



PROGRAMME KÕICILE

TARTU
1987

XI
A-2491
TARTU RIIKLIK ÜLIKOOL

Arvutuskeskus

ANDMEPANGA
JUHTIMISSÜSTEEM
STAMP

Programme kõigile

Koostanud P.Eelma
Ü.Kaasik

TARTU 1987

Kinnitatud matemaatikateaduskonna nõukogus

20. märtsil 1987.a.

Tartu Riikliku Ülikooli
Raamatukogu
N

Käesolev väljaanne kujutab endast meetodilist juhendit kõigile nendele, kellel on tarvis tutvuda andmepanga juhtimissüsteemi STAMP üldstruktuuri ja kasutamisevõimalustega. Sellist eesmärki arvestades kirjeldataksegi nii süsteemi kui ka tema koosseisu kuuluvaid keeli formaliseerimata esituses - ranged definitsioonid asendatakse enamasti vaid sõnaliste selgituste või hoopis näidetega. Nimetamisväärseid eelteadmisi arvutite ja/või programmeerimise kohta käesoleva väljaande lugejalt ei eeldata.

СИСТЕМА УПРАВЛЕНИЯ ШТАМП БАНКА ДАННЫХ.
Программы для всех.
Составители Пезтер Э з л м а, Ю л о К а а з и к.
На эстонском языке.
Тартуский государственный университет.
ЭССР, 202400, г.Тарту, ул.Ülikooli, 18.
Vastutav toimetaja J. Tapfer.
Paljundamisele antud 16.04.1987.
Formaat 60x84/16.
Rotaatoripaber.
Masinakiri. Rotaprint.
Tingtrükipoogmaid 5,58.
Arvestuspooonaid 4,06. Trükipoogmaid 6,0.
Trükiarv 200.
Tell. nr. 419.
Hind 15 kop.
TRÜ trükikoda. ENSV, 202400 Tartu, Tiigi t. 78.

1. SÜSTEEMI STRUKTUURIST

Andmepank on vahendite süsteem andmete ühtseks kogumiseks ja säilitamiseks ning valdkondade kaupa moodustatud nn. andmebaaside kollektiivseks kasutamiseks. TRÜ arvutuskeskuses ja rakendusmatemaatika laboratooriumis väljatöötatud andmepanga juhtimissüsteem STAMP kujutab endast tegelikult programmide komplekti, mis on orienteeritud põhiliselt just majandusalaste andmete kogumiseks ja töötlemiseks. Süsteemi loomisel arvestati paljude andmetöötlusülesannete praktilise lahendamise käigus selgunud vajadusi ning üldistati saadud kogemusi. Süsteem hõlmab nii vahendid andmebaaside loomiseks kui ka spetsiaalsed keeled andmete kirjeldamiseks, töötlusprogrammide koostamiseks ja tulemuste vormistamiseks trükitud kujul.

Süsteem STAMP on ette nähtud kasutamiseks EC-seeria arvutitel alates operatsioonisüsteemist OO 4.1 ja vajab tööks vähemalt 120 kilobaiti sisemälu. Suhtlemine arvutiga toimub peamiselt pakkrežiimis. Süsteemi aktiivseteks kasutajateks võivad sealjuures peale arvutuskeskuse töötajate olla ka nn. välistarbijad, kes saavad kasutada süsteemi kõiki võimalusi, s.t. koostada töötlusprogramme ning trükikirjeldusi, täita mõningaid andmebaasi administraatori funktsioone jne. Selle võimaldamiseks peeti süsteemi loomisel silmas, et põhiliseks

andmete edastamise vahendiks jääb välistarbijatele esialgu veel teletaibilt väljastatav perfolint - seetõttu on kõigis süsteemis kasutatavates keeltes piiratud ainult teletaibi sümbolitega.

Süsteemi kasutajatelt ei nõuta nimetamisväärsed teadmisi arvuti operatsioonisüsteemi kohta. Kõiki sagedamini vajaminevaid andmebaaside hooldamisega või andmete töötlemisega seotud tegevusi katavad nimelt süsteemi STAMP enese programmid, kus operatsioonisüsteemi poolt pakutud paljudest võimalustest on igal konkreetsel juhul valitud üks tüüpiline ja enamasti kõige lihtsam variant. Selline lähenemisviis piirab küll mõningatel juhtudel süsteemi võimalusi, kuid lihtsustab süsteemi arvutiga ja vähendab oluliselt mitmesuguste ekraanide tegemise tõenäosust.

Informatsiooni vahetamisel andmebaasiga on põhiühikuks kirje. Kõige lihtsamaks kirjetüübiks võib lugeda fikseeritud arvu elementaarandmete ehk elementide järjendit. Sageli on üksuguste lihtkirjete korral aga puuduseks asjaolu, et mitmete elementide ühesuguseid väärtusi tuleb korrata paljudes kirjetes. Süsteemis STAMP koondatakse sellised lihtkirjed ühtsate elementide ühiste väärtuste järgi üheks nn. STAMP-kirjeks. Neid ühiseid elemente nimetatakse võtmelementideks ning nende väärtuste kombinatsioon moodustab kirje võtme. Lisaks võtmelementidele võib kirjes olla muidugi veel teisi selliseid elemente, mille väärtus muutub ainult koos kirje võtmega. Kirje kõigi üksuguste ühekordselt esitatavate elementide järjendit nimetame selle kirje esimese taseme eksemplariks.

Kirje piires korduvate, s.t. erinevaid väärtuste komplekte omandavate elementide rühma nimetame kirje teise taseme eksemplariks. Ka teise taseme eksemplare saab veel analoogiliselt rühmitada, kui eraldada neist ühine osa ja moodustada korduvatest osadest kolmanda taseme eksemplariid. Sealjuures peavad need kolmanda taseme eksemplariid üle kogu kirje olema ühesuguse koosseisuga. Eksemplariide arv teisel ja kolmandal tasemel ei ole piiratud, küll aga ei tohi ühelgi konkreetsel juhul kogu kirje maht ületada 32 kilobaiti (see kitsendus on tingitud andmete füüsilise esituse realiseerimisest süsteemis).

Niisuguste lihtsate kolmetasemelise struktuuriga kirjetena saab esitada enamiku andmetöötluses vajalikest andmetest. Vähemalt kõik põhilised kasutajat huvitavad andmed (lähtedokumendid, väljundtabelid) saab enamasti esitada STAMP-kirjetena. Vaid üksikutel keerulisematel juhtudel tuleb sisuliselt kokkukuuluvad andmed esitada mitme kirjena, sidudes need omavahel ühise võtmega.

Iga kirje tuleb varustada nimega. Sealjuures kõik sama nimega kirjed peavad olema ühesuguse struktuuriga, mis määratakse erilise kirjeldusega - legendiga. Legendis antakse kirje iga elemendi iseloomustus: taseme number, andmetüüp, sümbolite maksimaalarv ja asukoht eksemplaris. Vajaduse korral teatatakse veel ka talendavad omadused, näiteks et tegemist on võtmeelemendiga või elemendil on muutuv pikkus jne. Legend (mille nimi ühtib kirjeldatava kirje nimega) peab olema koostatud ja translereitud enne vastavate kirjete tegelikku sisestamist andmebaasi.

Legend sisaldab kogu süsteemile vajaliku informatsiooni kirjete töötlemiseks. Muuhulgas määrab legend üheselt ära kirjete sisestamiseeskirja, samuti on ta aluseks kirjete nn. kontrolltrüki süsteemsele programmile ja kõigile süsteemi STAMP translaatoritele. Töötlusprogrammides tuleb andmete kirjeldamiseks ära näidata ainult kasutatavad legendid. Et programmi tekst ei sisalda kasutatavate andmete detailset kirjeldust, siis tuleb lähteandmete struktuuri mõninga muutmise korral teha programmis vaid minimaalseid parandusi, paljudel juhtudel saab isegi piirduda ainult programmi uuesti transleerimisega.

Andmebaasi lülitatavad kirjed peavad olema ühendatud andmefailidesse. Ühte faili võib koondada kas ainult kõik ühenimelised kirjed või ka mitme erineva nimega kirjed. Süsteemi STAMP failid on operatsioonisüsteemi seisukohalt tavalised jadafailid, kuhu uus informatsioon lisatakse lõppu või siis uuendatakse faili sisu teatud kohast alates kuni faili lõpuni. Kõigi failide jaoks kasutab süsteem STAMP ühist plokki pikkust 1600 baiti. STAMP-kirjed on üldjuhul muutuva pikkusega ja paiknevad failis vahetult üksteise järel. Sealjuures võib kirje jätkuda ühest plokist teise. Kirje väljaeraldamine ühest või mitmest järjestikusest plokist toimub juba süsteemi enda vahenditega.

Kirjete paiknemise järgi võib failid jagada kahte rühma: järjestatud failid (nimetame neid edaspidi põhifailideks) ja nn. kollektori tüüpi failid. Põhifailis on kirjed paigutatud võtmeväärtuse kasvavas järjekorras ning vajaliku kirje otsimine toimub kogu faili järjest läbivaatamise teel. Selline

kirjete paigutus osutub sobivaks nendel juhtudel, kui programmi täitmisel on vaja läbi vaadata kas kõik või siis küllalt suur osa failis asuvatest kirjetest.

Kui põhifailist on enamasti vaja leida vaid mõningaid üksikuid kirjeid, siis on otstarbekohane lasta sellele failile moodustada eriline indeks, mille põhjal saab määrata iga otsitava kirje asukoha failis ühe (või suure faili korral mõne) ploki täpsusega.

Uute kirjete lisamiseks või varem failis olevate kirjete parandamiseks tuleb põhifail uuendada esimesest muudetavast plokist alates. Et faili selline uuendamine osutub üsna töömahukaks, siis on loomulik niisugust tööd sooritada võimalikult harva. Seepärast süsteemis STAMP kogutakse vajalikud muudatused kõigi põhifailide jaoks esialgu erilisse nn. sisendfaili. Sisendfail on ühe lahendusseansi vältel kasutatav ajutine tööfail, kuhu salvestatakse kõik uued kirjed ja samuti ka kirjetena vormistatud korraldused mingite kirjete parandamiseks. Iga põhifaili kirjete jaoks formeeritakse sisendfailis omaette kataloog, kus on näidatud kirjete võtmed ja nende asukohad sisendfailis. Enne lahendusseansi lõppu tuleb süsteemile üldiselt anda korraldus kogu sisendfaili kogutud informatsiooni salvestamiseks põhifailidesse, sest seansi lõpul sisendfail kustutatakse.

Lisaks sisendfailile loob süsteem uute või muudetud kirjete kogumiseks veel kõigi andmefailide jaoks ühise püsifaili - kollektori. Kollektori struktuur on sarnane sisendfaili omaga, s.t. iga põhifaili kohta luuakse seal kataloog, kus näidatakse kirjete asukoht kollektoris. Kollektori viimases

plokis moodustatakse veel sisukord viitadega kataloogidele. Kui mingisse põhifaili on vähe parandusi, siis võib vastavad sisendfailis olevad kirjed kanda põhifaili asemel hoopis kollektorisse. Uued kirjed lisatakse alati kollektori lõppu, kusjuures kirjete järele kirjutatakse ka muutunud kataloogid ja sisukord. Kollektori täitumisel tuleb kirjed kanda sealt põhifailidesse. Nii sisendfailis, kollektoris kui ka põhifailis asuvad kirjed on lahendamise ajal ühtviisi kättesaadavad.

Ühe kasutaja või kasutajate grupi failid moodustavad nende isikliku andmebaasi ehk fondi. Fond tuleb kirjeldada spetsiaalse kirjega (mille süsteemi nimi on TNT), kus loetletakse nii kõik vastavat fondi moodustavad failid kui ka näidatakse iga faili jaoks sellesse faili kuuluvate kirjete nimed. Fondi andmetega suhtlemine toimub edaspidi enamasti just kirjenimede tasemel. Millisesse faili need kirjed kuuluvad, selle määrab süsteem juba kirje TNT vahendusel. Eri-nevate fondide vahelisi seoseid saab luua ühiste failide kaudu. Sellised failid tuleb nimelt ühtmoodi kirjeldada kõigis nendele fondidele vastavates kirjetes TNT.

Süsteemiga STAMP juhitud andmebaasides sisalduvate andmete töötlemise põhiliseks vahendiks on nn. STAMP-keeles kirjutatud programmid ehk STAMP-programmid. Üldjuhul kujutab STAMP-programm endast eeskirjade loetelu, mille alusel teatud hulgast lähtekirjetest moodustatakse tulemuskirjed. Tulemuskirjed võib sealjuures trükkida väljundtabelitena, aga samuti saab neid kasutada lähteandmetena kas samas või siis mingis järgmises programmis.

Programmis kasutatavaid eeskirju võib vaadelda teatud operatsioonidena, milles igaühes on antud nii operatsiooni nimi kui ka tulemusoperandide ja lähteoperandide ehk argumentide loetelu. Operatsiooni argumentideks on enamasti ühe või mitme kirje elemendid, kuigi argumenti kohal võib vajaduse korral esitada ka konstandi. Keele omapäraks on see, et programmis pole vaja näidata tsükleid üle kirje eksemplari-de: sõltuvalt operandide tasemest genereerib translaator vajalikud tsüklid ise.

