuse lõpetamistunnus: väärtus 1 tähendab normaalset lõppu, väärtus 2 tõrkesituatsiooni ja väärtus 3 faili lõpuni jõudmist. Muutuja  $t_2$  väärtuseks saadakse sisestatud baitide arv (väljastamisel pole seda elementi mõtet kasutada).

Objektide loetelu l võib ootamisdirektiivis puududa. Kui aga l lisatakse, siis peab ta määrama täpselt sama piirkonna nagu vastav sisestamis- või väljastamisdirektiiv (mis muide tohib paikneda ka hoopis teises moodulis).

Ootamisdirektiivis ei tohi direktiivinumbril k avaldises (nagu asünkroonsuspiirkonnaski) kasutada sisestamispiirkonna muutujaid. Ka ei või seal pöörduda nende funktsioonide poole, mida kasutatakse indeksite arvutamisel objektide loetelus.

Asünkroonsuspiirkonnas ei lubata (peale sisestamis- ning väljastamispiirkonna muutujate) otseselt ega mingi ühispiirkonna- või ekvivalentuskirjelduse, või siis mingi alamprogrammi parameetrite kaudu muuta objektide loetelus indekseid määravaid muutujaid. Kui nende indeksite avaldistes esineb väärtustatud funktsioon, siis ei tohi sama funktsiooni poole asünkroonsuspiirkonnas pöörduda.

Mitme asünkroonse informatsioonivahetuse üheaegsel kasutamisel ei tohi nende sisestamis- või väljastamispiirkonnad kattuda. Näiteks programmikatkendiga

```
K = 0
10 READ (7, ID=10) ...A(50)
   IF (K.EQ.0) GO TO 30
20 WRITE (8, ID=20) B(21)...B(70)
30 WAIT (7, ID=10, COND=I1)
   GO TO (40, 100, 110), I1
40 ----- (massiivi A muutmine)
   IF (K.EQ.0) GO TO 80
50 WAIT (8, ID=20, COND=I2)
   IF (I2.GT.1) GO TO 120
60 DO 70 L=1,50
70 B(20+L)=A(L)
   GO TO 10
80 K = 1
   GO TO 60
```

sisestatakse teatud arv kirjeid, töödeldakse neid ja väljastatakse teise faili. Alates teisest kirjest toimub sisestamine ja väljastamine üheaegselt. Tegevus tõrkesituatsioonide (märgendid 100, 120) ja faili 7 lõppemise (märgend 110) korral on selles näites täpsustamata jäetud.

## 2. OPERATSIOONISÜSTEEMI KASUTAMISEST

### 2.1. Transleerimine

Koostatud FORTRAN-programmi kasutamisele arvutil eelneb selle programmi kõigi moodulite (ja vajaduse korral ka algandmete) perforeerimine. Perforeerimiseks tuleb FORTRAN-programm esitada vastavatele blankettidele vormistatud kujul.

FORTRAN-blankett (vt. lk. 45) sisaldab 20 rida, igas reas 72 positsiooni. Ühte ritta kirjutatakse ülimalt üks kirjeldus või direktiiv. Kui see kuulub märgendamisele, siis märgend kirjutatakse positsioonidesse 1-5. Positsioonis 6 (blanketil JT) paikneb konstruktsiooniühiku jätkamise tähis - kui sinna kirjutada nullist või tühikust erinev sümbol, siis seda rida loetakse eelmise rea järjeks. Tavaliselt kirjutatakse jätkamise tähisteks just jätkude järjekorranumbrid.

FORTRAN-blanketi positsioonidesse 7-72 kirjutatakse kirjeldus või direktiiv. Tuleb arvestada, et nendes positsioonides tühikul ei ole tähendust (välja arvatud tekstikonstandi koosseisus) ja teda võib kasutada suvaliselt kirjapiiri kujundamiseks. Programmi loetavuse parandamiseks võib aga blanketi mistahes ritta kirjutada veel kommentaare, mille



cunnuseks on täht C positsioonis 1 (transleerimisel jäetakse need read lihtsalt vahele).

FORTTRAN-blanketi päisesse tuleb kirjutada šiffer (arvutuskeskuse poolt programmeerijale ühe probleemi lahendamise ajaks antud tinglik nimi) ja lehe järjekorranumber. Programmi ning koostaja nimed märgitakse tavaliselt vaid esimesele lehele. Soovitav on alustada iga mooduli kirjutamist uuel lehelt. FORTRAN-blankettide puudumisel võib programme kirjutada ka 80-positsioonilistele ASSEMBLER-blankettidele, kus 8 viimast positsiooni tuleb igas reas vabaks jätta.

FORTTRAN-programm perforeeritakse perfokaartidele - blanketi igale reale vastab üks perfokaart. Siitpeale jääb programmi blankettidele kirjutatud tekst programmeerijale arhiivdokumendiks, kogu edasine töö toimub saadava perfokaartide pakiga (mõnikord perforeeritakse programm küll kaartide asemel perfolindile, kuid seda võimalust me siin ei vaatle).

Programmi sisaldavale kaardipakile tuleb alati lisada veel operatsioonisüsteemi poole pöördumise juhtlauseid omanette perfokaartidena. Niiviisi moodustatud lahendamistellimuse esimeseks kaardiks on tingimata töölause, mis TRÜ arvutuskeskuses kehtestatud korra kohaselt peab sisaldama töö nimetuse (soovitavalt tellija nimi või selle nime mingi lühend), šifri ja tellija täieliku nime (kuni 20 sümbolit). Seega töölause võib esitada näiteks kujul:

//KAASIK JOB B-333,Y.KAASIK

FORTTRAN-programmi transleerimine tööprogrammiks toimub ühega operatsioonisüsteemi OS kahest võimalikust translaatorist, mida järgnevas nimetame translaatoriteks G ja H.

Sarja "Programme kõigile" 1980.a. väljaandes "Algoritmi-keele FORTRAN kasutamine arvutil EC-1022" kirjeldati samanimelisi translaatoreid, kuid järgnevas vaadeldakse nende uusi versioone, mida kirjanduses sageli nimetatakse ka vastavalt SE ja OE. Translaator G (ehk SE) on n.õ. silumistranslaator, mis transleerib kiiresti, kuid annab suhteliselt aeglaselt töötavaid tööprogramme. Seda translaatorit kasutatakse eeskätt siis, kui saadava tööprogrammiga kavatakse lahendada vaid lihtsaid kontrollülesandeid programmi silumise käigus.

Optimiseeriv translaator H (ehk OE) töötab küll aeglasemalt, kuid saadavad tööprogrammid osutuvad märksa kiiremateks ning ratsionaalsemateks, kui translaatoriga G saadud. Seetõttu kasutatakse translaatorit H peamiselt alles pärast programmi õigsuses veendumist, tegeliku ülesande lahendamiseks määratud tööprogrammi saamisel.

Translaatorite poole pöördumiseks tuleb lahendamistellimusse lisada vastavad juhtlauseid, mis määravad translaatori nime ja tema töörežiimi täpsustavad parameetrid. TRÜ arvutuskeskuses on selliste juhtlausete esitamise lihtsustamiseks koostatud terve rida standardseid juhtprotseduure - juhtlausete komplekte, mida saab lahendamistellimusse lülitada üheainsa juhtlausega. FORTRANiga seotud juhtprotseduuride nimedes on esitäheks alati F (=FORTRAN), teiseks täheks üldiselt translaatori nimi. G või H ja edasi vajalik kombinatsioon tähtedest C (=COMPILE), L (=LINKEDIT) ning G (=GO) vastavalt sellele, kas soovitakse transleerida, komplekteerida või saadud tööprogrammi täita. Üliõpilastele määratud juhtprotseduuride korral tuleb nime ette lisada veel täht S.

Asjaperforeeritud programmi töötlemine algab enamasti just nn. kontrolltransleerimisega, mille eesmärgiks on üksnes perforeerimis- ja süntaksivigade avastamine. Saadava tööprogrammi säilitamist, komplekteerimist või täitmist sellisel juhul ei nõuta ja tellimus esitatakse näiteks järgmise struktuuriga kaardipaki kujul:

```
//KAASIK14 JOB B-333,Y.KAASIK
// EXEC FGC
//SYSIN DD *
(kaardid FORTRAN-programmi tekstiga)
//
```

kus kolmanda juhtlause (mis teatab, et järgnev informatsioon paikneb kaartsisendil) võib ka ära jätta - selle lisab siis operatsioonisüsteem. Viimane, ainult kaht kaldkriipsu sisaldav kaart on töö lõpu tunnuseks.

Sellist vahetult perfokaartidelt transleerimist kasutatakse praktikas siiski üksnes erandjuhtudel. Suurte kaardipakkidega opereerimise vältimiseks on hoopis loomulikum lasta programmi tekst kõigepealt teekida ja edasises töös kasutada vaid seda teegitud teksti.

Tekstide teekimiseks (ja samuti nende hilisemaks korrigeerimiseks) tuleb kasutada TRÜ arvutuskeskuse standardset protseduuri UPDATE või vastavat üliõpilastele määratud varianti nimega UPDATES. Selle protseduuri abil teegitakse tavaliselt ühe nime all korruga kõik vaadeldava programmi koosseisu kuuluvad moodulid. Peab aga tingimata arvestama, et TRÜ arvutuskeskuses tohib teekimisel kasutada üksnes vastavale kasutajale määratud prefiksiga algavaid nimesid.

Näiteks kui prefiksiks on määratud täht X, siis perfo-  
kaartidel paikneva programmi teekimiseks nime XS52 all ja  
samas töös toimuvaks kontrolltransleerimiseks võib esitada  
tellimuse kujul:

```
//KAASIK15 JOB B-333,Y.KAASIK  
// EXEC UPDATE  
./ A XS52  
./ N 100  
(kaardid FORTRAN-programmi moodulitega)  
// EXEC FGC,LIB=TEX,  
// PARM.FORT=MAP,  
// NAME=XS52  
//
```

Selle tellimuse esimese sammu toimel kirjutatakse prog-  
ramm standardsesse tekstide teeki nimega SYS1.TEXTLIB (või  
üliõpilastele määratud protseduuri UPDATES kasutamise korral  
teeki SYS1.STDLIB). Koos teeki kandmisega toimub automaat-  
selt ka kaartide nummerdamine alates numbrist 100, kusjuures  
nummerdamissammu väärtuseks on samuti 100. Selles näites (ja  
ka edaspidi) on kasutatud protseduuri UPDATE juhtlausete lü-  
hendatud kirjutusviisi: näiteks lause ./ A XS52 asendab te-  
gelikult pikemat lauset ./ ADD NAME=XS52.