Operatsioonid võib jagada kahte rühma: spetsiaalsed ja standardsed. Esimese rühma operatsioonid (kirjete lugemine, salvestamine, trükkimine, kirjetevaheliste seoste loomine jms.) on süsteemsed, olles tihedalt seotud kirje füüsilise esituse ja süsteemi paljude programmidega. Standardsed operatsioonid seevastu ei nõua süsteemi sisemise ehituse tundmist: tulemusoperandi(de) väärtused leitakse argumentide väärtuste järgi teatud hulga aritmeetiliste või loogiliste tehete abil. Uute standardsete operatsioonide koostamine on jõukohane igale keelt ASSEMBLER valdavale programmeerijale. Uute operatsioonide kasutuselevõtmiseks on vaja ainult lisada nende kirjeldused spetsiaalsesse kirjesse nimega OPTAB. Koostades sobiva komplekti operatsioone, võib süsteemiga STAMP lahendatavate ülesannete klassi märgatavalt laiendada.

Enamiku STAMP-programmis esinevatest vigadest avastab juba translaator, mis kontrollib operandide vastavust operatsiooni kirjeldusele, kasutatava elemendi eelnevat väärtustamist jne. Keerulisemate vigade avastamiseks saab aga kasutada süsteemi kuuluvaid spetsiaalseid silumisvahendeid.

Süsteemi STAMP lahenduseseanss täidetakse üldjuhul operatsioonisüsteemi ühe sammuna. Täidetavate tööde loetelu esitatakse süsteemile lihtsa tellimuskeele abil, kusjuures standardsete tööde korral saab ära kasutada ka varem ette valmistatud tellimusi.

Süsteem koosneb juhtprogrammist, tuumast ja tööprogrammidest. Juhtprogramm paigaldab süsteemi tuuma ning kutsub sobivas järjekorras välja tellimuses nimetatud tööprogramme. Tuumas koosseisu kuuluvad niisugused programmid, mis täidavad kõige sagedamini vaja minevaid funktsioone: suhtlemine välisseadmetega, mälu planeerimine, kirje genereerimine, teadete trükkimine jms. Süsteemseteks tööprogrammideks on andmete sisestaja, mitmesugused translaatorid, STAMP-programmi interpretaator, püsifailide korrigeerija ning paljud muud spetsiaalseid funktsioone täitvad programmid.

Süsteemi STAMP abil lahendatavate ülesannete maht võib olla väga erinev. Leidub kasutajaid, kel on tarvis täita ainult paari-kolme programmi, kusjuures fond koosneb üksnes ühest failist - kollektorist. Samal ajal on aga süsteemi vahenditega võimalik realiseerida keskmise ettevõtte AJS. Sellist suurt süsteemi on praktikas otstarbekohane luua teatavate osade kaupa: alustades üksikutest ülesannetest täiendatakse leodavat süsteemi uute ülesannete lisamise teel mingite enam-vähem terviklike allsüsteemideni ja ühendatakse lõpuks omavahel ka saadud allsüsteemid.

Et kõiki vajalikke variante pole niisuguse lähenemise viisi korral enamasti võimalik aegsasti ette näha, siis tuleb varem koostatud programmides ja/või andmekirjeldustes teha

töö käigus sageli muudatusi. Süsteemi STAMP vahendid ongi koostatud just nii, et soodustada sellist lähenemist. Esiteks kujutab iga kirje omaette üksust ja uute kirjete lisamine faili tavaliselt ei nõua varasemate programmide või andmete muutmist. Et kirjetevahelised seosed eksemplaride tasemel formeeritakse alles vastava programmi täitmise käigus, siis ei pea andmete planeerimise ajal ette nägema ka kõiki tulevikus vaja minevaid seoseid ja juurdepääse. Teiseks tuleb arvestada veel asjaolu, et tegelikult täidetava programmi koostab andmekirjelduste põhjal translaator. Seejärel saab uute elementide lisamise korral kirjesse enamikul juhtudel piirduda üksnes seda kirjet kasutavate programmide uuesti transleerimisega.

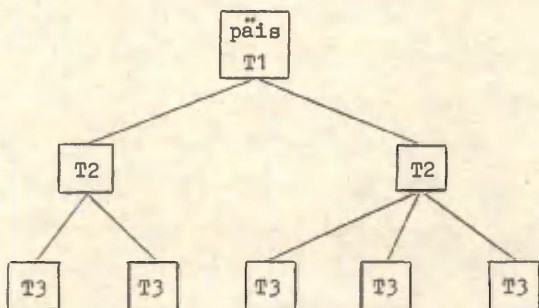
Suurte süsteemide koostamisel ei saa piirduda ainult ühe süsteemi vahenditega. Süsteemis STAMP on ette nähtud ka andmete vahetamise võimalus teiste süsteemidega andmefailide vahetamise kaudu. Nimelt võib STAMP-kirjetest kergesti formeerida fikseeritud pikkusega kirjetega faile, nagu neid kasutavad näiteks FORTRAN- või PL/1-programmid. Samuti saab teiste keelte vahenditega loodud fikseeritud pikkusega kirjetest moodustada tagasi STAMP-kirjed.

Süsteemi STAMP võib kasutada ka mittestandardsete andmete sisestamiseks (näiteks nende eelnevaks töötlemiseks ja mingile teisele süsteemile sobival kujul etteandmiseks). Teiselt poolt saab aga mingi muu süsteemi poolt genereeritud andmemassiive ka süsteemi STAMP vahenditega edasi töödelda, näiteks vormistada saadud tulemused sobivalt kujundatud väljundtabelitena.

2. KIRJE JA LEGEND

Kõige põhilisemaks mõisteks süsteemis STAMP on kirje. Kirjete kujul esitatakse kõik sisend- ja väljundandmed, tellimus tööde täitmiseks, lähtekeelsed tekstid, translaatorite vahetabelid ning ka transleerimise tulemused (STAMP-programmid, trükikirjeldused, andmekirjeldused).

Oma loogilise struktuuri poolest on kirje puu, milles tuleb piirduda ülimalt kolme tasemega, nagu skemaatiliselt kujutab kirje struktuuri järgmine joonis:



Puu juure moodustab kirje päis ja selle juurde kuuluv n.ö. esimese taseme informatsioon T1 (näiteks kui kirjes salvestatakse andmeid mingi kooli kohta, siis T1 sisaldab kooli üldandmed). Esimese taseme alluvuses on üldiselt juba terve hulk ühesuguse koosseisuga informatsioonikogumeid - need on puu teise taseme eksemplarid T2 (iga eksemplar sisaldab näiteks üldandmed vaadeldava kooli mingi klassi kohta). Lõpuks võib veel teise taseme iga eksemplari alluvuses omakorda olla mistahes arv kolmanda taseme eksemplare T3 (näiteks andmed vastava klassi kõigi õpilaste kohta). Need

kolmanda taseme eksemplarid peavad kogu kirje ulatuses jälle olema ühesuguse koosseisuga, s.t. sisaldama kindla sisulise tähendusega andmeelemente etteantud järjekorras.

Et esimese taseme ainsa ja teise taseme iga eksemplari alluvusse kuulub üldiselt terve rühm järgmise taseme ühesuguse koosseisuga eksemplare, siis öeldakse, et tegemist on eksemplaride korduvate rühmadega.

Mistahes taseme eksemplar jaguneb andmeannusteks ehk elementideks, mis võivad olla kas kõigis sama taseme eksemplarides ühesuguse, fikseeritud pikkusega, või siis igas eksemplaris üldiselt isesuguse, muutuva pikkusega. Arvuti mälus hõivab iga eksemplar tervikliku mäluvälja. Selle struktuur on üldjuhul

VN	VE	VV	L	V	F	VM	...	VM
----	----	----	---	---	---	----	-----	----

kus alamväljade tähendus on järgmine:

VN - viit sama taseme naabereksemplarile;

VE ja VV - viitavad alluva taseme esimesele ja viimasele eksemplarile (viimasel tasemel puuduvad);

L - lisaväljad viitamaks sama või mingi teise kirje teatud eksemplaridele, tingimuste täidetuse tähistamiseks jne.;

V - eksemplari nn. võtmeelemendid, mille väärtuste järgi eksemplare eristatakse;

F - muud fikseeritud pikkusega elemendid;

VM - viit muutuva pikkusega elemendile.

Väljad VN,...,L moodustavad eksemplari päise. Kõik viidatud on 2-baidised ja tähendavad kaugust kirje algusest. See-ga on kirje maksimaalseks lubatavaks pikkuseks 32K.

Sama taseme eksemplarid on korrastatud ahelasse võtmelementide järjendi (ehk võtme) väärtuste kasvamise järgi. Kui teisel või kolmandal tasemel võtmeelementid puuduvad, siis genereeritakse võtmeks eksemplari järjekorranumber. Esimese taseme võtmeelementidest moodustub kirje enda võti. Võtme puudumine esimesel tasemel näitab, et tegemist on ühekordse kirjega, s.t. andmebaasis saab olla vaid üks sellise struktuuriga kirje.

Kirje päis hõlmab 24 baiti. Päises on antud kirje nimi ja andmed iga taseme kohta: eksemplari pikkus EP, eksemplari päise pikkus PP, võtme pikkus VP ning muutuva pikkusega elementidele viitavate väljade kogupikkus MP. Päise struktuur arvuti mälus on järgmine:

süsteemne info			
EP1	FP1	VP1	MP1
EP2	FP2	VP2	MP2
EP3	FP3	VP3	MP3
kirje nimi			

Süsteemisiseselt vaadeldakse kirjet tervikuna, mistõttu päises olevast informatsioonist piisab nii kirje genereerimiseks kui ka sealt vajaliku eksemplari leidmiseks. Kasutaja programmides tegeldakse aga enamasti mitte eksemplaridega, vaid konkreetsete elementidega mingites eksemplarides. Seepärast tuleb peale kirje üldstruktuuri ette ära määrata ka kõik elemendid eraldi. Sellist kirje struktuuri ja elementide omadusi määravat andmekirjeldust nimetatakse legendiks.

Legendide esitamise ehk andmekirjelduskeele jaoks me ei anna siin ranget sùntaksit. Selle asemel võtame vaatlusele ühe niisuguse näitelegendi, kus on kasutamist leidnud enamik keele võimalustest. Olgu nimelt vaja viia arvutisse andmed ühe kooli õpilaste kohta. Andmed esitatakse klassi hõlmavate kirjete kaupa, kus iga õpilase kohta tuuakse üks teise ja iga õppeaine kohta üks kolmanda taseme eksemplar. Legendi lähtetekst sisestatakse kirjena LEG, mille iga rida iseloomustab kirje üht elementi.

LEG KLASS

1 NR X3-K KLASSI NR.

KLJUH T-V KLASIJUHATAJA PEREK., EESNIMI

AARV N2 AINETE ARV

2 PNIMI T12-K QPIL. PEREK. NIMI

ENIMI T12-K

SKUUP X8 SYNNTIAEG

KH N1.2-P KESKMINE HINNE

3 AINE X2-K AINE NR. TUNNISTUSEL

HINNE N1-4 VEERANDIHINDED

Legendi nimi (antud juhul KLASS), määrab vastava kirje nime. Elemendi kirjelduses antakse järjekorras tase, elemendi nimi, tüüp, täis- ja murdosa, omadus ning vajaduse korral kommentaar.

Tase määratakse tasemenumbriga 1, 2 või 3. Kui tase on mõnes reas andmata, siis võetakse selleks eelmise elemendi tasemenumber (esimese elemendi korral 1). Antud näites on KLJUH ja AARV seega esimese, ENIMI, SKUUP ja KH aga teise taseme ning HINNE kolmanda taseme elemendid.

Elemendi nimi on tähega algav tähtede ja/või numbrite järjend pikkusega kuni 8 sümbolit. Ühe legendi piires nimi ei tohi korduda, sama tähendusega elemendid erinevates legendides on aga soovitatav tähistada sama nimega.

Tüüp määrab elemendi esitusviisi arvutis, kusjuures kasutatakse järgmisi tüübinimesid:

- N - märgita kahendtäisarv,
- I - märgiga kahendtäisarv,
- D - kümnendarv,
- R - ujupunktarvuna esitatav reaalarv,
- X - 16-ndarv,
- T - tekst.

Kümnend- ja reaalarvu vaadeldakse alati märgiga arvude-na, seevastu 16-ndarvu tõlgendatakse märgita arvuna, mis võib lisaks tavalistele numbritele sisaldada ka tähti A...F (seda on arvestatud legendis KLASS elemendi NR tüübi määramisel).

Arvu(de)ga vahetult tüübi nime järel määratakse elemendi kujutamise pilt. Näiteks arvulise elemendi korral näidatakse numbrikohtade arv enne ja pärast koma, kusjuures täis- ja murdosa kohtade arv eraldatakse punktiga. Tüüpide N, I ja D korral on murdosa fiktiivne ning seda kasutatakse vaid sisend- ja väljundpildis arvu loetaval või harjumuslikul kujul esitamiseks. Arvutis kujutatakse arv ja sooritatakse te-maga tehteid nagu täisarvuga (antud näites säilitatakse keskmist hinnet KH täissajandikes, kuid väljastatakse tavalisel kujul). Teksti ja 16-ndarvu jaoks antakse tüübinime järel ainult sümbolite maksimaalarv: kui konkreetse elemendi

sümbolite arv on sellest väiksem, siis täiendab süsteem 16-ndarvu eest nullidega ja teksti lõpust tühikutega.

Numbrikohtade või sümbolite arvu võib jätta ka andmata. Sel juhul määratakse elemendi pilt vastava tüübinimega vaikimisi - süsteem lisab kohtade arvu järgmiselt: N7.2, I7.2, D5.2, R5.2, X8 ja T8.

Omadus täiendab elemendikirjeldust või näitab elemendi spetsiaalset kasutamisi. Omaduseks võib miinusmärgi järele kirjutada ühe või mitu järgmistest võimalustest:

- K - võtmeelement (sõnast KEY);
- V - muutuva pikkusega element (sõnast VARIABLE);
- <korduste arv> - korduv element;
- P - pseudo- ehk fiktiivne element;
- L - lisaelement.

Igal tasemel võib määrata ühe või mitu võtmeelementi, mille konkreetsete väärtuste kombinatsioon peab määrama eksemplari antud taseme piires üheselt. Kui teisel või kolmandal tasemel pole võimalik selliseid elemente määrata, siis jäetakse vastaval tasemel võtmelemendid tähistamata ja võtmeks loetakse eksemplari järjekorranumber.

Antud näites on kirje võtmeks klassi number, õpilase võtmeks tema ees- ja perekonnanimi (järjest kujutatult) ning hinnete võtmeks aine number.

Kuigi kirje füüsilises esituses paiknevad võtmelemendid eksemplari alguses, ei pea nad legendis olema kirjeldatud tingimata oma taseme esimeste elementidena ega isegi mitte järjest. Võtmelemendile ei tohi omistada teisi omadusi, s.t. ta ei või olla muutuva pikkusega, fiktiivne ega korduv.

Kui elemendi pikkus kõigub laias piires või maksimaalset sümbolite arvu ei osata anda, siis näidatakse elemendi omaduseks muutuv pikkus. Element võib olla muutuva pikkusega vaid tüüpide T ja X korral. Sümbolite arvu pole sel juhul kohustust anda, kuid vähimisi määratakse sümbolite arvuks teksti korral 100 ja 16-ndarvu korral 255. Kui orienteeruv maksimaalne pikkus on siiski antud, siis seda arvestatakse kirje väljatrükis lahtri laiuse planeerimisel. Antud näites on muutuva pikkusega klassijuhataja nimi.

Korduste arv antakse naturaalarvuna, kusjuures igas konkreetses eksemplaris võib tegelik korduste ehk selle elemendi komponentide arv olla ka väiksem. Kui elemendi mingile komponendile ei ole omistatud väärtust, siis vastav väli on sõltumata tüübist täidetud nullidega. Legendis KLASS on korduva elemendina antud veerandihinded: elemendi HINNE neli komponenti näitavad antud aine veerandihindeid klassitunnistusel (kokku on kolmanda taseme igas eksemplaris viis arvu).

Kui maksimaalne lubatud korduste arv on oluliselt suurem keskmisest korduste arvust, siis võib korduva elemendi vormistada muutuva kordumisega elemendina. Selleks tuleb elemendi kirjelduses omadusena näidata nii muutuv pikkus kui ka (pärast võrdusmärgi) korduste arv, näiteks võib mingisse leandi lülitada read

ARVUD N2.2-V=25

TEKSTID T-V=12

Rõhutame, et siin elemendis TEKSTID käib omadus V korduste arvu, aga mitte komponendi pikkuse kohta (milleks vähimisi võetakse 8).

Ülejäänud kaks omadust P ja L puudutavad ainult andmete sisestamist. Omadust "pseudo" kasutatakse selliste elementide jaoks, mida esialgu eksemplari koosseisus ei sisestata, vaid lisatakse vajaduse korral hiljem töötlusprogrammiga või erilise elementide salvestamise operatsiooniga.

Praktikas leidub vahel andmeelemente, mis omavad mõtet ainult mõningate eksemplaride korral. Et vältida ülejäänud eksemplarides nullide perforerimist, võib selliseid elemente vaadelda lisaelementidena. Lisaelemendid peavad tingimata olema kirjeldatud antud tasemel viimastena.

Sisestatud legendi teksti töötleb süsteemi STAMP koosseisu kuuluv andmekirjelduse translaator, mis annab tulemuseks kirje nimega LEGEND. Selle esimesel tasemel on vastava kirje näidispäis ja tasemete kaupa elementide ning võtmeelementide arv. Legendi igale elemendile vastab kirjes LEGEND üks järgmise struktuuriga 24-baldine teise taseme eksemplar:

VN			JNR		
elemendi nimi					
TYYP	L		SA		
TO	MO	V	T	TR	KA
lisaväli					

Joonisel näidatud väljad sisaldavad järgmisi elementi iseloomustavaid andmeid:

VN - viit järgmisele eksemplarile;

JNR - eksemplari järjekorranumber kirjes LEGEND;

TYYP - elemendi tüüp ja omadused kodeeritud kujul;

L - elemendile eraldatud baitide arv (muutuva pikkuse korral null, korduva elemendi jaoks komponendi pikkus);

SA - elemendi asukoht eksemplari alguse suhtes;

TO ja MO - täis- ja murdosa sümbolite arv;

V - võtmeelemendi tunnus;

T - tasemenumber;

TR - trükipositsioonide arv;

KA - korduste arv (ainult korduval elemendil).

Elemendile eraldatud baitide arv kirjes sõltub elemendi tüübist ning sümbolite maksimaalarvust. Järgnevas tabelis on iga tüübi ja sümbolite arvu S jaoks näidatud elemendile eraldatav baitide arv ning elemendi maksimaalne väärtus. Tüüpide N, I, R ja D korral on sümbolite arvuks S täis- ja murdosa numbrikohtade summa. Numbrikohtade arvu kontrollitakse ainult andmete sisestamisel (kui elemendi väärtus arvutatakse programmis, siis võib numbrikohti olla legendis kirjeldatust mõnevõrra rohkem).

Tüüp	S	baite	maks. väärtus
N	1...2	1	255
	3...4	2	32 767
	5...7	3	16 777 215
	8...9	4	2 147 483 647
I	1...4	2	32 767
	5...9	4	2 147 483 647
R	1...7	4	16 777 215
	8...14	8	2**24-1
D	1...15	$[(S+2)/2]$	
X	1...255	$[(S+1)/2]$	
T	1...100	S	

Põhiline osa mingi programmi või programmide komplekti jaoks vajalikest legendidest koostatakse juba projekteerimisel. Sellest, kui võrd hästi need legendid kirjeldavad antud probleemi olemust ja vastavad süsteemi STAMP võimalustele, sõltub oluliselt kogu komplekti realiseerimise lihtsus ning kasutamise efektiivsus. Seepärast esitame mõningad soovitused, mida peab legendi kirjutamisel silmas pidama.

Kasutamise iseloomu järgi võib kirje elemendid jagada kolme rühma: arvud, koodid ja tekstid. Arvud on sellised elemendid, milledega üldjuhul tuleb sooritada aritmeetilisi tehteid. Tehte sooritamise ajal peavad temast osa võtvad arvud olema samas arvusüsteemis. Et vältida ülearseid vahetisendusi ühest arvusüsteemist teise, on soovitatav määrata arvude jaoks legendis ühesugune tüüp. Süsteemis STAMP on põhitüübiks valitud kahendtäisarvud: kuigi neli aritmeetilist tehet on realiseeritud kõigi arvutüüpide jaoks, kasutab enamik spetsiaalseid operatsioone ainult kahendtäisarvude aritmeetikat. Muud arvude esitusviise läheb vaja põhiliselt andmete vahetamisel teiste süsteemidega.

Koode kasutatakse objektide tähistamiseks lühendatud kujul ja neid võib esitada kas N, D või X-tüüpi arvudena või ka tekstina. Kui pole mingeid erilisi kitsendavaid tingimusi, siis on soovitatav kasutada koodidena 16-ndarvuseid. Nimelt on koodid praktikas sageli valitud selliselt, et ka koodi üksikud numbrkohad omavad sisulist tähendust. Kui element on kirjeldatud N-tüüpi arvuna, siis süsteemis puudub võimalus sellise koodi väärtust komponentide kaupa "lahti pakkida". Kümndarvude põhiliseks puuduseks on aga asjaolu,

et nendega opereerivad käsud annavad "puhta" nulli korral tõrke. Null on aga tüüpiline väärtuse puudumise tunnus.

Koodi esitamisel tekstina tuleb silmas pidada, et tekst paigaldatakse alati vastava välja algusse. Võtmelemendi korral võib niiviisi mõningatel juhtudel saada ebasoovitava eksemplaride või kirjete järjekorra. Kui näiteks kirjes KLASS oleks klassi number antud tekstina, siis saaksime kirjetele sellise järjestuse: 1A,...,1D,10A,...,11B,2A,...,9C.

Tekstide korral on üldse soovitatav järele mõelda, kas neid pole antud kirjesse lülitamisel hoopis otstarbekohasem asendada koodiga. Näiteks võiks kirjes KLASS õpilase perekonna- ja eesnime asemel kasutada õpilase järjekorranumbrit klassipäevikus, nimed ja muud püsivad andmed aga siestada omaette kirjena.

Võtmelementide paigutamisel legendi peab arvestama, et nende järjekord (taseme piires) määrab kas eksemplaride järjestuse kirjes või kirjete järjekorra failis. Samuti tuleb kontrollida, kas valitud võtmelemendid ikka määravad eksemplari üheselt või siis on hoopis mõni element ülearuselt võtmesse lisatud.

Suurte kirjete korral tuleb kontrollida, kas kirje maht maksimaalsel võimalikul juhul ei tule üle 32K. Kirjes KLASS on teise ja kolmanda taseme eksemplaride pikkused vastavalt 36 ja 8 baiti (sealhulgas viitadeks 6 ja 2 baiti). Kui aine- te arv on näiteks 16, siis ühe õpilase andmete jaoks kulub kokku $36+8*16=164$ baiti. Seega peab klassi õpilaste arv olema mõnevõrra väiksem kui 200 - niisugune tingimus on ilmselt alati täidetud.

3. ANDMETE SISESTAMINE

Süsteemis STAMP sisestatakse kõik andmed kindla skeemi järgi, mis on määratud kirje legendi ja kasutatava sisestamisoperatsiooniga. Nende operatsioonide komplekti koos vastavate vormistamisreeglitega võib nimetada süsteemi STAMP sisestamiskeeleks.

Sõltumata sellest, kas tegemist on uue kirje lisamisega andmebaasi või paranduste viimisega varem sisestatud kirjesse, tuleb sisestatavad andmed kõige üldisemal juhul vormistada sisestamiskeelse lausena järgmiselt:

```
// <operatsioon> <kirje nimi> <1. taseme elemendid>  
/  
: <3. taseme elemendid>  
.....  
: <3. taseme elemendid>  
/  
: <2. taseme elemendid>  
.....
```

Iga eksemplari piires eraldatakse elemendid üksteisest tühikutega ja nad peavad tingimata paiknema legendis näidatud järjekorras. Fiktiivsed elemendid (legendis omadus P) jäetakse eksemplari koosseisus vahele. Kui elemendil mingis konkreetses eksemplaris puudub sisuline väärtus või see on andmete sisestamise hetkel teadmata, siis kirjutatakse elemendi väärtuseks null. Erandi moodustavad taseme viimased lisaelemendid, mis võib puudumise korral lihtsalt ära jätta.

Korduva elemendi korral kasutatakse üksikkomponentide eraldamiseks plussmärki. Kui komponentide arv on maksimaal-

sest väiksem, siis võib puuduvad komponendid lõpust ära jätta (puudevate komponentide väärtuseks omistatakse null sõltumata elemendi tüübist). Ära võib jätta ka vahepealsed komponentide puudumist tähistavad nullid, kirjutades mitu plussi järjest.

Arvu täis- ja murdosa eraldatakse teineteisest komaga. Murdosa lõpust võib nullid ära jätta, kuid ühest väiksema arvu korral tuleb täisosa null alati näidata. Näiteks võib arvude 0,240 ja 45,000 asemel kirjutada 0,24 ja 45, kuid kirjutusviis ,24 pole lubatud.

Kui elemendi tekst ise sisaldab niisuguseid eraldajaid, mida võib tõlgendada järgmise elemendi, komponendi või tase-me algusena, siis tuleb vastav tekst panna ülakomade vahele. Sellisteks eraldajateks on tühik, kaldkriips, koolon kolmandal tasemel ja pluss korduva elemendi koosseisus. Kahte järjestikust kaldkriipsu pole elemendi koosseisus üldse lubatud kasutada.

Uued kirjed sisestatakse operatsiooniga L (= lisada) või S (= salvestada). Kirje lisamisel kontrollib süsteem, kas antud võtmeväärtusega kirjet andmebaasis juba ei leidu. Kui selline kirje on olemas, siis antakse vastavasisuline teade ja kirjet ei sisestata. Operatsiooni S korral sisestatakse kirje igal juhul, s.t. kas lisatakse, kui teda varem polnud või asendatakse, kui selline kirje eelnevalt oli olemas.