Tellimuse teisel sammul toimub teegitud teksti kontroll-  
transleerimine, kusjuures parameetri väärtuse MAP lisamisega  
teatatakse, et transleerimisprotokollis soovitakse saada ka  
mälujaotus. Olgu märgitud, et teegi nime teatamine (LIB=TEX)  
on siin tarbetu, sest nime SYS1.TEXTLIB (või SYS1.STDLIB) võ-  
tab standardne juhtprotseduur vaikimisi (küll aga peab juht-  
lause samas kohas näitama igä teise teeginime vastava lõike).

Transleerimise käigus trükitav transleerimisprotokoll sisaldab kõigepealt moodulite kaupa FORTRAN-programmi täieliku teksti, kus iga rea ees paikneb translaatori poolt lisatud neljakohaline järjekorranumber (prefiksiga ISN), taga aga teekimisel pandud kaheksakohaline järjekorranumber.

Kui translaator avastab programmis vea, siis vigasele järgnevasse ritta trükib ta transleerimisprotokollis vastava märkuse (kus näidatakse vigase lekseemi number), protokollilõppu aga väljastab veateated kõigi avastatud vigade kohta.

Iga veateade algab vea numbriga, mis translaatori G korral on prefiksiga IGI, translaatori H korral aga prefiksiga IFE. Numbrile järgneb seda viga iseloomustav ingliskeelne tekst. Translaatori G (ehk SE) veateated (prefiks välja arvatud) langevad kokku lk. 47 alguses nimetatud väljaande lissas loetletutega. Seepärast on käesoleva väljaande lissas 1 toodud ainult translaatori H (ehk OE) veateadete enam-vähem täielik loetelu vea numbrite järjekorras koos vastavate ingliskeelsete tekstide lühitõlgetega. Selles lissas on ühtlasi näidatud iga vea tase, mis iseloomustab vea olulisust programmis. Vead tasemega 0 on tegelikult hoiatused, millega translaator soovib kontrollida programmi vastavat kohta. Ka vea tase 4 ei takista veel transleerimist, kuid arusaamatuse ületamiseks kasutatud standardne tõlgendamisviis ei tarvitse programmeerija kavatsustega kokku langeda. Vead tasemega 8 pole enam standardselt tõlgendatavad ja -annavad tööprogrammi, mis ei kõlba komplekteerimiseks ega täitmiseks. Vead tasemega 16 on aga niisugused, mis muudavad edasise transleerimise üldse võimatuks.

\* K A A S I K 1 5 \*  
 \* \* \* \* \*

```
//KAASIK15 JOB B-333,Y.KAASIK
// BXBC FHC
// NAME=XS52
```

LEVEL 2.3.0 (JUNE 78) OS/360 FORTRAN H EXTENDED DATE 83.031

```
ISN 0002-      INTEGER FUNCTION GCD(M,N)      00000100
ISN 0003      GCD=0      00000200
ISN 0004      IF(M.LE.0.OR.N.LE.0) RETURN      00000300
                C NEGATIVSETE KORRAL NULL      00000400
ISN 0006      GCD=N      00000500
ISN 0007      K=M      00000600
ISN 0008      1 L = MOD(K,GCD)      00000700
ISN 0009      IF(L.EQ.0) RETURN      00000800
ISN 0011      1 K = GCD      00000900
```

ERROR DETECTED - SCAN POINTER = 0

```
ISN 0012      GCD=L      00001000
ISN 0013      GO TO 1      00001100
ISN 0014      END      00001200
```

```
      / GCD/ SIZE OF PROGRAM 000138 HEXADECIMAL BYTES
NAME TYPE ADD. NAME TYPE ADD. NAME TYPE ADD.
  K I*4 00008C L I*4 000090 M I*4 000094
  N I*4 000098 GCD I*4 00009C
```

SOURCE STATEMENT LABELS

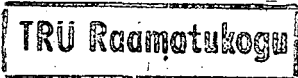
LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
1	-1	0000D8						

COMPILER GENERATED LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
100001	2	0000AC	200001	5	0000C2	400002	5	0000C8
100003	6	0000D2	100004	10	0000EC			

LINE 00000900

```
IFB193I SEVERITY 12(S) ISN 0011 THE STATEMENT NUMBER HAS BE
*STATISTICS* STATEMENTS= 13, PR. SIZE= 312, SUBPR. NAME= GCD
*STATISTICS* 1 DIAGNOSTICS GENERATED, HIGHEST SEV. CODE = 12
*****END OF COMPILATION***** 116K BYTES OF CORE NOT USED
```



Pärast iga moodulit sisaldab transleerimisprotokoll ülevaate saadud tööprogrammi mälujaotuse kohta. See ülevaade koosneb kolmikutest pealkirjadega NAME, TYPE ning ADD., kus esimese all trükitakse moodulis esineva muutuja identifikaator, teise all tema tüüp ja kolmanda all vastav suhtaadress kuueteistkümnendarvuna. Analoožilistes tabelites näidatakse veel moodulis kasutatud märgendite ja samuti ka translaatori poolt lisatud täiendavate märgendite suhtaadressid. Transleerimisprotokollile ühele moodulile vastav osa lõpeb veeteade loeteluga, kus iga avastatud vea korral antakse veeteade ees veel vigase rea number. Lõpuks näidatakse vigade koguarv ja avastatud vigade kõrgeim tase.

Leheküljel 51 on kujutatud (tehnilistel põhjustel veidi moonutatult) transleerimisprotokoll, mis saadi lk. 45 esitatud mooduli kontrolltransleerimisel. Nagu protokollist selgub, on mooduli perforeerimisel tehtud viga - ritta ISNØØ11 lisatud tarbetu märgend 1 on kahjuks juba varem esinenud.

Programmis kontrolltransleerimisel avastatud vigade parandamiseks tarvitseb kaardipakiga opereerimise korral lihtsalt lasta perforeerida vastavad uued kaardid ning panna need pakki vigaste asemele.

Vigade parandamiseks teegitud programmis tuleb kasutada protseduuri UPDATE (üliõpilastel vastavalt UPDATES). Ka nüüd peab perforeerima vaid asendatavad ja/või lisatavad kaardid, kuid seekord juba koos järjekorranumbritega. Järjekorranumber kirjutatakse vahetult vastava direktiivi järele, kusjuures eraldajaks tuleb panna sümbolite kombinatsioon "□" (tingimata üks tühik!).

Vaadeldud näite korral võib vea parandamiseks ja uueks kontrolltransleerimiseks (seekord ilma mälujaotuse trükkimise) koostada lahendamistellimuse:

```
//KAASIK16 JOB B-333,Y.KAASIK
// EXEC UPDATE
./ C XS52
      K = GCD "900
// EXEC PGC;
// NAME=XS52
//
```

Protseduuriga UPDATE saab mitte ainult asendada vigaseid kaarte, vaid ka lisada uusi või kõrvaldada ülearuseid. Kui vaadeldud näitemoodulis soovitakse koos vea parandamisega kaardil 900 kõrvaldada programmis seni olnud kommentaar (kaart 400) ja lisada uus kommentaar lõpukirjelduse (kaart 1200) ette, siis võib viimase tellimuse asemel esitada tellimuse kujul:

```
//KAASIK16 JOB B-333,Y.KAASIK
// EXEC UPDATE
./ C XS52
./ D 400
      K = GCD "900
C GCD=GREATEST COMMON DIVISOR "1150
// EXEC PGC;
// NAME=XS52
//
```

Tuleb silmas pidada, et kõik parandused peavad tellimuses tingimata olema esitatud vastavate kaartide järjekorranumbrite kasvamise järjekorras.

Asjavaadeldud näites sai kahe kaardi vahele lisada uue lihtsalt sobiva järjekorranumbri valimise teel. Mõnikord (näiteks siis, kui on tarvis lisada terve uus moodul olemasolevate moodulite vahele) tuleb aga programmi kahe kaardi vahele lisada nii palju uusi, et eelmisel teekimisel saadud numeratsioon seda ei võimalda. Sellisel juhul saab protseduuriga UPDATE teostada üheaegselt nii uute kaartide lisamise ja nummerdamise, kui ka vanade osalise ümbernumerdamise (lisatavatele kaartidele sealjuures numbreid pole nüüd enam vaja perforerida).

Viimati vaadeldud programmi korral võib direktiivide niisuguse lisamise ja uue kontrolltransleerimise tarvis koostada näiteks järgmise struktuuriga tellimuse:

```
//KAASIK17 JOB B-333,Y:KAASIK
// EXEC UPDATE
./ C XS52
./ N 1000,1010,10
(kaardid lisatavate direktiividega)
// EXEC FGC,
// PARM,PORT=MAP,
// NAME=XS52
//
```

Tellimuse neljanda reaga teatatakse, et tegemist tuleb nummerdamisega, kusjuures esimene arv (1000) määrab selle kaardi järjekorranumbri teegitud tekstis, mille järele peab lisatavad kaardid paigutama. Teise ja kolmanda arvuga antakse vastavalt esimese lisatava kaardi järjekorranumber ja nummerdamissamm. Teegitud teksti järgnevate direktiivide ümbernumerdamine toimub nüüd tarbe korral juba automaatselt.

## 2.2. Silumine

Kui perforeerimis- ja süntaksivead on kõrvaldatud, siis algab programmi sisuline silumine. See toimub programmi täitmise teel, kusjuures lahendatavat ülesannet lihtsustatakse kas algandmete spetsiaalse valiku või algoritmi teatavate osade vahelejätmise teel.

FORTRAN-programmi saab siluda tavaliste võtetega, kuid on ette nähtud ka süsteemsed silumisvahendid. Nende kasutamiseks tuleb FORTRAN-programmi juba koostamise ajal lülitada nn. silumiskirjeldus ja vajalikul hulgal silumisdirektiive, mille toimele saab silumise ajal teha ajutisi muutusi programmis ja väljastada mitmesugust täiendavat informatsiooni programmeerija poolt valitud faili, nn. silumisfaili. Silumisvahendite kasutamine on võimalik üksnes siis, kui transleerimine toimub translaatoriga G, sest translaatoris H vastavad vahendid puuduvad.

Silumiskirjeldus ja silumisdirektiivid lisatakse iga silumisele kuuluva mooduli lõppu vahetult enne lõpukirjeldust (muidugi võib FORTRAN-programmi koosseisu kuuluda ka niisuguseid mooduleid, mis enam silumist ei vaja ning kuhu midagi ei lisata). Kui silutud programmi soovitakse hiljem kasutada lahendamiseks, siis tuleb parandada kõik silumisel avastatud vead, eemaldada silumisse puutuvaid konstruktsioonihikuid sisaldavad perfokaardid (või teegitud programmi read) ning transleerida uuesti - nüüd juba translaatoriga H.