Järgnevates näidetes on võetud aluseks eelmises paragrahvis toodud legend KLASS (vt. lk. 15). Sisestame näiteks andmed 3A klassi kohta, piirdudes ainult kahe õpilase ja kolme esimese ainega:

//L KLASS 3A 'AASA ANNE' 10

/AAV ARVI 19760230 :1 4+4+5+4 :2 3+4+4+4 :3 4+4+3+4

/PAJU FILLE 19760608 :1 5+5+5+5 :2 5+5+4+5 :3 5+5+5+5

Selle lausega alustatakse kirje KLASS moodustamist 3A klassi õpilaste jaoks. Esimesel tasemel on kirje nime järel antud vastava taseme kõik elemendid: klassi number, klassijuhataja nimi ja ainete arv. Teise taseme elementidest tuli muidugi ära jätta keskmine hinne, mis legendis on kirjeldatud pseudoelemendina. Kolmandal tasemel antakse aine number ja korduv element HINNE, mille iga komponent näitab üht veerandihinnet.

Kirjeid saab kustutada operatsiooniga K, andes ette vaid kustutatava kirje nime ja esimese taseme võtmeelemendid. Näiteks sisestamiseks lausega

//K KLASS 4E

tehakse korraldus kustutada kirje KLASS 4E klassi andmetega.

Täienduste ja paranduste viimiseks varem sisestatud kirjetesse tuleb operatsiooni nimes näidata veel taseme number, näiteks L2. Parandatava kirje määramiseks on aga esimesel tasemel nüüd tingimata vaja ette anda kõikide võtmeelementide väärtused (ja ainult need).

Teise või kolmanda taseme uued eksemplarid lisatakse vastavalt operatsioonidega L2 või L3. Seejuures tuleb teise taseme eksemplari lisamisel talle alluvad kolmanda taseme eksemplarid kirjutada vahetult selle eksemplari järele. Operatsiooni L2 korral võib aga kolmanda taseme eksemplarid jätta ka hoopis andmata ja lisada need hiljem omaette operatsiooniga L3. Näiteks lausega

//L2 KLASS 3A

/SUSI SULEV 19760412 :1 4+4+3+3 :2 4+4+4+3

/MAASIKAS MARJU 19760913

lisatakse 3A klassi jaoks moodustatud kirjesse KLASS kaks uut teise taseme eksemplari, kusjuures MAASIKAS MARJU veerandihinded on näitamata.

Lisatav eksemplar paigutatakse kirjesse võtmeväärtusega määratud kohale. Kui selline eksemplar on juba olemas, siis antakse teade ja eksemplari ei lisata. Kolmanda taseme eksemplaride lisamisel tuleb peale võtme näidata ka teise taseme võtmeelementide väärtused, näiteks

//L3 KLASS 3A /MAASIKAS MARJU :1 4+5+5+4 :2 5+4+5+4

Eksemplaride asendamiseks teisel või kolmandal tasemel kasutatakse operatsioone S2 ja S3. Kui asendatav eksemplar kirjes puudub, siis vastav eksemplar lisatakse. Teise taseme eksemplari asendamisel tuleb aga arvestada, et selle eksemplari alluvuses olnud endised kolmanda taseme eksemplarid lähevad kõik kaduma. Nende kirjete jaoks, kus legend määrab võtmelemendid nii teisel kui ka kolmandal tasemel, on andmete esitamise skeem lisamisel ja salvestamisel praktiliselt ühesugune.

Näiteks õpilase SUSI SULEV jaoks sisestasime me ülalpool juba esimese ja teise aine veerandihinded. Seega uue parandusega kujul

//S3 KLASS 3A /SUSI SULEV :2 3+3+3+4 :3 4+3+4+3

asendatakse teise aine ja lisatakse kolmanda aine hinded.

Eksemplaride kustutamiseks tuleb ette anda ainult võtmelemendid. Näiteks lausetega

//K2 KLASS 3A /MAASIKAS MARJU

//K3 KLASS 3B /PALU PAUL :7

kästakse 3A klassi jaoks moodustatud kirjest kustutada teise taseme eksemplar MAASIKAS MARJU ja 3B klassi jaoks moodustatud kirjest õpilase PALU PAUL seitsmendale ainele vastav kolmanda taseme eksemplar (kui selliseid andmeid pole varem üldse sisestatud, siis annab süsteem teate eksemplari puudumise kohta).

Asendamisoperatsioonidega A1, A2 ja A3 saab muuta vastavalt esimese, teise või kolmanda taseme üksikute elementide väärtusi. Selleks tuleb ette anda eksemplari võtmeelementide väärtused ja asendatava elemendi nimi koos uue väärtusega. Samas eksemplaris võib korraga (ühes lauses) muuta mitme elemendi väärtust. Näiteks lausega

//A2 KLASS 3A /AAV ARVI ENIMI ARVO SKUUP 19760330

omistatakse õpilase AAV ARVI uueks eesnimeks ARVO ja parandatakse ühtlasi tema vigane sünnikuupäev. Järgmistes paranduslausetes tuleb nüüd see eksemplar tingimata määrata juba võtmega AAV ARVO. Põhimõtteliselt võib muuta ka kirje esimese taseme võtmeelementide väärtusi, kuid sellega kaasneb uue kirje tekkimine. Seepärast vaatleme niisugust juhtu mõnevõrra hiljem, kui räägime failide parandamisest.

Parandada võib ka korduva elemendi üksikut komponenti. Komponenti järjekorranumber antakse sel juhul koos elemendi nimega, eraldades nime numbrist punktiga. Näiteks lausega

//A3 KLASS 3A /PAJU PILLE :2 HINNE.3 5

parandatakse PAJU PILLE teise õppeaine kolmanda veerandi hinne viieks.

Paranduste korral on loomulikuks eelduseks parandatava kirje olemasolu. Erandi moodustavad aga siiski operatsioonid L2 ja L3, mille korral parandatava kirje puudumisel genereeritakse automaatselt uus antud võtmeväärtustega kirje. Seda erandit saab ära kasutada näiteks niisuguste konkreetsete ülesannete puhul, kus kirjeid tuleb moodustada pikema perioodi vältel üksikute eksemplaride pideva lisamise teel ning lisamise hetkel pole alati teada, kas uut kirjet on juba alustatud või mitte.

Sisestatavad parandused vormistatakse esialgu spetsiaalsete kirjetena, milles iga konkreetsete võtmeväärtustega eksemplar võib esineda ainult ühekordselt. Seepärast tuleb operatsioonide A2, A3, L3 ja S3 korral jälgida, et samasse eksemplari või ühe konkreetse eksemplari alluvatesse eksemplaridesse tehtavad parandused oleksid kokku koondatud, s.t. teostatud järjest.

Vaatleme järgnevalt parandamise mõningaid iseärasusi neis kirjetes, kus teisel või kolmandal tasemel pole võtmelemente üldse määratud. Sellisel juhul loetakse võtmeks eksemplari järjekorranumber, s.t. mitmendana see eksemplar kirjesse sisestatakse. Üheks niisuguseks kirjeks on legendi teksti enda sisestamiseks määratud kirje nimega LEG, mille legend on süsteemis STAMP ette antud järgmiselt:

LEG LEG

1 NIMI T-K

2 RIDA T-V EL. KIRJELDUS

Esimese taseme elemendi NIMI väärtuseks oleva nimega legendi iga elemendi tarvis on kirjes LEG ette nähtud üks tei-

se taseme eksemplar, mis koosneb ainult ühest muutuva pikkusega elemendist RIDA (selle väärtuseks ongi legendi järjekordse rea tekst).

Vaatleme juba tuttavat legendi KLASS, kuhu mitmesugustel põhjustel oleks vaja sisse viia muudatusi. Lisame kõigepealt ühe elemendikirjelduse lausega

//L2 LEG KLASS

/EHINNE N1-L EKSAMIHINNE

Et kirjes LEG puudub teisel tasemel võti, siis lisatakse see antud taseme viimaseks eksemplariks, s.t. kolmanda taseme elemente AINE ja HINNE kirjeldavate ridade järele. Paneme tähele, et muutuva pikkusega elemendi RIDA väärtuse sisestamisel on kasutatud sisestamiskeeles lubatud lihtsustusreeglit: kui mitmest sõnast koosnev või eraldajaid sisaldav teise taseme element on sellel tasemel viimane ja kolmas tase puudub, siis võib elemendi algust ja lõppu tähistavad ülakommad ära jätta.

Eksemplari asendamisel tuleb võtmeelemendi puudumise korral ära näidata asendatava eksemplari järjekorranumber. Näiteks legendi KLASS neljanda ja viienda rea asendamine uudega tuleb vormistada järgmiselt:

//S2 LEG KLASS

/4 2 PNIMI T14-K

/5 ENIMI T14-K

Kirjes LEG on oluline eksemplaride (s.o. legendi elemendikirjelduste) järjekord. Ilma võtmeta eksemplari saab sobivasse kohta lisada sisestamisoperatsiooniga V (= panna vahele). Näiteks lause

//V2 LEG KLASS

/3 AINED T15-V=16

lisab kirjesse LEG võtmeväärtusega KLASS uue kirjeldusrea kolmanda eksemplari järele, s.t. esimese taseme viimase elemendi kirjelduseks. Uue eksemplari järjekorranumbriks omistatakse samuti 3. Et ülejäänud eksemplaride numeratsioon seetõttu ei muutu, saab samas paranduste seerias näiteks kustutada elemendi KH kirjelduse lausega

//K2 LEG KLASS /7

Uued järjekorranumbrid omistatakse eksemplaridele alles siis, kui vastava nimega parandatud kirje LEG järgmisel korral sisemällu loetakse.

Sisestamisoperatsioonide hulka kuuluvad veel P (= sisestada pseudokirje) ja T (= tähistada kirje).

Operatsiooniga P sisestatakse sellised kirjed, mida tahetakse kasutada ainult antud lahendusseansi jooksul. Need kirjed on küll seansi ajal kõik kättesaadavad üldistel alustel, kuid andmebaasi salvestamise käigus jäetakse nad vaatluse alt välja.

Operatsiooniga T saab välja eraldada ehk tähistada üksikuid kirjeid kõigi samanimeliste kirjete hulgast. Tähistatud kirjete võtmed paigutatakse spetsiaalsesse kirjesse, mida kasutavad mõned kirjete kontrollrüki ja ülekandmise süsteemsed programmid.

Andmete sisestamisel saab kasutada veel mõningaid lihtsustusreegleid. Näiteks kui kolmandal tasemel on suhteliselt vähe elemente, siis võib taseme algust tähistava kooloni kooni ära jätta. Seejuures tuleb aga tingimata silmas pidada,

et vea esinemise korral mingis ühes kolmanda taseme eksemplarid sisestamata ka kõik järgmised selle taseme eksemplarid.

Sageli on kirje teise või kolmanda taseme mõnel elemendil niisugune väärtus, mis kordub ka paljudes järgnevates eksemplarides. Sellise korduva väärtusega elemendi ette võib panna punkti ja jätta ühtlasi järgnevates eksemplarides see element kas lihtsalt vahele või tähistada väärtuse samaks jäämist ainult punktiga. Väärtuse muutumisel tuleb nüüd uus väärtus, kui ka see jääb mitmes järgmises eksemplaris püsima, anda jälle punktiga. Kui aga antud element edaspidi enam ei omanda püsivat väärtust, siis tuleb järjekordse väärtuse ette panna kaks punkti ja edaspidi muidugi vastavad punktid ära jätta.

Järgnevas spetsiaalselt konstrueeritud näites toimub vasakus tulbas kirje AB sisestamine ilma ja paremas tulbas koos punkti kasutamisega (vastav legend pole siinkohal oluline, mistõttu jätame ta esitamata):

//L AB	//L AB
/1 AAAAA 41763 16	/.1 .AAAAA .41763 16
/1 AAAAA 41763 28	/28
/1 AAAAA 41763 42	/42
/1 BBBBB 37259 11	/. .BBBBB .37259 11
/1 BBBBB 37259 36	/36
/2 BBBBB 54730 47	/.2 . .54730 47
/2 CCCCC 54730 16	/. . .CCCCC 16
/2 DDDDD 54730 21	/DDDDD 21
/2 EEEEE 54730 40	/EEEEEE 40

4. FAILID JA FONDI KIRJELDAMINE

Kõik andmebaasi kuuluvad kirjed peavad olema ühendatud failideks. Failid süsteemis STAMP on operatsioonisüsteemi seisukohalt jadafailid ning koosnevad fikseeritud pikkusega plokkidest. Ploki pikkuseks on valitud 1600 baiti. Kirjed paiknevad failis vahetult üksteise järel, kusjuures nii kirje kui ka eksemplar kirjes võib jätkuda ühest plokkist teise.

Kirje füüsiline esitus välisseadmetel erineb mõnevõrra nende esitusest sisemälus. Välismällu viidud kirje algab küll päise ja esimese taseme eksemplariga, kuid puuduvad viidad alluvatele, naabereksemplaridele ning muutuva pikkusega elementidele. Eksemplari päist asendab nüüd kahebaidine väli, mille esimeses poolbaidis on taseme number ja kolmes järgnevas poolbaidis eksemplari pikkus. Eksemplari koosseisu kuuluvad muutuvad elemendid paiknevad vahetult eksemplari järel legendiga määratud järjekorras. Kui kirjes on ka kolmas tase, siis igale teise taseme eksemplarile järgnevad talle alluvad kolmanda taseme eksemplarid.

Kasutamise iseloomu järgi võib andmefailid süsteemis STAMP rühmitada järgmisel viisil.

1. Püsifailid kirjete pidevaks säilitamiseks:
 - a) põhifailid (kasutajate poolt antud nimedega);
 - b) kollektor (süsteemisese nimega COLL).
2. Lahendusseansi piires kasutatavad ajutised failid:
 - a) sisendfail (nimega SISE);
 - b) tööfail (nimega TQQ).

Ühte põhifaili võivad kuuluda kas ainult kõik ühe nimega

kirjed või ka mitme erineva nimega kirjed. Kirjed paiknevad põhifailis võtmeväärtuste kasvavas järjekorras. Erinimeliste kirjete järjestamiseks samas failis tuleb iga loodava faili jaoks määrata võtme maksimaalne pikkus ja seada igale kirje nimele vastavusse kirje järjekorranumber failis. Kui tegeliku kirje võtme pikkus osutub maksimaalsest väiksemaks, siis täiendab süsteem seda lõpust nullidega. Kahest erinimelisest sama võtmeväärtusega kirjest paikneb failis eespool alati see kirje, mille järjekorranumber on väiksem.

Kirjete otsimine põhifailist toimub faili järjestikuse läbivaatamise teel. Selline kirjete lugemine sobib neil juhtudel, kus võtmeväärtuste järjekorras tuleb läbi vaadata enamik failis olevatest kirjetest. Kui aga failist loetakse vaid üksikuid kirjeid või kirjeid on vaja lugeda juhuslikus järjekorras, siis võib põhifailile lasta moodustada indeksi. Indeksisse kantakse iga ploki viimase kirje võti ja ploki number, indeks ise aga salvestatakse faili viimasesse ploki. Et indeks peab ära mahtuma ühteainsasse plokki, siis küllalt suure faili korral tehakse indeks suurema sammuga, s.t. ploki viimase kirje võti ja asukoht failis näidatakse mitte iga ploki jaoks, vaid üle ühe, kahe või enama ploki.

Uute kirjete või paranduste lisamiseks põhifaili tuleb fail üldiselt ümber kirjutada sellest plokist alates, kuhu tehti esimene muudatus. Üksikute täienduste korral on niisugune lähenemiski viis liialt kulukas. Seepärast kuulub püsifailide hulka veel kõigi põhifailide jaoks ühine spetsiaalse struktuuriga fail - kollektor. Põhifaili lisamise asemel võib üksikud kirjed salvestada kollektorisse, kust need põ-

hifailidesse üle kantakse alles siis, kui kollektorisse on kogunenud juba piisaval hulgal kirjeid.

Uued kirjed lisatakse alati kollektori lõppu. Iga faili kirjete järele salvestatakse kollektorisse veel selle faili kataloog. Kataloogis on andmed kõigi kollektoris leiduvate kirjete kohta: kirje võti ja järjekorranumber failis, ket-
taaadress kollektoris ning salvestusoperatsioon (L, S või K). Faili kataloog on kollektoris vormistatud omaette kirje-
na, mille teise taseme iga eksemplar kirjeldab üht kollektori asuvat ja antud faili kuuluvat kirjet.

Kollektori viimasesse plokki kirjutatakse veel sisukord, kus on viidad failide kataloogidele. Järgmisel kollektorisse kirjutamisel säilitatakse ka eelmine sisukord, mis võimaldab vajaduse korral taastada kollektori endist seisust. Informat-
siooni paiknemist kollektoris illustreerib järgnev joonis:

F1	C1	F2	C2	SK	F1	C1	SK
----	----	----	----	----	----	----	----

kus on tähistatud: F1 (F2) - esimese (teise) faili kirjed, C1 (C2) - esimese (teise) faili kataloog, SK - sisukord.

Kõik sisestatavad kirjed salvestatakse esmalt antud lahendusseansi jaoks loodud ajutisse sisendfaili. Välisseadmetelt (perfolint, magnetlint, kuvari ekraan) sümbolkujul saabuv info teisendatakse sealjuures kõigepealt arvutisisesele kujule. Samuti viiakse sisendfaili ka töötusprogrammides genereeritavad kirjed. Andmed sisendoperatsioonidega L,S,P või K salvestatud kirjete kohta kantakse ühtlasi vastavasse kataloogi: iga faili jaoks luuakse omaette kataloog selle faili kirjete esmakordsel salvestamisel sisendfaili. Paran-

duskirjete (operatsioonid A1, L2 jne.) asukohta sisendfailis kataloogides ei fikseerita.

Kirjete püsivaks säilitamiseks tuleb nad ajutisest sisendfailist ümber kirjutada kas põhifailidesse või kollektorisse. Seda tehakse tellimuses antava korralduse põhjal erilise süsteemse programmiga OUT. Lisaks asendatavate ja lisatavate kirjete ülekandmisele või kirjete kustutamisele on programmi OUT ülesandeks veel paranduste tegemine varem sisestatud kirjetesse, tuginedes sisendfailis olevatele paranduskirjetele. Põhifailide muutmisel kasutab programm OUT sealjuures teist ajutist faili - tööfaili.

Iga välisseadmele salvestatava kirje jaoks peab olema määratud fail, kuhu see kirje loogiliselt kuulub, kuigi füüsiliselt võib vastav fail isegi puududa. Seose kirjete ja failide vahel määrab kasutaja erilise kirjega TNT. Igale kirje nimele seatakse seal vastavusse seadme ja faili nimi, mis sisaldavad antud nimega kirjeid. Kirje TNT legend genereeritakse süsteemi poolt kujul

LEG TNT

1 SIFFER T8-K

KASUTAJA T-V=P

2 FNR N3-K

FNIMI T8

FT N2

IKNR N3

VMP N2

3 KNR N3-K

KNIMI T8

Kirje TNT võtmeelement SIFFER on ühtlasi antud kasutaja isikliku andmebaasi ehk fondi nimeks. Sellesse andmebaasi kuuluvad need ja ainult need failid, mis on loetletud antud kirjes TNT.

Elemendiga KASUTAJA võib anda loodava fondi kasutaja (omaniku) täieliku nime, mida läheb tarvis aga üksnes väljundtabelite kujundamisel.

Kirje TNT teise taseme eksemplariga võib kirjeldada kas faili või infokandja, kusjuures kirjelduse liigi määrab element FNR. Kui selle elemendi väärtus kuulub vahemikku 1...99, siis on tegemist failikirjeldusega, kui aga vahemikku 100...255, siis infokandja kirjeldusega. Kasutaja võib oma faile siiski nummerdada vaid alates 6-st, sest numbrid 1...5 on süsteemi poolt reserveeritud (nende numbrite tähendus selgub lk. 40 toodavast kirjest).

Element FNIMI määrab olenevalt FNR väärtusest kas faili või infokandja nime. Faili nimi peab olema fondi piires uniikaalne ja see seatakse üksühesesse vastavusse faili numbriga FNR, nii et süsteemi programmides saab nime asemel kasutada faili numbrit.

Faili tunnuse FT väärtuseks võib olla vaid 0 või 1. Failikirjelduse korral tähendab 0 tavalist ja 1 indeksiga varustatud faili, infokandja kirjelduses aga määrab 0 kettafaili ja 1 magnetlindile kantava faili.

Infokandja numbri IKNR väärtuseks võib olla kas 0 või siis arv vahemikust 100...255. Väärtus 0 tähendab, et fail asub kettal, mille nimi täpsustatakse alles vastava lahendusprotseduuri parameetriga. Igasugune muu väärtus viitab

aga samas kirjes TNT määratud seadmele. Väärtust O kasutatakse aga ka sellisel juhul, kui kirjeldatav fail füüsiliselt puudub.

Elemendi VMP väärtus näitab kirje võtme maksimaalse pikkuse (baitides) antud failis. See väärtus peab tingimata olema paarisarv. Infokandjat kirjeldavas eksemplaris on elementide IKNR ja VMP väärtuseks null.

Kirje TNT kolmanda taseme eksemplarid antakse ainult failikirjelduste korral. Iga kolmanda taseme eksemplariga teatatakse süsteemile, millise nimega (elemendi KNIMI väärtus) kirjed kuuluvad sellesse faili ja seatakse kirje nimele vastavusse number (elemendi KNR väärtus), mida muide süsteemisiseselt ka nime asemel kasutatakse. Need numbrid määravad ühtlasi erinimeliste kirjete järjestuse failis, kui nende kirjete võtmeväärtused kokku langevad.

Kirjete määramisel kas ühte või siis eraldi failidesse tuleb silmas pidada mitmesuguseid asjaolusid. Samasse faili paigutatakse eeskätt sellised ühesuguste võtmelementidega kirjed, mida lahendusprogrammides kasutatakse samaaegselt. Erinimeliste kirjetena on nad vormistatud kas seepärast, et sisestamine toimub erinevatelt dokumentidelt, või siis ei saa neid ühise kirjena vormistada süsteemi STAMP kirjete kolmetasemelise struktuuri tõttu.

Samasse faili ühendatakse sageli ka niisugused kirjed, kus ühe kirje võtmeelementid on samad, mis teise kirje esimesed võtmeelementid, kuid teises kirjes leidub lisaks veel üks või mitu võtmelementi. Esimest kirjet võib sellisel juhul vaadelda nn. juhtkirjena ja teisi kirjeid temale alluva-

te kirjetena. Juhtkirje number failikirjelduses peab vastava järjestuse saamiseks muidugi olema alluvate kirjete omadest väiksem.

Failide planeerimisel tuleb arvestada ka nende täiendamise ja uuendamise sagedust. Näiteks suhteliselt püsivad kirjed tuleks igapäevaselt täiendatavatest eraldi paigutada. Siinjuures peetakse eeskätt silmas duublite tegemise vajadust, s.t. faili kopeerimist magnetlindile iga järjekordse uuendamise järel, et andmete võimaliku rikkemise korral saaks neid lihtsamalt taastada.

Omaette failidesse on soovitatav koondada veel niisugused kirjed, mis teatud perioodi (kuu, kvartali) lõppedes kuuluvad kustutamisele. Lihtsam on nimelt tühjendada kogu fail, selle asemel, et eemaldada mingist failist kõik teatud nimega kirjed.

Ühisesse faili võib paigutada ka sellised erinimelised kirjed, mida fondi kavatsetakse sisestada kas ainult üks või siis mõni üksik eksemplar. Neil ei tarvitse sealjuures olla omavahelisi seoseid, kuid üldiselt pole otstarbekohane luua palju väikseid faile.

Kõigile kirjes TNT nimetatud failidele ei tarvitse fondis füüsilist faili vastata - nende failide kirjed võib säilitada kollektoris. Näiteks uue programmide komplekti väljatöötamise ajal võib kogu fond koosneda ainult kollektorist, mis sisaldab lähteandmetest vaid üksikuid silumiskirjeid. Kui kollektor kirjete järjest uute versioonide lisamisega täis saab, siis tuleb sooritada n.ö. kollektori kokkusuurumine, mille tulemusena säilitatakse kõigest kollektoris

olevatest kirjetest ainult viimane versioon. Varem juba põhifailidesse kantud kirjed lähevad kokkusuurumisel muidugi kollektorist üldse kaduma.

Sisendfaili saab salvestada ka niisuguseid kirjeid, mille nimi puudub kirjes TNT. Sellised kirjed loetakse automaatselt faili SISE kuuluvateks ja nende jaoks luuakse eraldi kataloog. Kirjes TNT fikseerib süsteem ühtlasi ajutiselt nende kirjete nimed, kuid võtme pikkus ei tohi seejuures olla üle 8 baidi.

Ühe faili kirjed ja isegi mingi konkreetne kirje võivad samaaegselt olla nii sisendfailis, kollektoris kui ka põhifailis. Kui kirje lugemisel pole faili ära näidatud, siis otsitakse seda kirjet fondist just ülalnimetatud järjekorras, s.t. sisendfaili kirjet eelistatakse kollektori omale, viimast aga põhifaili kirjele. Kõigi samanimeliste kirjete läbivaatamisel peetakse eraldi järge sisendfaili ja kollektori kataloogis ning põhifailis.

Kõigi süsteemis STAMP kasutatavate püsifailide nimed peavad koosnema kahest komponendist, milleks on fondi nimi ja faili nimi kirjes TNT määratud kujul. Kui faili tegelik nimi on teistsugune, siis tuleb antud fail kirjeldada vasta-va DD-kaardiga.

Kirjed TNT peavad asuma omaette failis. Niisuguse ühekirjelise faili võib luua eraldi iga fondi jaoks. Siiski osutub kasulikumaks koondada kõikide fondide kirjed TNT kokku ühisesse faili, mis paikneb ühel pidevalt kasutusel oleval kettal ning on kirjeldatud kõigis süsteemi STAMP lahendusprotseduurides.

Uue fondi loomisel tuleb kõigepealt sisestada andmed, mida hiljem vajatakse selle fondi sisu moodustavate kirjete sisestamisel. Nendeks andmeteks on kirje TNT ning kirjete TNT ja LEG legendid. Selleks tuleb täita lahendusseanss, mis sisaldab tellimuse kujul

```
//TELLIMUS-<fondi nimi>
```

```
/OUT R=P FN=TNT
```

```
/OUT KN=LEG
```

Muidugi peab sealjuures hoolitsema, et eelnevalt oleks loodud fail nimega COLL.<fondi nimi> (esitatud tellimuse üksik-osa tähtsust selgitame järgmises paragrahvis).

Kui täidetavas tellimuses teatatud fondi nimele vastavat kirjet TNT veel ei leidu, siis kirjutab süsteem automaatselt sisendfaili nimetatud legendid ja moodustab kirje TNT järgmise miinimumvariandi:

```
//L TNT <fondi nimi>
```

```
/1 SISE 0 0 8
```

```
/2 COLL 0 0 0
```

```
/3 TQQ 0 0 0
```

```
/4 TNT 0 0 8 :1 TNT
```

```
/5 LEG 0 0 8 :1 LEGEND :2 LEG
```

(Sellest miinimumvariandist selgub ühtlasi failinumbrite 1...5 tähendus süsteemis STAMP).