Esimeseks silumisse puutuvaks konstruktsioonihikuks peab alati olema silumiskirjeldus. See esitatakse kujul

DEBUG  $J_1, \dots, J_n$

kus  $J_1, \dots, J_n$  on soovitud jälgimisrežiimide nimed, mille konkreetseteks väärtusteks saavad olla reserveeritud sõnad TRACE (märgendite jälgimine), SUBTRACE (protseduuri jälgimine), INIT (muutujate jälgimine), SUBCHK (indeksite jälgimine), või viimased kaks koos objektide loeteludega. Jälgimisrežiimi kasutamisel väljastatakse jälgitavate objektide kohta teateid silumisfaili. Silumisfaili number n teatatakse samuti jälgimisrežiimide nimede hulgas kujul UNIT(n), tingimata peab see kõigis korraga silutavates moodulites ühesugune olema. Kui silumisfaili numbrit silumiskirjelduses ei teatata, siis võtab operatsioonisüsteem  $n = 6$ , s.t. sellise teate puudumisel väljastatakse silumisfaili saatmiseks määratud informatsioon vahetult trükiseadmel. Järgnevas ongi piiratud vaid selle kõige tavalisema juhuga: silumisfaili väljastamise asemel räägitakse käesolevas punktis kõikjal trükkimisest.

Märgendite jälgimise režiimi korral trükitakse teadetes välja kõik märgendid, millele vastavaid direktiive mooduli läbimisel täidetakse. Märgendite väljatrükkimine toimub muidugi vaid nendes moodulites, kus vastav režiim on silumiskirjeldusega määratud. Peale selle tuleb aga eriliste silumisdirektiividega (vt. lk. 59) veel täpsustada nn. jälgimispiirkonnad, mille ulatuses vaadeldav režiim üldse kehtib (kõik teised jälgimisrežiimid saab kehtestada üksnes kogu silutava mooduli ulatuses).

Protseduuri jälgimise režiim kehtestatakse niisugustes (alamprogramme või funktsioone kujutavates) moodulites, mille poole pöördumisi soovitakse programmi silumise käigus registreerida. Iga pöördumise korral selle režiimiga varustatud protseduuri poole trükitakse nüüd nimelt välja vastava alamprogrammi või funktsiooni nimi, naasmisel põhiprogrammi aga sõna RETURN.

Muutujate jälgimise režiim on määratud ülevaate saamiseks muutujate väärtuste muutumisest programmi töö käigus. Iga kord, kui jälgitavale muutujale omistatakse uus väärtus, trükitakse välja teade kujul  $m = v$ , kus  $m$  on muutuja nimi ja  $v$  - omistatav väärtus.

Kui muutujate jälgimise režiim kehtestatakse silumiskirjelduses ainult oma nimega (INIT), siis kuuluvad jälgimisele kõik moodulis kasutatavad muutujad (nii liht- kui ka indeksitega). Kui aga režiimi nimi esitada silumiskirjelduses kujul

INIT ( $m_1, \dots, m_n$ )

kus  $m_i$  on muutujad või massiivide identifikaatorid, siis jälgitakse vaid selles loetelus antud muutujaid (massiivi identifikaatori esinemine loetelus tähendab, et jälgimisele kuuluvad kõik selle massiivi muutujad).

Indeksite jälgimise režiimi korral kontrollitakse, kas muutujate indeksite väärtused jäävad massiivikirjeldusega määratud piiridesse. Tuleb nimelt arvestada, et FORTRAN-programmi transleerimisel eraldatakse indeksitega muutujatele

massiivikirjelduses määratud ulatuses mälu, kuid tööprogrammi tavalisel täitmisel indeksite konkreetsete väärtuste kuulumist kirjeldatud piiridesse ei kontrollita. Kui silumine toimub indeksite jälgimise režiimis, siis jälgitava massiivi muutujate esinemisel programmis selline kontroll toimub. Programmi täitmine jätkub küll igal juhul, kuid massiivikirjeldusest välja ulatava indeksitega muutuja m esinemisel trükitakse teade kujul SUBCHK m.

Kui vaadeldav režiim kehtestatakse silumiskirjelduses ainult oma nimega (SUBCHK), siis kuuluvad jälgimisele kõik moodulis kirjeldatud massiivid. Kui aga režiimi nimele lisada massiivide identifikaatorite loetelu

SUBCHK ( $m_1, \dots, m_n$ ) ,

siis jälgitakse vaid nendesse massiividesse kuuluvate muutujate indekseid.

Silumiskirjeldust sisaldavasse moodulisse võib lisada suvalisel arvul nn. silumisdirektiive, mis tuleb kõik kirjutada mooduli lõppu, silumiskirjelduse ja lõpukirjelduse vahele. Silumisdirektiivide osa jaguneb rühmadeks, iga rühm sisaldab need direktiivid, mis kuuluvad täitmisele mooduli ühes punktis. Silumisdirektiivide iga rühm algab nn. punktikirjeldusega ja selle rühma lõpetab kas silumisdirektiivide järgmist rühma alustav punktikirjeldus või siis silutava mooduli lõpukirjeldus.

Iga punktikirjeldus esitatakse kujul

AT m

kus  $m$  on silutava mooduli selle direktiivi märgend, mille täitmisele peab eelnema vastava rühma silumisdirektiivide täitmine.

Silumisdirektiivideks võivad olla kõik FORTRANi direktiivid, ainult tsükliidirektiivi kasutamisel peab arvestama, et tsükli mõjupiirkond ei tohi vastavast silumisdirektiivide rühmast välja ulatuda. Silumisdirektiive võib ka märgendada, kuid need märgendid ei tohi muidugi kokku langeda silutava mooduli omadega ning silutav moodul ise ei tohi nende järgi silumisdirektiividele juhtimist anda (küll aga võivad silumisdirektiivid anda juhtimist silutavasse moodulisse - ainult mitte mõne tsükli mõjupiirkonda). Silumisdirektiivide hulka ei tohi lülitada sisendi- või lihtfunktsioonikirjeldusi, kuigi lubatavad on näiteks formaadikirjeldused.

Peale FORTRANi direktiivide võib kasutada veel kolme spetsiaalset silumisdirektiivi (ainult mitte loogilise suunamisdirektiivi koosseisus). Nendeks on jälgimispiirkonna avamise ja sulgemise direktiivid

TRACE ON  
TRACE OFF

ning näitamisdirektiiv

DISPLAY  $m_1, \dots, m_n$

Jälgimispiirkonna avamise ja sulgemise direktiivid (neid tohib kasutada muidugi üksnes siis, kui märgendite jälgimise režiim on silumiskirjelduses teatatud) määravad moodulis punktid, kus märgendite jälgimine vastavalt algab või lõpeb. Niisuguseid piirkondi võib moodulis avada suvalise arvu, kuid neil piirkondadel ei saa loomulikult olla ühisosi.

Näitamisdirektiiv on määratud temas loetletud muutujate väärtuste väljatrukkimiseks selle direktiivi täitmise momendil. Loetelu elementideks tohivad olla üksnes lihtmuutujad ning massiivide nimed, keelatud on lülitada sinna indeksitega muutujaid või vastava mooduli kui protseduuri formaalseid parameetreid. Kui loetelu elemendiks on massiivi nimi, siis tähendab see, et trükitakse välja selle massiivi kõigi muutujate väärtused. Kogu trükkimine, näitamisdirektiivi abil toimub täpselt samas vormis nagu väljastamisel nimistu järgi (vt. lk. 34).

Silumiseks määratud vahendite kasutamise illustreerimiseks on järgmisel leheküljel toodud vastav formaalse sisuga ühemooduliline näide. Selle (vahelejätmistega esitatud) mooduli silumisel kehtib muutujate  $X, Q(1), \dots, Q(30)$  jälgimise režiim kogu ulatuses, märgendite jälgimise režiim aga ainult märgenditega 30 ja 26 varustatud direktiivide vahel (esimene kaasa, teine välja arvatud). Kui silumisel jõutakse direktiivini märgendiga 10, siis selle asemel täidetakse vastava silumisdirektiivide rühma esimene direktiiv ja minnakse kohe üle direktiivile märgendiga 15. Seega mooduli osa märgendite 10 ja 15 vahel jääb silumisel täitmata. Enne moodulist väljumist trükitakse muutujate  $F$  ja  $S$  väärtused.

```

SUBROUTINE SAP(A,C,Y,N)
  INTEGER Q(30)
  DIMENSION C(N),Y(N)

      . . .
5  X = A
30 Y(1) = Q(1) * F
      . . .
25 Q(2) = F
26 GO TO 3
      . . .
DO 9 I = 2,N
9  Y(I) = A + B * C(I)
10 Y(N) = Y(N) - A
      . . .
15 S = S + Y(N)
      . . .
35 RETURN

  DEBUG TRACE, INIT(X,Q)
  AT 30
  TRACE ON
  AT 26
  TRACE OFF
  AT 10
  Y(N) = B * C(N)
  GO TO 15
  AT 35
  DISPLAY P,S
  END

```

### 2.3. Lühendamine

FORTRAN-programmi moodulite transleerimisel saadavad tööprogrammid ei ole ka vigade puudumisel veel vahetult täidetavad - nad tuleb kõigepealt komplekteerida. Komplekteerimine kujutab endast transleerimistulemuste ja vajalike süsteemsete protseduuride ühendamist täitmiseks valmis olevaks tervikprogrammiks, nn. paigaldusmooduliks.

Kui FORTRAN-programmi soovitakse otsekohe pärast transleerimist ka täita (näiteks silumiseks), siis tuleb kasutada transleerimiseks, komplekteerimiseks ja täitmiseks määratud juhtprotseduuri FGCLG. Näiteks varem nime XS52 all teegitud FORTRAN-programmi korral võib sellise töö tarvis moodustada järgmise tüüpilise tellimuse:

```
//KAASIK18 JOB B-333,Y.KAASIK
// EXEC FGCLG,
// NAME=XS52
//GO.SYSIN DD *
(kaardid FORTRAN-programmi algandmetega)
//
```

Kaardid algandmetega tuleb lisada muidugi vaid siis, kui programm kasutab kaartsisendit. Algandmete puudumisel jääb ära ka kolmas juhtlause, mis teatab, et algandmed paiknevad kaartsisendil.

TRU arvutuskeskuses võib protseduuri FGCLG asemel soovi korral kasutada ka samu operatsioone sooritavat juhtprotseduuri FGCLD, mis vajab küll veidi rohkem sisemälu, kuid töötab tunduvalt kiiremini.