Elmist tellimust sisaldavas lahendusseansis tuleb veel sisestamisoperatsiooniga L2 näidata täiendused sellele süsteemselt genereeritud kirjele TNT ning lisada kõigi nende kirjete legendid, milliseid järgnevalt on kavas hakata andmebaasi sisestama.

5. TELLIMUS

Süsteemi STAMP lahendusseanss täidetakse operatsiooni-
süsteemi seisukohalt ühe sammuna. Tüüpiline kaardipakk töö
käivitamiseks on järgmine:

```
//<töö nimi> JOB ...  
// EXEC STAMP,VOL=<ketta nimi>  
//SYSIN DD DATA  
//TELLIMUS-<fondi nimi>  
/<tellimuse rida>  
.....  
///  
<sisestatavad kirjed>  
/*  
//
```

EXEC-lause parameetriga VOL näidatakse ketas, kus asuvad need kasutatavad failid, mille jaoks kirjes TNT on infokandja määrata. Antud fondi mitte kuuluvad failid kirjeldatakse DD-kaardiga EXEC-lause järel. Tellimuses endas antakse kõigepealt fondi nimi ja seejärel täidetavate tööde loetelu. Iga kaldkriipsuga algav tellimuse rida on tegelikult korraldus mingi programmi käivitamiseks, sisaldades programmi nime ja parameetrid. Sagedamini kasutatavate programmide korral võib programmi tegeliku nime asendada kokkuleppelise nimega. Näiteks nimede M1TRAN ja M2OUT asemel võib kasutada lühemaid variante TRAN ja OUT. Lubatud lühendamised on näidatud vastavate programmide juhendites. Programmi parameetrid antakse üldiselt kujul

<parameetri nimi>=<väärtus>,...,<väärtus>

kus parameetri nimi võib olla ühe- või kahetäheline (pikemast nimest arvestatakse ainult kaht esimest tähte). Milliseid parameetreid mingi programm vajab, see on öeldud vastava programmi juhendis. Sama parameetri nimi võib erinevates programmides olla erineva tähendusega, kuid üldkasutatavad on järgmised parameetrite nimed:

KN - kirje nimi,

FN - faili nimi,

P - STAMP-programmi nimi,

ST - tunnus kirjete või programmide nn. ühispiirkondade säilitamiseks sisemälus,

TT - tunnus tellimuse täitmise jätkamiseks tõrke korral.

Parameetri väärtuseks võib kirjutada kuni 8-sümbolilise teksti, arvu vahemikust 0...32767 või 16-ndarvu, kusjuures viimase tunnuseks on täht X, millele järgneb number (näiteks X10, X19861231, XOABC). Kui teksti tüüpi parameeter algab kas numbriga, tähe X ja numbriga või sisaldab hoopis mingeid tähtedest/numbritest erinevaid sümboleid, siis tuleb parameetri väärtus panna ülakomade vahele. Väärtuste loetelus võib koma asemel eraldajana kasutada ka tühikut.

Tellimuse ridu saab ka märgendada: märgendid võimaldavad muuta sammude täitmise järjekorda ning omistada mitmele sammule ühiseid parameetri väärtusi. Märgendamiseks lisatakse tellimusse rida, kus programmi nime asemel on arv vahemikust 1...255. Sellist rida ei vaadelda omaette sammuna, vaid nimetatud arvu loetakse järgnevate töösamude märgendiks. Kui arvu järel on antud parameetrid, siis need lisatakse järgne-

vate töösammude parameetritele. Seega näiteks võib ridade

/TRAN P=PROG5 /LAH P=PROG5

asemel tellimusse kirjutada read

/4 P=PROG5 /TRAN /LAH

Mingi töösammu täitmine võib üldiselt lõppeda järgmisel kolmel erineval viisil.

1. Lahendamine õnnestus. Sel juhul tuleb automaatselt täitmisele tellimuse järgmine rida.

2. Lahendamine lõppes avariisituatsiooniga, mille tekitas näiteks välisseadmete ületamatu tõrge, vaba sisemälu puudumine vms. Tellimuse täitmine katkestatakse ja trükitakse süsteemprogrammeerijale määratud informatsioon.

3. Lahendamine lõppes tõrkega. Tõrge võib olla tingitud vigadest lähteandmetes või vajalike andmete puudumisest (tellimuses pole antud parameetrit, puudub nõutav kirje, transleerimisel avastati vigu jne.). Tellimuse edasine täitmine sõltub nüüd märgenditest ja parameetri TT väärtusest. Parameetriga TT saab nimelt näidata märgendi, kust alates tuleb tõrke korral jätkata tellimuse täitmist. Kui selline märgend tellimuse järgnevates ridades puudub, siis töö lõpetatakse. Parameetri TT puudumisel jätkatakse tellimuse täitmist esimesest uue märgendiga reast, TT=0 korral aga tellimuse järgmisest reast.

Vahetult pärast tellimuskaartide sisestamist tellimust analüüsitakse ning moodustatakse kirje TELL, kus igale reale vastab üks teise taseme eksemplar ja igale parameetri nimele kolmanda taseme eksemplar. Kirje TELL asub kogu seansi välisel sisemälus ja on aluseks tööde täitmise organiseerimisel.

Kui algandmed sisestatakse perfolindilt, siis on sageli otstarbekohane sisestada ka tellimus perfolindilt. Sellisel juhul tuleb kaardipakis olevas tellimuses anda korraldus uue tellimuse täitmiseks. Seda võib teha näiteks reaga

/EX T=PL ,

mille toimeel tellimus nimega PL sisestatakse perfolindilt koos teiste andmetega tavalise kirjena kujul:

//L TELLIMUS PL /<tellimuse rida> /...

Korralduse EX põhjal sisestatakse süsteemne programm, mis analüüsib tellimust nimega PL ja lisab seal näidatud tööd kirjesse TELL järgmisena täidetavateks sammudeks. Tellimus PL võib aga ka omakorda sisaldada korraldust EX (seda võimalust on vaja näiteks varem sisestatud standardsete tellimuste täitmise organiseerimiseks).

Lõpuks anname nende süsteemsete programmide loetelu, mis leiavad tellimuses kõige enam kasutamist (sulgudega on programmi täielikust nimest eraldatud see osa, mille võib pöördumisel ära jätta):

(M2)IN - tuua andmed sisendfaili;

(M2)OUT - viia kirjed sisendfailist püsifaili;

(M1)TRAN - transleerida STAMP-programm;

(M1)TK - transleerida trükikirjeldus;

(M1)LAH - täita STAMP-programm;

(M2)PRINT - trükkida failis asuvate kirjete võtmed;

(M3)TR - trükkida kirjed;

(M3)LEG - trükkida legendid;

(M2)TEEN - osutada muud laadi teenus (tühjendada, kustutada või ümber nimetada fail, anda kirjete pikkused jne.).

6. KIRJETE VIIMINE ANDMEBAASI

Uute kirjete lisamine või andmebaasis olevate kirjete parandamine toimub kahes etapis. Esimesel etapil teisendatakse sisestamiskeeles esitatud andmed arvutisisesele kujule ja salvestatakse sisendfaili SISE. Teisel etapil viiakse kirjetesse parandused ning salvestatakse nii uued kui ka parandatud kirjed püsifailidesse. Vastavaid töid teevad tellimuses antud korralduste põhjal süsteemi koosseisu kuuluvad programmid IN ja OUT.

Programm IN võib sisestamiskeeles vormistatud andmeid sisse lugeda nii kaardipakist kui ka jadafailist või teekfailist. Kaardipakis paigutatakse andmed vahetult pärast tellimust kolme kaldkriipsuga kaardi järele. Vajaduse korral võib kaardipakis olevaid andmeid sealjuures jagada portsjoni-teks, eraldades portsjonid üksteisest kaardiga ///. Enne tellimuses näidatud tööde juurde asumist käivitatakse automaatselt programm IN esimese andmeportsjoni sisselugemiseks. Iga järgmise portsjoni sisestamiseks tuleb tellimuses juba anda korraldus kujul

/IN K=1

Jadafailina vormistatakse harilikult perfolintidelt sisestatavad andmed. Vastav fail tuleb kirjeldada DD-kaardiga ja andmete lugemiseks sellest failist anda korraldus

/IN PL=<DD-kaardi nimi>

Andmeid teekfailist sisestatakse tellimuse reaga

/IN N=<säiliku nimi>,<säiliku nimi>,...

Seejuures peab teek-andmekogum olema kirjeldatud kaardiga

//TEEK DD DSN=<teegi nimi>,DISP=SHR

Sisestamiskeele lausetena vormistatud andmed teisendatakse ja pakitakse sisendfaili kandmisel vastavalt legendile ning sisestamisoperatsioonile. Kirjega samanimeline legend peab sealjuures kas juba varem olema andmebaasis või siis sisestatama kirjena LEG enne esimest antud nimega kirjet. Vahetult pärast kirje LEG sisselugemist genereeritakse selle põhjal ka sama nimega kirje LEGEND.

Kirjete parandamine on süsteemse programmi OUT ülesandeks. Erandi moodustavad vaid kirjed TNT ja LEG, millesse muudatused viiakse koheselt. Seejuures moodustatakse parandatud kirje LEG jaoks ka vastav uus kirje LEGEND. Parandused muudesse kirjetesse vormistatakse paranduskirjetena, mis salvestatakse samuti sisendfaili. Kui ühes lahendusseansis viiakse sisse uusi kirjeid ja ühtlasi ka parandusi nendes samadesse kirjetesse, siis tuleb tingimata jälgida, et parandused sisestataks kirjetest hiljem - vastasel korral jäetakse parandused arvestamata.

Lähtetekstis avastatud vigade kohta trükitakse seansi protokollis teade ja lähteteksti lõik, kus vigase elemendi ette on paigutatud tunnusena sümbol #. Kui viga esineb operatsioonis, kirje nimes või esimese taseme eksemplaris, siis jääb kogu kirje sisestamata. Kui aga viga on kas teise või kolmanda taseme eksemplaris, siis jääb sisestamata ainult vigane eksemplar. Hoiatav teade "korduv eksemplar" tähendab, et samas kirjes juba esines sellise võtmeväärtusega eksemplar. Kirjesse lisatakse niisugustest korduvatest eksemplari-dest alati ainult viimane.

Programmi IN täitmise järel leidub sisendfailis kahte liiki kirjeid: nn. täiskirjed ja paranduskirjed. Täiskirjeteks on operatsioonidega L, S ja P salvestatud kirjed. Nende kirjete võtmed ja asukoht fikseeritakse ühtlasi sisendfaili kataloogides ning seetõttu on nad antud lahendusseansi järgnevatele programmidele kättesaadavad. Tinglikult võib täiskirjeteks lugeda ka operatsiooniga K antud korraldused kirjete kustutamiseks, mis samuti fikseeritakse kataloogis. Paranduskirjetena salvestatud muudatuste viimiseks kirjetesse ja kirjete salvestamiseks püsifaili tuleb täita programm OUT, mille poole pööratakse tellimuse reaga

/OUT [R=<režiim>] [KN=<kirjenimede loetelu>]

[FN=<failinimede loetelu>]

Programmi OUT töös osalevad kirjed võib jagada kolme rühma: täienduskirjed, lähtekirjed ja tulemuskirjed. Täienduskirjeteks võivad olla nii sisendfailis asuvad täiskirjed, kus operatsioon ei ole P, samuti aga kõik paranduskirjed ning teatud režiimide korral isegi kollektoris olevad kirjed. Konkreetselt saab täienduskirjeid programmi poole pöördumisel määrata parameetriga KN või FN: parameeter KN muudab nendeks loetletud nimedega kirjed, parameeter FN aga kõik nimetatud failidesse kuuluvad kirjed. Nende parameetrite puudumisel on täienduskirjeteks kõik sisendfailis (või ka kollektoris) olevad kirjed.

Lähtekirjeteks on need varem sisestatud kirjed, mille jaoks leidub sisendfaili salvestatud parandusi. Tulemuskirjeteks on lähtekirjetest paranduste tegemisel saadavad kirjed, aga samuti ka otseselt ülekantavad täienduskirjed.

Parameetriga R teatatakse programmi töörežiim, s.t. näidatakse, kust võtta täiendus- ja lähtekirjed ning kuhu salvestada tulemuskirjed. Režiimi määramisel kasutatakse tähisusi: S = sisendfail, C = kollektor, P = põhifail. Kui parameetri R väärtus on kahetäheline, siis näitab esimene täht, kust võtta täienduskirjed ja teine täht, kuhu viia tulemuskirjed (ühetäheline parameetri väärtus osutab just tulemuskirjete asukohale). Seost režiimi ja kirjete asukoha vahel näitab alljärgnev tabel:

R	täiendus- kirjed	lähte- kirjed	tulemus- kirjed
S	S	S,C,P	S
C	S	S,C,P	C
P	S,(C)	S,C,P	P
SS	S	S	S
SC	S	S,C	C
SP	S	S,P	P
CP	C	-	P,(C)
CC	C	-	C

Režiimi R=S kasutatakse tavaliselt just silumise ajal. Näiteks STAMP-programmi teksti või trükikirjelduse parandusi katsetatakse seni, kuni tulemus rahuldab - alles seejärel salvestatakse lõplik tekst või trükikirjeldus püsifaili. Režiim R=S on vajalik ka veel kirje esimese taseme võtmeelemendi väärtuse parandamisel. Võtmeelemendi muutmise tekitatakse nimelt sisendfaili uus kirje, mis järgneval püsifaili salvestamisel paigutatakse juba uue võtmeväärtusega määratud kohale.