Korduvalt kasutatavaid programme pole otstarbekohane enne iga täitmist uuesti transleerida. Sellised programmid saadetakse pärast silumise lõpetamist transleeritult ja komplekteeritult teeki hoiule. TRÜ arvutuskeskuses kasutatakse selleks otstarbeks peamiselt kaht standardset teeki - SYS1.MODUL.LINK ja SYS1.MODUL.LOAD . Esimene neist on määratud üksikprotseduuride ja silumisel olevate paigaldusmoodulite, teine aga valmisprogrammide jaoks. Kui FORTRAN-programmi tekst on viidud teeki SYS1.TEXTLIB, siis transleerimise (ntüüd juba translaatoriga H), komplekteerimise ja tulemuse salvestamise teeki SYS1.MODUL.LOAD võib realiseerida näiteks järgmise tellimusega:

```
//KAASIK19 JOB B-333,Y.KAASIK
// EXEC FHCL,MLIB=LOAD,NCAL=,
// NAME=XS52
//
```

Selle tellimuse toimel saab paigaldusmoodul sama nime XS52, mida kannab lähtetekst. Kui paigaldusmoodulile soovitakse anda mingi teine nimi (näiteks XS72), siis tuleb tellimust täiendada ühe juhtlausega ja esitada ta kujul:

```
//KAASIK19 JOB B-333,Y.KAASIK
// EXEC FHCL,NCAL=,
// NAME=XS52
//LKED.SYSLMOD DD DSN=SYS1.MODUL.LOAD(XS72)
//
```

Kui FORTRAN-programmi tekst pole teegitud, vaid paikneb perfokaartidel, siis viimase tellimuse võib esitada kujul:

```
//KAASIK19 JOB B-333,Y.KAASIK
// EXEC FHCL,MLIB=LOAD,NCAL=,
// NAME=XS72
(kaardid.FORTRAN-programmi tekstiga)
//
```

Olgu märgitud, et juhtprotseduuri FHCL (ning samuti ka juhtprotseduuri FGCL) võib ühes töös korduvalt kasutada.

Viimases kolmes tellimuses on kõigis nõutud täielikku komplekteerimist - kui mõningaid vajalikke mooduleid (mille poole teised moodulid pöörduvad) ei leita, siis jääb komplekteerimine osaliseks ja teatatakse kõigi puudevate moodulite nimed. Kui soovitaksegi saada n.ö. tühikomplekteerimist, mille korral kõiki vajalikke mooduleid ei otsita, siis tuleb kirjutis NCAL= vastavas juhtlauskes ära jätta. Sellisel juhul saadakse paigaldusmoodul, mis ei kõlba täitmiseks, kuid mida saab vahetult ära kasutada hilisematel komplekteerimistel.

Programmis nõutavaid mooduleid otsitakse komplekteerimisel algul samas töös transleeritud moodulite hulgast, siis standardfunktsioonide teegist SYS1.FORTLIB, paigaldusmoodulite teegist SYS1.MODUL.LINK ja arhiivist SYS1.MODUL.SUB. Sealjuures võib mõne mooduli mitteleidmise põhjuseks olla ka asjaolu, et nad on n.ö. võõraste nimede all teegitud (näiteks kui põhiprogramm pöörduv protseduuri poole mitte selle enese nime, vaid tema sisendi nime kaudu). Seetõttu on täielikul komplekteerimisel otstarbekohane teatada nende paigaldusmoodulite nimed, kust tuleb vajalikke nimesid otsida. Näiteks kui niisuguste paigaldusmoodulite nimed on XS70 ja XS71, siis võib vastava tellimuse esitada kujul:

```
//KAASIK20 JOB B-333,Y.KAASIK
// EXEC FHCL,MLIB=LOAD,NCAL=,
// NAME=XS52
//LKED.SYSIN DD *
  I XS70,XS71
//
```

Teegitud paigaldusmooduli täitmiseks võib kasutada juhtprotseduuri FG (kus G viitab mitte translaatorile, vaid täitmist realiseerivale alamprotseduurile GO). Kui programm ei kasuta muid välisseadmeid peale kaartsisendi ja trüki-seadme, siis tellimuse teeki SYS1.MODUL.LOAD komplekteeritud paigaldusmooduli XS72 täitmiseks võib esitada näiteks kujul:

```
//KAASIK21 JOB B-333,Y.KAASIK
// EXEC FG,TIME=3,LPR=200,
// MNAME=XS72
(kaardid FORTRAN-programmi algandmetega)
//
```

Paneme tähele, et paigaldusmooduli nimi tuleb nüüd teatada parameetriga MNAME. Et teek SYS1.MODUL.LOAD võetakse vaikimisi, siis võiski kirjutise MLIB=LOAD siin ära jätta. Parameetritega TIME ja LPR teatatakse protsessoriaja (minuteis) ja trükitavate ridade arvu maksimaalväärtused. Nende puudumisel võtab protseduur vaikimisi TIME=1 ja LPR=600.

Programmide silumise käigus on kasulik transleerimist ja komplekteerimist teostada moodulite kaupa: iga mooduli jaoks rakendatakse eraldi juhtprotseduuri FGCL (tühikomplekteerimisega), lõpuks aga lastakse tervikprogramm komplekteerida ja täita juhtprotseduuriga FLG.

Kui paigaldusmooduli täitmisel esineb viga, siis programmi tõi üldiselt katkestatakse ja trükiseadmel väljastatakse vastav teade. Veateade lahendamisel algab teate numbriga, millele enamasti järgneb veel teate ingliskeelne tekst ja informatsioon programmi seisundi kohta vea tekkimise momendil. Olulisemate veateadete loetelu võib leida lk. 47 all nimetatud väljaande lisas 2.

Teate tekstile järgnevaks informatsiooniks on enamiku vigade puhul katkestamise momendil töötanud protseduuride nimede loetelu, alustades vea avastanud protseduurist ning lõpetades põhimooduliga. See loetelu väljastatakse tabelina TRACEBACK näiteks kujul:

```

IHC215I CONVERT - ILLEGAL DECIMAL CHARACTER W
TRACEBACK ROUTINE ISN REG. 14 REG. 15 REG. 0 REG. 1
FCVTH# 000522D0 00052390 00000005 0005204C
MAIN 0000E410 01051B58 FD000008 0005FFFF
ENTRY POINT = 01051B58

```

Selle tabeli esimeses veerus (ROUTINE) tuuakse protseduuri nimi ja edasi registrite seisund pöördumisel. Eraldi reas teatatakse lõpus veel põhimoodulisse sisenemise aadress. Toodud näites avastati viga (formaadiga sisestamisel esines väljakirjeldajale I, E, F, D või Q vastavate sümbolite jadas lubamatu sümbol W) süsteemes protseduuris FCVTH#.

Kui süsteem töötab nn. laiendatud veatõttluse režiimis, siis väljastatakse koos veateatega veel täiendavat informatsiooni. Olulisim sellest on programmi seisundi sõna (PSW = Program Status Word), mis trükitakse 16-kohalise kuueastikümnendärvana, kujul

Seisundi sõna kuueteistkümnendnumbritest pakub huvi eeskätt kaheksas (1), mis kujutab endast nn. katkestuse tunnust ja täpsustab esinenud vea põhjust. Võimalike põhjuste ja neile vastavate i väärtuste loetelu on toodud järgmises tabelis:

1	Katkestuse põhjus
4	Esines aadress väljastpoolt kaitsepiirkonda
5	Esines aadress, mis ületab sisemälu mahu
6	Esines rajastamata algusaadressiga muutuja
7	Esines võimatu kujuga arv.
9	Toimus täisarvu jagamine nulliga
C	Tehe reaalarvudega põhjustas ületäitumise
D	Tehe reaalarvudega põhjustas alatäitumise
F	Toimus reaalarvu jagamine nulliga

Peale nimetatud põhjuste võib programmi täitmine lõppeda veel mitmesuguste operatsioonisüsteemi või operäätori poolt teostatud katkestuste tõttu. Nii näiteks katkestus koodiga SYSTEM=222 tähendab, et programmi täitmine lõpetati operäätoripuldilt (enamasti juhul, kui programm on sattunud lõpmatult kestvasse tsükklisse). Katkestus koodiga 722 tähendab tavaliselt, et trükiseadmele väljastatav informatsioon ületab lubatava mahu. Tuleb nimelt arvestada, et tööprogrammi täitmise ajal salvestatakse trükiseadmele saadetak informatsioon ajutiselt ketasmällu - tegelik trükkimine toimub alles pärast töö lõpetamist. Nimetatud katkestuse korral saadakse trükiseadmel enamasti vaid osaline väljund.

Kui programmis kasutatakse peale kaartsisendi ja trüki-seadme veel teisi välisseadmeid, siis tuleb lahendamistelli-muses sellest eriliste andmelausetega tingimata teatada, määrates igale kasutatavale failinumbrile vastavusse sobiva välisseadme. Üksikasjaliku ülevaate andmelausete kirjutamise reeglite kohta võib saada näiteks sarjas "Programme kõigile" 1979. aastal ilmunud väljaandest "Operatsioonisüsteemist OS arvutile EC-1022". Põhjalikum tutvumine selle väljaandega on hädavajalik siiski üksnes neile, kelle programmides tuleb tegemist välisseadmete kasutamise keerulisemate variantide-ga. Suuremale osale programmeerijatest piisab enamasti vaid lihtsamate tüüpsituatsioonide tundmisest. Järgnevas piirdu-megi mõne FORTRAN-programmide korral kõige tüüpilisema situ-atsiooni kirjeldamisega.

Iga välisseadet määrava andmelause peab tingimata varus-tama nimega kujul FTnF001, kus n on kahekohalisena kirjuta-tud failinumber (vajaduse korral lisatakse selle nime ette veel sammu nimi - vt. lk. 70).

Andmelause rohkearvulistest parameetritest tuleb alati esitada vastava andmekogumi nimi välisseadmel (parameeter DSNAME ehk DSN) ja selle andmekogumi seisundite kirjeldus (parameeter DISP), määrates nii lähteseisundi (reeglina NEW, OLD või MOD) kui ka lõppseisundi (kas DELETE, PASS, KEEP või CATLG). Sageli, eeskätt loodavate andmekogumite korral tuleb aga lisaks näidata veel seadme rühmanimi (parameeter UNIT), kõite nimi (VOL), nõutava mälu mah. (SPACE), kirje maksi-maalpikkus (BLKSIZE) ning magnetlindi korral ka märgendamis-viis (LABEL).