Parameetri R puudumisel loetakse režiimiks R=C.

Režiimi R=P korral parandatakse ainult neid faile, mille jaoks sisendfailis leidub täienduskirjeid. Lisaks sisendfaili kirjetele kantakse põhifaili ka parandatava faili kollektoris olevad kirjed. Kollektorist võetud kirjed ühtlasi kustutatakse kollektori kataloogidest ja uuendatud kataloogid kirjutatakse kollektori lõppu.

Režiimis R=CP kantakse parameetriga FN või KN määratud kirjed kollektorist põhifailidesse, kusjuures ülekantud kirjed ühtlasi kustutatakse kollektori kataloogidest. Kui parameetreid FN ja KN pole antud, siis kantakse põhifailidesse kõik need kollektoris leiduvad kirjed, mille jaoks põhifailid on juba loodud. Alles jäänud kirjed lükatakse kokku kollektori algusesse.

Režiimi R=CC kasutamisega saab kollektorit kokku lükata ka ilma kirjeid põhifailidesse kandmata. Kirjete viimased versioonid kogutakse kõigepealt tööfaili, seejärel kollektori sisu kustutatakse ja sinna kantakse tööfailis olevad kirjed. Kollektoris säilitatakse seejuures ainult need kirjed, mis on kirjeldatud kirjes TNT. Seega kui kirjest TNT ajutiselt kustutada mingi teise või kolmanda taseme eksemplar, saab režiimi R=CC abil kõrvaldada kollektorist kas kõik antud faili kuuluvad või siis antud nimega kirjed.

Tööfaili kasutatakse ka kirjete salvestamisel põhifaili. Nimelt kantakse põhifaili kirjed alates esimesest parandatavast plokist üle tööfaili, viies seejuures parandatavatesse kirjetesse sisse muudatused, lisades uusi või jättes vahele kustutatavaid kirjeid. Lõpuks dubleeritakse tööfaili sisu tagasi põhifaili.

7. VÄLJUNDTABELITE KIRJELDAMINE

Andmetöötlusprogrammi üheks tüüpiliseks ülesandeks on koostada ja välja trükkida andmebaasis olevate kirjete põhjal üks või mitu väljundtabelit. Tabeli trükkimiseks tuleb ette anda nii tabeli vormistamise eeskiri kui ka tabelisse kantavad andmed. Trükieskirja genereerib kasutaja poolt koostatud erilise trükikirjelduse põhjal süsteemne programm M1TK, kõik tabelisse trükitavad andmed tuleb aga eelnevalt koondada ühte nn. väljundkirjesse.

Trükikirjelduse translaator M1TK nõuab väljastatavate tabelite teatud määral standardset vormistamist. Iga süsteemis STAMP lubatav väljundtabel koosneb üldiselt pealkirjast, päisest, tabeli ridadest (sisust) ja lõpposast. Tabeli pealkiri võib sealjuures sisaldada ühe või mitu rida, mille järele enne tabeli päist saab vajaduse korral paigutada ka veel mõningaid üldandmeid tabeli kohta.

Kui trükitavas tabelis on palju ridu, siis võimaldab süsteem tabeli vormistada jätkuvana ühelt trükileheküljelt teisele. Sel juhul tuleb trükikirjelduses peale lehekülgede vahetamise reegli ära näidata ka eelmise lehekülje lõpu ja järgmise lehekülje alguse kujundamise viis. Uue lehekülje algusesse võib sealjuures planeerida tabeli üldandmed kas koos tabeli kogu päisega või ainult koos veergude numeratsiooni näitamisega (nn. lühendatud päis).

Seega tuleb tabelis lisaks päisele ja sisule (s.t. tabeli ridadele) eristada üldiselt veel viit erinevat osa, mida edaspidi tähistame tähtedega A...E:

- A - tabeli pealkiri;
- B - igal trükileheküljel korratavad üldandmed;
- C - eelmise trükilehekülje lõpp;
- D - järgmise trükilehekülje algus;
- E - tabeli lõpp.

Tabelite väljastamisviisi määrav trükikirjeldus sisestatakse omaette kirjena TRYKL vastavalt legendile

LEG TRYKL

1 NIMI T-K

2 RIDA T-V

Esimese taseme elemendi NIMI väärtus määrab trükikirjelduse nime ja vaikimisi ka väljundkirje ning selle legendi nime. Muutuva pikkusega tekstina antud kirjelduse rida võib olla kas väljundtabeli ühe rea kirjeldus, või päise lahtri ja tabeli sisu vastava veeru kirjeldus või hoopis nn. formaadikirjeldus. Trükikirjelduse iga rida koosneb sealjuures kahest osast - indeksist ja kirjeldusest.

Indeks sisaldab kuni kuus punktiga eraldatud komponenti. Sealjuures indeksi esimene komponent näitab, millise kirjeldusega on üldse tegemist. Nimelt täht A...E selle komponendi kohal viitab tabeli vastava nimega osa reakirjeldusele, arv veerukirjeldusele ja täht S sisusse trükitava veergudest sõltumatu rea kirjeldusele. Tähe F korral on tegemist formaadikirjeldusega.

Indeksi teine komponent määrab tähega A...E algavas reakirjelduses reanumbri. Read trükitakse nimelt reanumbri mittekahanevas järjekorras, kusjuures sama reanumber võib ka korduda.

Trükitava tabeli päis ja sisu määratakse veerukirjeldusega. Igas veerukirjelduses antakse kas päise ühte lahtrisse paigutatav tekst, või siis tekst koos selle lahtri alla tabeli sisusse trükitavaid väärtusi andva elemendi nime teatamisega. Veerukirjelduse indeks määrab päise lahtrite järjekorra (vasakult paremale) ja alluvuse. Lahtrititel võib nimelt vajaduse korral olla puu struktuur, kusjuures maksimaalne tasemete arv on kuus. Seost lahtrite paigutuse ja indeksi vahel näitab järgmine joonis:

	:			2		:	:		
	:					:	:		
1	:			2.3		:	3.1	:	4 5
	:	2.1	2.2			:		:	
	:			2.3.1	2.3.2	:		:	
	:					:	:		

Kõrvuti paiknevad erineva tasemete arvuga või erineva ülemustipu juurde kuuluvad veerud (lahtrid) eraldatakse koolonitest moodustatud püstjoontega. Veergusid saab püstjoonega eraldada ka fiktiivse ülemustipu kaudu, nagu seda joonisel on tehtud lahtri 3.1 korral.

Päise ühte lahtrisse mitmele reale paigutatavad tekstid eraldatakse omavahel plussiga. Kirje iga teise või kolmanda taseme eksemplari kohta trükitakse tabeli sisusse üldjuhul üks rida. Andmeid samast eksemplarist võib aga paigutada ka mitmele reale, näiteks esimesele reale koguselised ja teisele reale rahalised näitajad. Erinevatele ridadele paigutatavate elementide nimed eraldatakse niisugusel juhul plussmärgiga, näiteks

KOKKU=KOGUS+SUMMA

Nagu juba öeldud, peavad kõik tabelisse trükitavad andmed sisalduma ühes väljundkirjes: tabeli algusesse ja lõppu paigutatavad ühekordsed andmed peavad sealjuures paiknema selle väljundkirje esimesel tasemel ja tabeli ridadena trükitavad andmed teisel tasemel. Kui aga tabelis on vaja teha mitmetasemelisi vahekokkuvõtteid või trükkida teatud ridade rühma kohta veel eraldi reale mingeid pikemaid tekste, siis moodustatakse väljundkirje kolmetasemelisena. Kirje teise (või kolmanda) taseme iga eksemplari kohta trükitakse tabeli sissusse üldjuhul ikka üks rida, mille reanumbriks võetakse kokkuleppeliselt 5.

Selline kokkulepe on vajalik lisaridade koha määramiseks. Omaette reale tabeli sissusse saab nimelt täiendavat informatsiooni trükkida S-reaga, mis algab indeksiga

S.<taseme nr.>.<reanumber>

kus taseme number näitab, millise väljundkirje taseme eksemplari juurde antud rida kuulub ja reanumber asendit antud tasemele vastava rea suhtes. Kui reanumber selles indeksis on väiksem viiest, siis trükitakse antud S-rida enne veerukirjeldustega määratud rida, kui aga suurem viiest, siis pärast seda rida. Kui taseme numbriks kirjutada 1, siis paigutatakse vastavad S-read (milles võib olla ka esimese taseme elementide väärtusi) ühekordselt vahetult päise järele.

Tavaliselt vormistatakse S-reana pikemad tekstid, mis veergudesse paigutatuna teeksid tabeli liiga laiaks. Samuti aga kasutatakse S-rida näiteks ka sel juhul, kui parema loetavuse huvides on vaja iga rea järele trükkida joon või jätta tühi rida.

Väljundtabelite kirjeldamisel nii ridade kui veergude kaupa on tabelisse kantavateks trükiobjektideks alati kas kirjelduses etteantavad tekstid või väljundkirje elementide väärtused. Väljastatav tekst paigutatakse trükikirjeldusse üldiselt ülakomade vahele. Üksiku sõna korral, mis ei sisalda eraldajat, võib ülakomad sealjuures siiski ka ära jätta. Kui aga kirjelduses antavat teksti alustada ühe asemel kahe ülakomaga (kuid lõpetada ikka ühega), siis paigutatakse see tekst tabelisse sõrendatult.

Kahte ülakoma kasutatakse veel tabeli sisusse spetsiaalsete ridade genereerimiseks. Kui ülakomade vahel on nimelt antud mingi eraldaja (näiteks "'-'), siis trükitakse antud märgist koosnev joon kogu tabeli laiuses. Kui selleks märgiks on tühik, siis jäetakse tühi rida (kuhu tabeli sisus paigutatakse aga automaatselt siiski lahtritevahelisi püstjooni moodustavad koolonid).

Kui ülakomade vahel on antud ühe- või kahekohaline täisarv (näiteks kujul "'9'), siis genereeritakse veerunumbrite loetelu lühendatud päise jaoks alates etteantud arvust (toodud näite korral seega alates 9-st).

Kirje elemendi tunnuseks trükiobjektide loetelus on võrdumärk, millele järgneb väljastatava elemendi nimi. Kui nimi langeb kokku samas kirjelduses vahetult viimati esinenud sõnaga, siis võib nime ära jätta. Elemendi trükipilt on määratud tema kirjeldusega legendis, kuid trükikirjelduses saab seda vajaduse korral ka täpsustada. Üldkujul on elemendikirjeldus järgmine:

=<el.nimi>-<tunnus><laius>

Tunnuse väärtuseks võib siin kirjutada kas ühe tähtedest N,T,K,R, arvu vahemikust 0...9 või mõlemad korraga. Kui elemendi väärtus on null, siis tunnuse N korral trükitakse null, T korral tühik ja nende tunnuste puudumisel miinusmärk. Kui kirje elemendil on tabeli mitmes järjestikuses reas ühesugune väärtus, siis tunnuse K korral trükitakse see element ainult antud väärtuse esmakordsel esinemisel. Tunnuse R korral aga jäetakse elemendi väärtuse kordumisel kogu rida üldse trükkimata.

Arv tunnuse kohal määrab väljastamise mastaabi. Reaalarvude korral näitab mastaap, mitu kohta murdosast tuleb välja trükkida, täisarvude (tüübid N,I,D) korral näitab aga mastaap seda, mitmenda tüvikoha ette lõpust soovitakse paigutada koma.

Laiuse etteandmisega saab muuta elemendi väärtusele tabelis eraldatavat trükipositsioonide arvu. Muutuva pikkusega elemendi korral on laiuse etteandmine kohustuslik. Kui elemendi tegelik väärtus osutub antud laiusest pikemaks, siis jätkatakse selle trükkimist järgmiselt realt samast positsioonist alates. Korduva elemendi korral võib laiuse ette anda kas kogu elemendi jaoks või üksikule komponendile eraldatud trükipositsioonide arvu koos kogulaiusega, näiteks (4,31) määrab komponendi laiuseks 4 ja kogulaiuseks 31 positsiooni. Kui kõiki komponente antud kogulaiuse piires pole võimalik ära paigutada, siis jätkatakse trükkimist järgmiselt realt.