Olgu näiteks ülalvaadeldud paigaldusmoodulis XS72 tege-  
mist arvutustulemuste salvestamisega jadafaili 8, mida soo-  
vitakse hilisemate tõe de tarvis säilitada magnetlindil. Mii-  
suguse säilitamise jaoks peab kasutajale olema kinnistatud  
lindi isiklik eksemplar. Kui sellise kinnistatud lindi nimi  
on KAASIK ja vaadeldav fail soovitakse paigutada sinna kol-  
manda andmekogumina nimega YK3, siis võib vastava tellimuse  
esitada kujul:

```
//KAASIK22 JOB B-333,Y.KAASIK  
// EXEC PG,MLIB=LINK,  
// MNAME=XS72  
(kaardid FORTRAN-programmi algandmetega)  
//FT08F001 DD UNIT=TAPE,DSN=YK3,DISP=(NEW,PASS),  
// VOL=(PRIVATE,RETAIN,SER=KAASIK),LABEL=(3,SI)  
//
```

Selles tellimuses eeldatakse, et ühegi kirje pikkus ei ulatu  
üle standardset määratava 800-baidilise piiri. Kui program-  
mis kasutatavate formaadikirjelduste järgi on aga teada, et  
moodustatakse ka pikemaid (näiteks kuni 1000-baidilisi) kir-  
jeid, siis tuleb andmelausesse tingimata lisada veel üks pa-  
rameeter kujul DCB=BLKSIZE=1000 .

Kui sedasama faili soovitakse mõnes hilisemas tõe s ues-  
ti kasutada, siis tuleb esitada samasugune andmelause, ai-  
nult lähteseisundina peab nüüd näitama kas OLD või MOD. Läh-  
teseisund OLD tagurdab faili algseisu, nii et esimene sises-  
tamisdirektiiv pöördub faili esimese kirje poole. Lähtese-  
isundi MOD korral saab väljastamisdirektiiviga kohe lisada  
faili lõppu järjekordse uue kirje.

Kui samas paigaldusmoodulis XS72 soovitakse tulemusi vaid ajutiseks salvestamiseks jadafaili 8 saata, siis on soovitatav määrata failile vastavaks seadmeks ketasmälu. Nüüd tuleb aga tingimata näidata ka andmekogumi jaoks nõutav maht kettal. Näiteks kui on teada, et salvestamisele kuulub tõenäoliselt kuni 50 kirjet, igauks keskmiselt 300 baiti, siis võib vastava lahendamistellimuse esitada kujul:

```
//KAASIK23 JOB B-333,Y.KAASIK
// EXEC PG,MLIB=LINK,
// MNAME=XS72
(kaardid FORTRAN-programmi algandmetega)
//PT08P001 DD UNIT=SYSDA,DSNAME=&NIMI,
// DISP=(NEW,DELETE),SPACE=(300,(50,10))
//
```

Sümboliga & algav nimi viitab siin ajutisele andmekogumile, mälu mahu nõudmises on veel täiendavalt teatatud, et kirjete arvu suurenemisel üle viiekümne tuleb mahtu juurde anda kümne kirje kaupa.

Kui tellimus sisaldab mitu sammu, näiteks kui programm läheb täitmisele otsekohe pärast komplekteerimist, siis tuleb vältisseedet määrava andmelause nimele lisada täitmismammu nimi GO ja esitada tellimus kujul:

```
//KAASIK24 JOB B-333,Y.KAASIK
// EXEC PHCLG,
// NAME=XS52
//GO.SYSIN DD *
(kaardid FORTRAN-programmi algandmetega)
//GO.PT08P001 DD DSN=&NIMI,UNIT=SYSDA,
// SPACE=(300,(50,10)),DISP=NEW
//
```

Otsefaile kasutava programmi korral tuleb vastavad seadmed määrata rühmanimega SYSDA, teatades ühtlasi kõite nime, mille lahendajale annab arvutuskeskuse dispetšer. Tingimata peab nüüd jälgima, et nõutava mälu maht oleks kooskõlas programmis antud failikirjeldustega.

Olgu näiteks teegis SYS1.MODUL.LOAD olev paigaldusmoodul IS75 saadud niisuguse FORTRAN-programmi transleerimisel, mis sisaldab failikirjelduse

```
DEFINE FILE 9(50,120,L,K), 10(100,50,U,L)
```

Selle programmi täitmisel loodavat faili 9 kavatakse kasutada ka järgmises tšüs, faili 10 seevastu aga soovitakse nime OS1.XFAIL all kataloogida hilisemate tšüde tarvis. Kui dispetšerilt saadud kõitenimi on OS0001, siis vastava lahendamistellimuse võib esitada näiteks kujul:

```
//KAASIK25 JOB B-333,Y.KAASIK
// EXEC FG,
// MNAME=XS75
(kaadrid FORTRAN-programmi algandmetega)
//FT090001 DD DSN=ANDMKD,UNIT=SYSDA,
// VOL=SER=OS0001,SPACE=(120,(50)),DISP=(NEW,KEEP)
//FT100001 DD DSN=OS1.XFAIL,VOL=SER=OS0001,
// UNIT=SYSDA,SPACE=(100,(200)),DISP=(NEW,CATLG)
//
```

Kul andmekogum OS1.XFAIL oleks juba varem moodustatud ja kataloogitud, siis tuleks viimane andmelause selles tellimuses esitada kujul

```
//FT100001 DD DSN=OS1.XFAIL,DISP=OLD
```

Lisa.1. Translaatori OE veateated

Vea nr.	Tase	Vea iseloom
IFE000I	8	Viga, mille tarvis puudub tekst
IFE002I	12	Tsüklite mõjupiirkonnad lõikuvad
IFE003I	12	Avaldises vigane loogilise tehte märk
IFE004I	8	Lõpukirjeldus on esitatud mitmel real
IFE005I	12	Sulgude vigane kasutamine avaldises
IFE006I	12	Märgend sisaldab lubamatu sümboli
IFE007I	12	Avaldises kaks eraldajat järjest
IFE008I	12	Esineb liiga suur konstant
IFE009I	12	Arvulises konstandis lubamatu sümbol
IFE010I	12	Konstandi järk on vigane
IFE011I	12	Välisfunktsiooni kasutatakse muutujana
IFE012I	8	Kompleksarvulise konstandi komponendi- deks pole reaalarvud
IFE013I	12	Eraldajana kasutatud lubamatut sümbolit
IFE014I	12	Esineb lubamatut tüüpi konstant
IFE015I	12	Muutujat kasutatakse välisfunktsioonina
IFE016I	12	Suunamisdirektiivis lubamatu eraldaja
IFE017I	12	Arvutatavas või määratavas suunamisdi- rektiivis esineb lubamatu märgend
IFE018I	4	Direktiivile on pandud lubamatu märgend
IFE019I	12	Määratavas suunamisdirektiivis kas sulg puudub või esineb lubamatu eraldaja
IFE020I		
IFE021I	12	Arvutatavas suunamisdirektiivis esinev muutuja pole lubatud tüüpi
IFE022I	12	Määratavas suunamisdirektiivis esinev muutuja pole lubatud tüüpi

IFB023I	12	Failikirjelduses vigane failnumber
IFB024I	12	Failikirjelduses vigane eraldaja
IFB025I	12	Failikirjelduses vigane kirjete arv
IFB026I	12	Failikirjelduses vigane vahetustunnus
IFB027I	12	Määramisdirektiivis sobimatu muutuja
IFB028I	12	Määramisdirektiivis vigane eraldaja
IFB029I	8	Lause enam kui 20 real (arvestatakse)
IFB030I	12	Tsüklidirektiivis lubamatu märgend
IFB031I	12	Tsüklidirektiivis või tsüklilises elemendis vigane lähteväärtus
IFB032I	8	Esineb kirjeldus pärast direktiivi
IFB033I	8	Algvärtustamisprotseduuris direktiiv
IFB034I	12	Omistamisdirektiiv ei alga muutujaga
IFB035I	8	Lause enam kui 20 real (ignoreeritakse)
IFB036I	12	Süntaksiviga - lause pole liigitatav
IFB037I	12	Üldnimekirjelduses lubamatu sümbol
IFB038I	8	Asünkroonse informatsioonivahetuse direktiivis muutuva pikkusega massiiv
IFB039I	12	Failikirjelduses sobimatu seosemuutuja
IFB040I	12	Märgendikonstanti kasutatakse väljaspool protseduuridirektiivi
IFB041I	8	Loetelu järgi sisestamise või väljastamise direktiivis puudub tärn
IFB042I	12	Objektide loetelu sisaldab üle 30 sulu
IFB046I	8	Avaldises pole sulgude paarsus õige
IFB047I	12	Tekstikonstant pikem kui 255 sümbolit
IFB048I	8	Moodulis puudub lõpukirjeldus
IFB050I	12	Aritmeetilises suunamisdirektiivis on märgendid vigaselt esitatud

IFEO52I	12	Avaläises sulud või apostroofid valesti
IFEO53I	12	Võrdusmärk paikneb vale kohal
IFEO56I	12	Funktsioonil peab olema vähemalt üks formaalne parameeter
IFEO57I	12	Protseduuri nimi on valesti kirjutatud
IFEO58I	12	Formaalne parameeter valesti kirjutatud
IFEO59I	12	Funktsioonikirjelduses on määratud tüübiga mitte sobiv pikkus
IFEO60I	12	Indeksitega muutujas puudub indeks
IFEO62I	12	Ekvivalentsuskirjelduses on massiivi indeksite arv ebaõige
IFEO64I	12	Nimistukirjelduses vigane eraldaja
IFEO65I	12	Nimistule on pandud lubamatu nimi
IFEO66I	12	Nimistul mitteunikaalne nimi
IFEO67I	12	Nimistukirjelduses lubamatu element
IFEO69I	12	Ühispiirkonnakirjelduses vigane eraldaja
IFEO70I	12	Ekvivalentsuskirjelduses vigane eraldaja
IFEO71I	12	Ekvivalentsusrühmas pole kaht muutujat
IFEO72I	12	Ekvivalentsuskirjelduses vigane muutuja
IFEO73I	12	Ekvivalentsuskirjelduses pole indeks esitatud täisarvulise konstandi kujul
IFEO74I	12	Indeksitega muutujas üle 7 indeksi
IFEO75I	12	Kahes ühispiirkonnas ühesugune muutuja
IFEO76I	12	Prefiksiga tüübikirjelduse ees on lause
IFEO77I	12	Prefiksiga tüübikirjelduses eraldaja puudub või paikneb vale kohas
IFEO78I	12	Prefiksiga tüübikirjelduses vigane tüüp
IFEO79I	12	Prefiksiga tüübikirjelduses puudub täht
IFEO80I	12	Prefiksiga tüübikirjelduses vigane täht

IFE081I	12	Prefiksiga tüübikirjelduses on kasutatud lubamatut eraldajat
IFE082I	12	Prefiksiga tüübikirjeldus ei lõpe suluga
IFE083I	12	Prefiksiga tüübikirjelduse parameetrite osas eraldaja puudub või vigane
IFE084I	12	Prefiksiga tüübikirjelduses esineb sõne
IFE085I	12	Prefiksiga tüübikirjelduses pole arvestatud tähestiku järjekorda
IFE086I	12	Ühispiirkonnakirjelduses lubamatu nimi
IFE087I	12	Ühispiirkonnal lubamatu nimi
IFE088I	12	Formaalne parameeter ühispiirkonnas
IFE090I	12	Välisprotseduuril lubamatu nimi
IFE091I	12	Välisprotseduuridel lubamatu eraldaja
IFE092I	12	Muutuja tüüpi kirjeldatakse korduvalt
IFE093I	12	Tüübikirjelduses lubamatu eraldaja
IFE094I	12	Tüübikirjelduses lubamatu nimi
IFE095I	12	Tüübikirjelduses pikkus ei sobi tüübiga
IFE096I	12	Tüübikirjelduses puudub eraldaja
IFE101I	12	Tsüklidirektiivis või tsüklilises elemendis esineb lubamatu eraldaja
IFE102I	12	Kirje/failitagurdamise või faililõpetamise direktiivis lubamatu eraldaja
IFE104I	12	Kirje/failitagurdamise või faililõpetamise direktiivis vigane failinumber
IFE109I	12	Peatamisdirektiivis vigane eraldaja
IFE110I	12	Peatamisdirektiivis vigane tekst
IFE111I	12	Peatamisdirektiivis liiga pikk tekst
IFE115I	12	Naasmisdirektiivis pole antud täisarv
IFE116I	12	Tsüklidirektiivis või tsüklilises elemendis lubamatu tsükliparameeter

IFE117I	12	Algväärtustamisprotseduuris vale sümbol
IFE118I	12	Tsükli parameetrit muudetakse selle tsükli mõju piirkonnas
IFE119I	12	Tsükli parameetrit muudetakse tsükli liiges elemendis
IFE120I	12	Algväärtustamisprotseduur pole moodul
IFE121I	12	Algväärtustamiskirjelduses vigane muutuja nimi
IFE122I	12	Algväärtustamiskirjelduses esineb indeksitega muutuja, mis pole kirjeldatud
IFE123I	12	Algväärtustamiskirjelduses vale sümbol
IFE124I	12	Algväärtustamiskirjelduses vigane indeks
IFE125I	12	Muutuja indeksitel on lubamatu eraldaja
IFE129I	12	Algväärtus on lubamatu tüüpi
IFE132I	12	Algväärtuste vahel on lubamatu eraldaja
IFE133I	12	Tsükli direktiiv ei tohi paikneda loogilise suunamisdirektiivi koosseisus
IFE134I	12	Tsükli loendaja pole lubatud tüüpi
IFE135I	12	Tsükli lõppvärtus pole lubatud kujul
IFE136I	12	Üle 26 tsükli üksteise mõju piirkonnas
IFE137I	12	Tsükli samm pole lubatud kujul
IFE138I	12	Tsükli lõppdirektiiv paikneb tsükli direktiivist eespool
IFE139I	12	Loogiline suunamisdirektiiv loogilise suunamisdirektiivi koosseisus
IFE140I	12	Tingimuslikus suunamisdirektiivis esineb lubamatu sümbol
IFE143I	12	Lihtfunktsioonikirjelduses formaalne parameeter pole muutuja
IFE144I	12	Lihtfunktsioonil üle 20 argumenti

IFE145I	12	Lihtfunktsioonikirjelduses vale eraldaja
IFE146I	12	Lihtfunktsioonis võrdusmärk vale kohal
IFE147I	12	Lihtfunktsioonikirjeldus vale kohal
IFE148I	12	Indeksi maksimaalväärtus vigasel kujul
IFE149I	12	Muutuva pikkusega massiivi nimi peab olema formaalne parameeter
IFE150I	12	Massiivikirjelduses vigane eraldaja
IFE151I	12	Massiivile on pandud lubamatu nimi
IFE152I	12	Protseduurikirjelduses lubamatu eraldaja
IFE153I	12	Väärtustatud funktsiooni nimi on vigane või lubamatut tüüpi
IFE154I	12	Asünkroonse informatsioonivahetuse direktiivis on lubamatu parameeter
IFE155I	12	Asünkroonse informatsioonivahetuse direktiivis on lubamatu eraldaja
IFE156I	12	Informatsioonivahetusdirektiivis vigane parameetri nimi võrdusmärgi ees
IFE157I	12	Informatsioonivahetusdirektiivi objektide loetelus lubamatu nimi
IFE158I	12	Informatsioonivahetusdirektiivi tsükli- lises elemendis sulud või tüüp ei sobi
IFE160I	12	Informatsioonivahetusdirektiivis vigane parameetrite eraldaja
IFE161I	12	Informatsioonivahetusdirektiiv sisaldab liigse parameetri
IFE162I	12	Asünkroonse informatsioonivahetuse direktiivi objektide loetelu on vigane
IFE163I	12	Informatsioonivahetusdirektiivis esineb kirjeldamata massiiv
IFE164I	12	Informatsioonivahetusdirektiivi objek- tide loetelus esineb avaldis

IFE165I	12	Informatsioonivahetusdirektiivi parameeter pole ei massiiv ega nimistu nimi
IFE166I	12	Informatsioonivahetusdirektiivis failinumber pole täisarvuline
IFE167I	12	Funktsiooni nime on valesti kasutatud
IFE168I	12	Alamprogrammile on pandud lubamatu nimi
IFE169I	12	Nimistu järgi pöördumine otsefalli poole
IFE170I	12	Otsefalli korral vigane kirjenumber
IFE171I	12	Sisendi nimena kasutatud lubamatut nime
IFE172I	12	Asünkroonse informatsioonivahetuse direktiivis puudub direktiivinumber
IFE173I	12	Asünkroonse informatsioonivahetuse objektide loetelus vigased indeksid
IFE176I	12	Informatsioonivahetusdirektiivis vigane tsükliline element
IFE177I	12	Informatsioonivahetusdirektiivis on formaadikirjeldus valesti näidatud
IFE192I	12	Direktiivile ja formaadikirjeldusele on pandud ühesugune märgend
IFE193I	12	Märgendit kasutatakse teist korda
IFE194I	12	Tüübikirjelduse algväärtustamise osas puudub vajalik eraldaja
IFE197I	12	Lõpetamisdirektiivis pole antud täisarv
IFE199I	12	Protseduuri päis pole mooduli alguses
IFE200I	12	Apostroofides tekstikonstanti sellisel kohal esineda ei tohi
IFE201I	4	Sama tsükli loendajat kasutati juba välises tsükli
IFE202I	8	Massiivi kirjeldatakse teist korda (keh-tima jääb esimene kirjeldus)
IFE203I	8	Põhiprogrammis esineb sisendikirjeldus

IFE204I	8	Lõpetamisdirektiivis vigane eraldaja
IFE205I	8	Arvutatavas või määratavas suunamisdi- rektiivis lõpusula järel vigane element
IFE206I	8	Indeksitega muutujas pole massiivi nime järel indekseid
IFE207I	8	Tühidirektiivis esineb lubamatu tekst
IFE208I	4	Tühidirektiivil pole märgendit
IFE209I	8	Algväärtusena on antud kaheksand arv (asendatakse nulliga)
IFE211I	8	Kompleksarvulise konstandi komponendid on erineva pikkusega
IFE224I	8	Sellisel kohal paiknev direktiiv peaks olema märgendatud
IFE225I	8	Kirjeldus on tarbetult märgendatud (märgendit ignoreeritakse)
IFE226I	8	Identifikaatoris üle kuue sümboli
IFE229I	8	Lihtfunktsiooni kirjelduses ei kasutata kõiki tema argumente
IFE251I	8	Formaadikirjelduses välja laius null
IFE252I	8	Formaadikirjelduses vigane sümbol
IFE253I	8	Formaadikirjelduses sulgude arv ei sobi
IFE254I	8	Formaadikirjelduses puudub avav sulg
IFE255I	8	Formaadikirjelduses teksti pikkus null
IFE256I	8	Formaadikirjelduses esineb poolikult kirjeldatud väli
IFE257I	8	Formaadikirjelduses puudub väljakirjel- dajate vahel eraldaja
IFE258I	8	Formaadikirjelduses välja pikkus suurem kui 255 sümbolit
IFE259I	8	Formaadikirjelduses on kirjeldajaga T määratud suurem positsioon kui 65535

IFE260I	8	Formaadikirjelduses kolmekordne välja- kirjeldajate rühmade sisalduvus
IFE261I	8	Formaadikirjelduses puudub lõpusulg
IFE262I	8	Formaadikirjeldus on märgendamata
IFE263I	8	Formaadikirjelduses teisenduse nimi ei sobi välja laiuse esitusviisiga
IFE264I	8	Formaadikirjelduses puudub kümnendpunkt
IFE280I	4	Translaator ei saa näidata mälujaotust
IFE281I	4	Programmis on liiga palju muutujaid ja konstante (mälujaotust ei näidata)
IFE301I	16	Programmis on liiga palju arvutatavaid suunamisdirektiive
IFE302I	12	Ekvivalentsuskirjeldus laiendab ühis- piirkonda ettepoole
IFE303I	12	Ekvivalentsuskirjeldus sisaldab kirjel- damata massiivi
IFE304I	12	Ekvivalentsuskirjeldus samastab kaks ühispiirkonda
IFE305I	8	Ekvivalentsuskirjeldus sisaldab liiga suurte indeksitega muutuja
IFE306I	12	Ekvivalentsuskirjeldused on vastuolus
IFE307I	4	Algväärtustamiskirjeldus sisaldab prog- rammis kasutamata muutuja
IFE308I	12	Ekvivalentsuskirjeldus samastab ühe ühispiirkonna kaks muutujat
IFE309I	8	Ekvivalentsuskirjeldus sisaldab massii- vi nime ilma indeksiteta
IFE310I	12	Ekvivalentsuskirjelduses on indeksitega muutujal mittekonstantne indeks
IFE311I	12	Ekvivalentsuskirjelduses on indeksitega muutujal indeksite arv vigane

IFE312I	12	Ekvivalentsuskirjelduses esineb välisprotseduuri nimi
IFE314I	4	Ekvivalentsuskirjelduses on muutujate järjestus ebaratsionaalne
IFE315I	4	Ekvivalentsuskirjeldus põhjustab ebaratsionaalse rajastamise
IFE317I	12	Algväärtustamisprotseduuris pole ühispiirkonnakirjeldust
IFE318I	12	Algväärtustamiskirjeldusega algväärtustatakse ühispiirkonda
IFE319I	12	Algväärtustamisprotseduuriga algväärtustatakse mitte ühispiirkonda
IFE320I	12	Algväärtustatakse formaalset parameetrit
IFE321I	4	Algväärtustamiskirjelduses ei sobi algväärtuse tüüp (toimub teisendamine)
IFE322I	4	Ühispiirkonnakirjelduses on muutujate järjestus ebaratsionaalne
IFE323I	4	Ühispiirkonnakirjeldus põhjustab ebaratsionaalse rajastamise
IFE332I	12	Kasutatakse kirjeldamata märgendit
IFE334I	12	Ühispiirkonnakirjelduses kirjeldatakse muutuva pikkusega massiiv
IFE350I	12	Algväärtustamiskirjelduses puudub sulg
IFE351I	8	Antakse liiga pikk algväärtus
IFE352I	8	Algväärtustamiskirjelduses ei sobi algväärtuste arv
IFE353I	12	Indeks peab olema täisarvuline
IFE354I	12	Muutuva pikkusega massiiv pole formaalne parameeter või paikneb ühispiirkonnas
IFE355I	16	Aadresskonstante programmis liiga palju
IFE356I	12	Formaalne parameeter ühispiirkonnas

IFE357I	12	Muutuva pikkusega massiivi on valesti kirjeldatud.
IFE358I	12	Indeksi maksimaalväärtus pole formaalne parameeter või paikneb ühispiirkonnas
IFE362I	12	Informatsioonivahetusdirektiivi tsükli- lised elemendid valesti üksteise sees
IFE401I ... IFE491I	16	Mitmesugused translaatori või translaatori poole pöördumise vead; edasine transleerimine ei osutu võimalikuks
IFE500I	12	Vale pöördumine standardprogrammi poole
IFE501I	12	Kompleksarv esineb astendajas
IFE502I	12	Kompleksarv mittetäisarvulisel astmel
IFE503I	12	Üldnime järgi pöördumisel argumentide arv või tüüp ei sobi
IFE504I	12	Astendamisel esineb loogiline muutuja
IFE505I	12	Informatsioonivahetusdirektiivis antud märgend ei vasta formaadikirjeldusele
IFE506I	12	Lõpetava sulu ees puudub operand
IFE507I	12	Massiivi või funktsiooni nimi esineb standardfunktsiooni argumentina
IFE508I	12	Massiivi või funktsiooni nimi esineb pöördumisel üldnime järgi
IFE509I	8	Moodul ei lõpe kas naasmis- või lõpetamisdirektiiviga
IFE510I	12	Loogiline tehe teist tüüpi väärtustega
IFE511I	12	Asünkroonse informatsioonivahetuse direktiivinumber pole täisarv
IFE512I	12	Loogilises suunamisdirektiivis pole loogilist avaldist
IFE513I	12	Funktsioonil ja alamprogrammil sama nimi
IFE514I	12	Parameetrite tabeli ületäitumine

IFB515I	12	Võrdluses on kompleksarvuline operand
IFB516I	16	Aritmeetilises suunamisdirektiivis kompleksarvuline avaldis
IFB517I	12	Massiivi nime kasutatakse lubamatult ilma indeksiteta
IFB520I	12	Koma vale kohas
IFB521I	12	Ülearune lõpetav sulg
IFB522I	12	Puudub lõpetav sulg
IFB523I	12	Võrdusmärk vale kohas
IFB524I	12	Lõpetava sulu taga puudub tehtemärk
IFB525I	12	Võrdlusmärgi või loogilise tehte märgi vale kasutamine
IFB529I	12	Tegeliku parameetrina kasutatav nimi pole kuulutatud välisfunktsiooniks
IFB530I	12	Indeksitega muutujas vale arv indekseid
IFB531I	12	Väärtustatud lihtfunktsioonis pole argumentide arv õige
IFB541I	12	Standardfunktsiooni poole pöördumisel argumentide arv või tüüp vigane
IFB542I	12	Loogiline avaldis lubamatus kohas
IFB550I IFB552I	16	Tabelitele eraldatud mälust ei jätku; transleerimine katkestatakse
IFB570I	0	Programm on liiga suur, mistõttu optimeerimine jäetakse ära
IFB580I IFB600I	16	Translaator ei leia vajalikke nimesid või toimus lubamatu suunamine
IFB610I	4	Ei leidu vajalikku märgendit
IFB620I	8	Juhtimine anti tsükli mõjupiirkonda
IFB630I ... IFB662I	16	Translaatorile ei jätku tabelite koostamiseks eraldatud mälust või tabelites vastuolu; transleerimine katkestatakse

Lisa 2. Standardfunktsioonid

(funktsiooni väärtuse definitsioonis tähendab  $a$  ainult argumenti,  $a_1$  esimest argumenti jne.; tüüpide nimed on lühiduseks asendatud nende esitähedega, näiteks I\*2 tähendab INTEGER\*2)

Funktsiooni nimi	Funktsiooni väärtus	Funkts. identif.	Argum. arv	Argum. tüüp	Funkts. tüüp	Kitsendused
Teisendus täisarvuks	entier( $a+0.5$ )	IFIX	1	R*4	I*4	
		IIFIX	1	R*4	I*2	
Teisendus reaalarvuks	reaalarv $a$	FLOAT	1	I*4	R*4	
		DFLOAT	1	I*4	R*8	
		QFLOAT	1	I*4	R*16	
Täpsuse vähendamine	$a$	SNGL	1	R*8	R*4	
		SNGLQ	1	R*16	R*4	
		DBLEQ	1	R*16	R*8	
Täpsuse suurendamine	$a$	DBLE	1	R*4	R*8	
		QEXT	1	R*4	R*16	
		QEXTD	1	R*8	R*16	

Maksimum	$\max(a_1, \dots, a_n)$	MAXØ	$n \geq 2$	I * 4	I * 4	
		AMAXØ	$n \geq 2$	I * 4	R * 4	
		MAX1	$n \geq 2$	R * 4	I * 4	
		AMAX1	$n \geq 2$	R * 4	R * 4	
		DMAX1	$n \geq 2$	R * 8	R * 8	
		QMAX1	$n \geq 2$	R * 16	R * 16	
Miinimum	$\min(a_1, \dots, a_n)$	MINØ	$n \geq 2$	I * 4	I * 4	
		AMINØ	$n \geq 2$	I * 4	R * 4	
		MIN1	$n \geq 2$	R * 4	I * 4	
		AMIN1	$n \geq 2$	R * 4	R * 4	
		DMIN1	$n \geq 2$	R * 8	R * 8	
		QMIN1	$n \geq 2$	R * 16	R * 16	
Ruutjuur	$\sqrt{a}$	SQRT	1	R * 4	R * 4	a $\geq$ 0
		DSQRT	1	R * 8	R * 8	
		QSQRT	1	R * 16	R * 16	
		CSQRT	1	C * 8	C * 8	leitakse peaväärtus
		CDSQRT	1	C * 16	C * 16	
		CQSQRT	1	C * 32	C * 32	

Märgi omistamine	$ a_1  \cdot \text{sign}(a_2)$	ISIGN	2	I*4	I*4	$a_1 \neq 0$
		SIGN	2	R*4	R*4	
		DSIGN	2	R*8	R*8	
		QSIGN	2	R*16	R*16	
Positiivne vahe	$a_1 - \min(a_1, a_2)$	IDIM	2	I*4	I*4	
		DIM	2	R*4	R*4	
		DDIM	2	R*8	R*8	
		QDIM	2	R*16	R*16	
Kompleksarvu moodustamine	$a_1 + ia_2$	CMPLEX	2	R*4	C*8	
		DCMPLEX	2	R*8	C*16	
		QCMPLEX	2	R*16	C*32	
Reaalosa	$\text{Re}(a)$	REAL	1	C*8	R*4	
		DREAL	1	C*16	R*8	
		QREAL	1	C*32	R*16	
Imaginaarosa kordaja	$\text{Im}(a)$	AIMAG	1	C*8	R*4	
		DIMAG	1	C*16	R*8	
		QIMAG	1	C*32	R*16	

Kaaskompleks	$\bar{a}$	GONJG	1	C*8	C*8	
		DCONJG	1	C*16	C*16	
		QCONJG	1	C*32	C*32	
Järgi leidmine	$a_1 - a_2 \cdot \text{entier}\left(\frac{a_1}{a_2}\right)$	MOD	2	I*4	I*4	$a_2 \neq 0$
		AMOD	2	R*4	R*4	
		DMOD	2	R*8	R*8	
		QMOD	2	R*16	R*16	
Lõige (täisosa)	$\text{sign}(a) \cdot \text{entier}( a )$	INT	1	R*4	I*4	
		IDINT	1	R*8	I*4	
		IQINT	1	R*16	I*4	
		AINT	1	R*4	R*4	
		DINT	1	R*8	R*8	
		QINT	1	R*16	R*16	
Absoluutväärtus	$ a $	IABS	1	I*4	I*4	
		ABS	1	R*4	R*4	
		DABS	1	R*8	R*8	
		QABS	1	R*16	R*16	

Moodul	a	CABS	1	C*8	R*4	
		CDABS	1	C*16	R*8	
		CQABS	1	C*32	R*16	
Eksponent	$e^a$	EXP	1	R*4	R*4	$-180.218 \leq a \leq 174.673$
		DEXP	1	R*8	R*8	
		QEXP	1	R*16	R*16	
		CEXP	1	C*8	C*8	$\operatorname{Re}(a) \leq 174.673$ $ \operatorname{Im}(a)  < \pi \cdot 2^{18}$
		CDEXP	1	C*16	C*16	$\operatorname{Re}(a) \leq 174.673$ $ \operatorname{Im}(a)  < \pi \cdot 2^{50}$
CQEXP	1	C*32	C*32	$\operatorname{Re}(a) \leq 174.673$ $ \operatorname{Im}(a)  < 2^{100}$		
Naturaallogaritm	$\ln(a)$	ALOG	1	R*4	R*4	$a > 0$
		DLOG	1	R*8	R*8	
		QLOG	1	R*16	R*16	
		CLOG	1	C*8	C*8	$a \neq 0$ ; leitakse peaväärtus
		CDLOG	1	C*16	C*16	
CQLOG	1	C*32	C*32			

Kümnerdlogaritm	log(a)	ALOG1Ø	1	R * 4	R * 4	a > 0
		DLOG1Ø	1	R * 8	R * 8	
		QLOG1Ø	1	R * 16	R * 16	
Siinus	sin(a)	SIN	1	R * 4	R * 4	a  < π · 2 <sup>18</sup>
		DSIN	1	R * 8	R * 8	a  < π · 2 <sup>50</sup>
		QSIN	1	R * 16	R * 16	a  < 2 <sup>100</sup>
		CSIN	1	C * 8	C * 8	Re(a)  < π · 2 <sup>18</sup>  Im(a)  ≤ 174.673
		CDSIN	1	C * 16	C * 16	Re(a)  < π · 2 <sup>50</sup>  Im(a)  ≤ 174.673
		CQSIN	1	C * 32	C * 32	Re(a)  < 2 <sup>100</sup>  Im(a)  ≤ 174.673
Keosinus	cos(a)	COS	1	R * 4	R * 4	samad, mis siinuse korral
		DCOS	1	R * 8	R * 8	
		QCOS	1	R * 16	R * 16	
		CCOS	1	C * 8	C * 8	
		CDCOS	1	C * 16	C * 16	
		CQCOS	1	C * 32	C * 32	

Tangens	tan(a)	TAN	1	R * 4	R * 4	$ a  < \pi \cdot 2^{18}; a \neq (2k+1)\frac{\pi}{2}$
		DTAN	1	R * 8	R * 8	$ a  < \pi \cdot 2^{50}; a \neq (2k+1)\frac{\pi}{2}$
		QTAN	1	R * 16	R * 16	$ a  < 2^{100}; a \neq (2k+1)\frac{\pi}{2}$
Kootangens	cot(a)	COTAN	1	R * 4	R * 4	$ a  < \pi \cdot 2^{18}; a \neq k \cdot \pi$
		DCOTAN	1	R * 8	R * 8	$ a  < \pi \cdot 2^{50}; a \neq k \cdot \pi$
		QCOTAN	1	R * 16	R * 16	$ a  < 2^{100}; a \neq k \cdot \pi$
Arkussiinus	arcsin(a)	ARSIN	1	R * 4	R * 4	$ a  \leq 1$
		DARSIN	1	R * 8	R * 8	
		QARSIN	1	R * 16	R * 16	
Arkuskoosinus	arccos(a)	ARCOS	1	R * 4	R * 4	$ a  \leq 1$
		DARCOS	1	R * 8	R * 8	
		QARCOS	1	R * 16	R * 16	
Arkustangens	arctan(a)	ATAN	1	R * 4	R * 4	
		DATAN	1	R * 8	R * 8	
		QATAN	1	R * 16	R * 16	
	arctan $\frac{a_1}{a_2}$	ATAN2	2	R * 4	R * 4	$ a_1  +  a_2  \neq 0$
		DATAN2	2	R * 8	R * 8	
		QATAN2	2	R * 16	R * 16	

Hüperboolne siinus	$sh(a) = \frac{e^a - e^{-a}}{2}$	SINH	1	R*4	R*4	a  < 175.366
		DSINH	-1	R*8	R*8	
		QSINH	1	R*16	R*16	
Hüperboolne kõosinus	$ch(a) = \frac{e^a + e^{-a}}{2}$	COSH	1	R*4	R*4	a  < 175.366
		DCOSH	1	R*8	R*8	
		QCOSH	1	R*16	R*16	
Hüperboolne tangens	$th(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$	TANH	1	R*4	R*4	
		DTANH	1	R*8	R*8	
		QTANH	1	R*16	R*16	
Gammafunktsioon	$\Gamma(a) = \int_0^{\infty} x^{a-1} e^{-x} dx$	GAMMA	1	R*4	R*4	$2^{-252} < a < 57.5744$
		DGAMMA	1	R*8	R*8	
Logaritm gamma- funktsioonist	$\ln \Gamma(a)$	ALGAMA	1	R*4	R*4	$0 < a < 4.2913 \cdot 10^{73}$
		DLGAMA	1	R*8	R*8	
Laplace'i funktsioon	$\frac{2}{\sqrt{\pi}} \int_0^a e^{-x^2} dx$	ERF	1	R*4	R*4	
		DERF	1	R*8	R*8	
		QERF	1	R*16	R*16	
	$\frac{2}{\sqrt{\pi}} \int_a^{\infty} e^{-x^2} dx$	ERFC	1	R*4	R*4	
		DERFC	1	R*8	R*8	
		QERFC	1	R*16	R*16	

Lisa 3. Koodi JKOM põhisümbolid

Sümboli 16-ndkood	Sümboli trükiukuju	Sümboli kood perfokaardil	Sümboli 16-ndkood	Sümboli trükiukuju	Sümboli kood perfokaardil
4Ø		—	6F	?	0-8-7
4A	[	12-8-2	7A	:	8-2
4B	.	12-8-3	7B	#	8-3
4C	<	12-8-4	7C	@	8-4
4D	(	12-8-5	7D	,	8-5
4E	+	12-8-6	7E	=	8-6
4F	!	12-8-7	7F	"	8-7
5Ø	&	12	A1	-	11-0-1
5A	]	11-8-2	B8	Ю	12-11-0-8
5B	⊠	11-8-3	BA	Б	12-11-0-8-2
5C	*	11-8-4	BB	Ц	12-11-0-8-3
5D	)	11-8-5	BC	Д	12-11-0-8-4
5E	;	11-8-6	BE	Ф	12-11-0-8-6
5F	⌋	11-8-7	BF	Г	12-11-0-8-7
6Ø	-	11	C1	A	12-1
61	/	0-1	C2	B	12-2
6A		12-11	C3	C	12-3
6B	,	0-8-3	C4	D	12-4
6C	%	0-8-4	C5	E	12-5
6E	>	0-8-6	C6	F	12-6

Simboli kood	Simboli trikiraja	Simboli kood perfoardid	Simboli kood	Simboli trikiraja	Simboli kood perfoardid
C7	G	12-7	E7	X	0-7
C8	H	12-8	E8	Y	0-8
C9	I	12-9	E9	Z	0-9
CB	N	12-0-9-8-3	EB	Y	11-0-9-8-3
CC	N	12-0-9-8-4	EC	X	11-0-9-8-4
CD	R	12-0-9-8-5	ER	B	11-0-9-8-6
CE	II	12-0-9-8-6	EF	H	11-0-9-8-7
D1	J	11-1	EP	Ø	0
D2	K	11-2	EP	1	1
D3	L	11-3	EP	2	2
D4	M	11-4	EP	3	3
D5	N	11-5	EP	4	4
D6	O	11-6	EP	5	5
D7	P	11-7	EP	6	6
D8	Q	11-8	EP	7	7
D9	R	11-9	EP	8	8
DC	II	12-11-9-8-4	EP	9	9
E2	S	0-2	FA	3	12-11-0-9-8-2
E3	T	0-3	FB	III	12-11-0-9-8-3
E4	U	0-4	FC	3	12-11-0-9-8-4
E5	V	0-5	FD	III	12-11-0-9-8-5
E6	W	0-6	FE	Y	12-11-0-9-8-6

## INDEKS

- |   |   |
|---|---|
| <p>andmeelement - 33</p> <p>andmelause - 68</p> <p>asünkroonne sisestamis-<br/>    direktiiv - 41</p> <p>asünkroonne väljastamis-<br/>    direktiiv - 41</p> <p>asünkroonsuupirkond - 40</p> <p>direktiivinumber - 41</p> <p>ekstraväljundi kirjeldus - 8</p> <p>fail - 3</p> <p>failikirjeldus - 36</p> <p>faililõpetamise<br/>    direktiiv - 13</p> <p>failinumber - 5</p> <p>failitagurdamise<br/>    direktiiv - 12</p> <p>formaadikirjeldus - 14</p> <p>formaadita sises-<br/>    tamine - 10, 41</p> <p>formaadita väljas-<br/>    tamine - 11, 42</p> <p>FORTTRAN-blankett - 44</p> <p>indeksite jälgimine - 57</p> <p>jadafail - 4</p> <p>juhtlause - 46</p> | <p>juhtprotseduur - 47</p> <p>juhtsümbol - 4</p> <p>jälgimispiirkonna<br/>    avamine - 59</p> <p>jälgimispiirkonna<br/>    sulgemine - 59</p> <p>jälgimisrežiimi nimi - 56</p> <p>jätkamise tähis - 44</p> <p>katkestuse tunnus - 67</p> <p>kirje - 3</p> <p>kirje lõpp - 14</p> <p>kirje maksimaalpikkus - 4</p> <p>kirjenumber - 36</p> <p>kirjetagurdamise<br/>    direktiiv - 12</p> <p>kommentaar - 44</p> <p>komplekteerimine - 62</p> <p>kontrolltransleerimine - 48</p> <p>kood ДКОМ - 92</p> <p>kordumiste arv - 24</p> <p>lahendamistellimus - 46</p> <p>lõpetamistunnus - 42</p> <p>lõpuväljund - 8</p> <p>mastaabitegur - 19</p> <p>muutujate jälgimine - 57</p> |
|---|---|

mälujaotus - 52  
märgendite jälgimine - 56  
nimistu nimi - 32  
nimistukirjeldus - 32  
näitamisdirektiiv - 59  
objektide loetelu - 6  
ootamisdirektiiv - 42  
otsefail - 4  
otsimisdirektiiv - 39  
positsioon - 14  
prefiks - 48  
programmi seisundi sõna - 66  
protseduuri jälgimine - 57  
protsessoriaeg - 65  
punktikirjeldus - 58  
ridade arv - 65  
seosemuutuja - 37  
silumisdirektiiv - 59  
silumisfail - 55  
silumiskirjeldus - 55  
sisestamine loetelu  
järgi - 35  
sisestamine nimistu  
järgi - 33  
sisestamisdirektiiv - 6, 37  
sisestamispiirkond - 41  
standardfunktsioon - 84  
standardne määrang - 5

sõna - 37  
sünkroonne informatsi-  
oonivahetus - 40  
šiffer - 46  
teekimine - 48  
teisenduse nimi - 15  
transleerimine - 46  
transleerimisprotokoll - 50  
transleerimisvead - 72  
trükkimisdirektiiv - 12  
tsükliline element - 6  
tühikomplekteerimine - 64  
tühisisestus - 10  
vahetustunnus - 37  
vea tase - 50  
veateade lahendamisel - 66  
veateade transleerimisel - 50  
veaväljund - 8  
väli - 14  
välisseade - 3  
välja laius - 15  
väljakirjeldaja - 14  
väljastamine loetelu  
järgi - 35  
väljastamine nimistu  
järgi - 34  
väljastamisdirektiiv - 11, 39  
väljastamispiirkond - 42

## S i s u k o r d

<b>1. SISEND-VÄLJUNDOOPERATSIOONID</b>	
1.1. Informatsioonivahetuse üldpõhimõtted . . . . .	3
1.2. Informatsioonivahetus jadafallidega . . . . .	6
1.3. Formaadikirjeldus . . . . .	14
1.4. Formaadiga informatsioonivahetus . . . . .	26
1.5. Informatsioonivahetus nimistu või loetelu järgi . . . . .	32
1.6. Informatsioonivahetus otsefallidega . . . . .	36
1.7. Asünkroonne informatsioonivahetus . . . . .	40
<b>2. OPERATSIOONISÜSTEEMI KASUTAMISEST</b>	
2.1. Transleerimine . . . . .	44
2.2. Silumine . . . . .	55
2.3. Lahendamine . . . . .	62
Lisa 1. Translaatori OE veateated . . . . .	72
Lisa 2. Standardfunktsioonid . . . . .	84
Lisa 3. Koodi ДКОИ põhisümbolid . . . . .	92
Indeks . . . . .	94