Tabeli osad A...E ja samuti ka S-read kirjeldatakse ridade kaupa. Indeksile järgnevas kirjelduses näidataksegi,

milliseid trükiobjekte tuleb antud reale paigutada. Tabeli laiuse ja ühtlasi ka kirjeldatava rea maksimaalse pikkuse määrab veerukirjeldustest tulenev päise laius. Ühele reale trükitavad objektid koondatakse reakirjelduses komaga eraldatud rühmadesse, kusjuures ühte rühma võib kuuluda üks või mitu trükiobjekti. Rühmad paigutatakse tabeli laiuse piires ühtlaselt, s.t. asendades iga reakirjelduses esineva koma ühepikkuse tühja vahemikuga. Järgneva joonise vasakus osas on näidatud kuus võimalust niisuguste rühmade (R1, R2, R3) kirjelduse esitamiseks ja paremal nende rühmade paigutamine tabeli ühe rea piires (tähelepanu tuleb pöörata komade arvule rühmade vahel või järel):

R1		R1	
R1,R2	R1		R2
R1,R2,R3	R1	R2	R3
R1,	R1		
,R1			R1
R1,R2,,R3,	R1	R2	R3

Teksti ja elemendi vahele tuleb rühmas panna eraldajaks võrdusmärk. Kui aga elemendile järgneb tekst või tekstile jälle tekst, siis kasutatakse eraldajana plussmärki. Näiteks võib ühe rühma kirjelduse esitada kujul:

$$= \langle \text{element} \rangle + \langle \text{tekst} \rangle + \langle \text{tekst} \rangle = \langle \text{element} \rangle = \langle \text{element} \rangle$$

Tabeli D-osa, s.t. järgmise trükilehekülje alguse vormistamiseks on kolm võimalust:

1. Kui D-osa kirjelduses puudub, siis iga uue lehekülje algusesse trükitakse B-osa (kui see on kirjeldatud) ja tabeli päis.

2. Kui D-osa kohta on antud ainult korraldus veergude numereerimiseks, siis trükitakse B-osa ja lühendatud päis (nummerdamine lisatakse ka tabeli kõige esimese lehekülje päisesse, kuigi seda päist määravates veerukirjeldustes otsestelt ei näidata ega saagi näidata).

3. Üldjuhul D-osa moodustatakse antud reakirjelduste järgi, mis muuhulgas võivad sisaldada ka korraldust veergude numereerimiseks.

Formaadikirjelduses antakse mitmesuguste tunnuste ja suuruste väärtused, mida kasutatakse tabeli üldpildi kujundamisel. Formaadikirjelduse indeks on F.1, millele järgneb parameetrite loetelu kujul

F.1 <par.nimi>=<väärtus>,<par.nimi>=<väärtus>,...

Parameetrite TA ja TL väärtustega näidatakse, mitu tühja rida tuleb jätta iga tabeli algusesse ja lõppu. Parameetri LK väärtusega saab anda korralduse väljundtabeli lehekülgedena vormistamiseks järgmiselt:

LK=0 - lehekülgi ei vormistata, s.t. tabel väljastatakse ühes tükis nii pikana, kui palju vastavas väljundkirjes andmeid leidub;

LK=1 - tabeli trükkimist alustatakse uuel leheküljelt, kuid tabelit ennast lehekülgedeks ei jagata;

LK=2 - tabel jagatakse lehekülgedeks, kusjuures üleminek järgmisele leheküljele toimub koos uue teise taseme eksemplari võtmisega;

LK=3 - tabel jagatakse lehekülgedeks, kusjuures lehekülgi vahetatakse koos iga uue kolmanda taseme eksemplari vaatluselevõtmisega.

Üleminekul ühelt leheküljelt teisele kasutatakse formaadikirjelduses veel parameetrit LP. Kui parameetri LK väärtuseks on määratud kas 2 või 3, siis vastava taseme iga uue eksemplari vaatlusele võtmisel kontrollitakse ühtlasi, mitu rida on veel jäänud lehekülje lõpuni. Kui ridu on jäänud mitte rohkem kui parameetri LP väärtusega määratud, siis katkestatakse tabeli sisu trükkimine ja minnakse kohe üle lehekülje lõpu vormistamisele.

Tabeli lõppu võib lasta trükkida lahendamise kuupäeva, andes ette KP=1. Sama tabeli trükkimist saab korrata, teades eksemplaride arvu parameetriga EA. Kui tabeli laius on alla 64 trükipositsiooni, siis EL=1 korral trükitakse kaks sama tabelit kõrvuti.

Vaatleme nüüd konkreetse näitena niisugust olukorda, mil on vaja välja trükkida lao sissetulekute aruanne hankijate ja materjali liikide lõikes. Kõik väljastamiseks vajalikud andmed olgu kogutud väljundkirjesse SSORT, mis olgu kirjeldatud legendiga

LEG SSORT

1 LADU X4-K

VKUUP X6 VIIMANE KUUPÄEV

2 HANK X4-K HANKIJA KOOD

HNIM T20 HANKIJA NIMETUS

3 ART T10-K MATERJALI KOOD

S1 N3.2 ESIMESE

S2 N3.2 TEISE JA

S3 N3.2 KOLMANDA SORDI MATERJAL

KCKKU N4.2

Et ühes laos säilitatakse üldiselt palju materjale, siis on loomulik kavandada tabel jätkuvana ühelt leheküljelt teisele. Väljastatavate tabelite üldkuju soovime saada sellisena, nagu järgmisel joonisel näidatud:

MATERJALIDE SISSETULEK HANKIJATELT						A	
SORTIDE JARGI							
LADU	12	SEISUGA 860813				B	
K O G U S							
HAN- KIJA	MATERJALI ARTIKKEL	:	11.	2.	3.	: KOKKU	pääs
1	2	:	3	4	5	: 6	
TEHAS "TERASTRAAT"						sisu	
303 T473	:	-	148,60	12,40	: 161,00		
8500	:	300,00	-	-	: 300,00		
JARGNEB						C	

-2-							
LADU	12	SEISUGA 860813				D	
1	2	:	3	4	5		: 6
VABRIK "VIISNURK"						sisu	
3401 M68104	:	137,12	-	12,30	: 149,42		
927-63	:	250,00	20,00	-	: 270,00		
06/AUG/1986						E	

Sellel joonisel on tabeli parempoolses servas joone taga antud veel väljastamisele mitte kuuluv informatsioon - tabeli erinevate osade nimetused ja tähistused.

Niisuguse tabeli saamiseks tuleb trükikirjeldus esitada näiteks järgmisel kujul:

TRYKL SSORT

A.1 'MATERJALIDE SISSETULEK HANKIJATELT'

A.2 'SORTIDE JARGI'

A.3 ' ' '

B.1 LADU=,SEISUGA=VKUUP

1 'HAN-' + KIJA = HANK

2 MATERJALI + ARTIKKEL = ART

3 ' ' KOGUS'

3.1.1 '1.' = S1

3.1.2 '2.' = S2

3.1.3 '3.' = S3

3.2 KOKKU =

S.2.1 = HNIM,

C.1 ' ' -'

C.2 , JARGNEB

D.1 ' ' 1'

E.1 ' ' -'

F.1 LK=3, KP=1

Trükikirjelduse transleerimiseks, s.t. kirjest TRYKL te-
geliku trükikirjelduse moodustamiseks tuleb tellimuses esi-
tada rida (meie näite korral on kirjelduse nimi SSORT):

/TK T=<kirjelduse nimi>

Tabeli maksimaalseks laiuseks on 128 trükipoitsiooni.
Laiemaid tabelleid tuleb trükkida kahes osas, s.t. koostada
kummagi osa jaoks eraldi trükikirjeldus. Teise osa kirjeldu-
se nimi erineb niisugusel juhul legendi nimest ja seepärast
tuleb selle transleerimiseks anda tellimuses veel parameeter
LN=<legendi nimi>.

8. STAMP-PROGRAMMID

Vaadeldava süsteemi põhiliseks andmete töötlemise vahendiks on erilises nn. STAMP-keeles kirjutatud programmid ehk STAMP-programmid. Üldjuhul kujutab niisugune programm endast nummerdatud (märgendiga varustatud) eeskirjade ehk lausete järjendit, mille alusel teatud hulgast lähtekirjetest moodustatakse tulemuskirjed. Neid tulemuskirjeid võib kas kasutada samas programmis järgmiste kirjete moodustamiseks, või trükkida kirjetesse koondatud andmed väljundtabelitena või siis salvestada hoopis sisendfaili (lähteandmeteks järgmisele programmile või edaspidiseks ülekandmiseks püsifaili).

STAMP-programmi tekst sisestatakse kirjena TEKST, mille struktuur on määratud järgmise legendiga:

LEG TEKST

1 NIMI T-K

2 MARGEND N4-K

LAUSE T-V

Programm koosneb kahest osast - kirjeldavad laused ja töölaused. Kirjeldav lause määrab vajalike kirjete legendid, kusjuures kirje kasutamiskiisist sõltuvalt saab legende määrata ühega neljast võtmesõnast: LEGK, LEGL, LEG või LEGT.

Võtmesõna LEGK tähendab, et programmis kasutatakse varem sisestatud legendi. Ühe võtmesõnaga võib määrata ka mitu sellist legendi, näiteks

LEGK)KIRJE1,KIRJE2

Võtmesõna LEGL korral kasutatakse samuti varem sisestatud legendi, kuid programmis vajatakse arvutusteks täiend-

vaid tööelemente. Nende tööelementide kirjeldused lisatakse lause LEGL järele, näiteks kirjeldava lausega

LEGL)KIRJE3

2 KOKKU N6.2

3 NIMETUS T-V

lisatakse nimetatud tööelementide kirjeldused programmi transleerimise ajaks legendi KIRJE3 vastavalt teise ja kolmanda taseme viimasteks elemendikirjeldusteks.

Võtmesõna LEG tähendab, et antud nimega kirjeid moodustatakse ainult selle STAMP-programmi vahendusel. Võtmesõna järel antakse legendi täielik tekst. Kui programm transleeritakse tõrgeteta, siis salvestatakse sisendfaili lisaks transleeritud programmile ka vastav kirje LEGEND, mida kasutatakse kas trükikirjelduse transleerimisel või mingi teise programmi lähteandmete kirjeldusena.

Võtmesõna LEGT korral sisaldub legendi täielik tekst samuti programmis, kuid tegemist on nüüd töökirjega, s.t. seda kirjet ei salvestata ega trükita ja loomulikult ei salvestata ka kirje legendi.

Kui kirje nime tuleb töölausetes kasutada palju kordi, siis võib pika nime lause DEF abil asendada ühe- või kahekojalise lühikese nimega. Lause DEF kirjutatakse pärast legendide määramist, kuid enne esimest lühikese nime kasutamist. Näiteks pärast lauset

DEF)KIRJE1=K1,KIRJE2=K2,KIRJE3=K

võib vastavas programmis täielike nimede KIRJE1, KIRJE2 ja KIRJE3 asemel kasutada nende kirjete jaoks vastavalt nimesid K1, K2 ja K.

STAMP-programmi kirjeldava osa võib sisestamiskeeles esitada näiteks järgmiselt:

```
//L TEKST PR1
/10 LEGK)KIRJE1,KIRJE2
/20 LEGL)KIRJE3
/30 2 KOKKU N6.2
/40 N N4
/50 3 NIMETUS T-V
/60 LEGT)T
/70 2 NR N4
/80 SUMMA N4.2
/90 DEF)KIRJE1=K1,KIRJE2=K2,KIRJE3=K
```

(võimalike hilisemate täienduste lisamiseks on laused siin märgendatud sammuga 10).

Töölaused kujutavad endast korraldusi mingiks tegevuseks. Kõige üldisemal kujul on töölause struktuur järgmine:

```
KOOD.MOD)OP,....,OP=OP,....,OP*M,....,M
```

Operatsiooni kood on kuni 5-täheline sõna, millele teatud juhtudel lisatav modifikatsioon (MOD) täpsustab operatsiooniga tehtavat tööd (modifikatsioon esitatakse ühe- või kahekohalise 16-ndarvuna). Vasakul pool võrdusmärgi olevaid operande (OP) nimetame edaspidi tulemusoperandideks ehk tulemusteks, paremal pool võrdusmärgi olevaid operande aga lähteoperandideks ehk argumentideks. Märgendeid (M) kasutatakse lausete täitmise järjekorra muutmiseks või tingimuste kehtivuspiirkonna määramiseks. Töölause kohustuslik komponent on ainult operatsiooni kood. Sõltuvalt operatsioonist võivad muud komponendid esineda osaliselt või üldse puududa.

Operandiks võib olla kas kirje nimi, kirje elemendi nimi või konstant (muidugi vaid argumendi kohal). Kirje elementi tähistatakse sealjuures konstruktsiooniga

<kirje nimi>.<elemendi nimi>

Kui mitu järjestikust operandi (elemendi) pärinevad samast kirjest, siis pole kirje nime vaja korrata, näiteks võib operandide loetelus kirjutada

K.KOKKU,NIMETUS või T.NR,SUMMA

Konstandi või võrdusmärgi esinemine operandide loetelus ei muuda sealjuures kirje nime. Seega loetakse lubatavaks ka näiteks kirjutised

K.KESKMINE,2,NIMETUS või T.NR=SUMMA

Operandiks oleva konstandi võimalikud kirjutusviisid on sõltuvalt arvutuübist esitatud järgmiste näidetega:

N: 2, 1937

I: 4, -763

R: 14.7, -21., 0.071

D: -361D, 861209D

X: 666X, OABX

T: 'TEKST', 'PUUDUB KIRJE'

Arvu täis- ja murdosa näidatakse eraldi ainult reaalarvu korral. Muud arvud kirjutatakse alati täisarvudena. Näiteks kui on vaja kontrollida, kas naturaalarvuna N4.2 kirjeldatud elemendi väärtus on suurem kui 100, siis vastav võrdluskonstant tuleb programmis kirjutada kujul 10000. Arvkonstante eristatakse nimedest selle järgi, et konstandid algavad numbriga. Kui aga 16-ndkonstant algab tähega A...F, siis tuleb sellise arvu ette kirjutada null.

Sissejuhatavaks tutvumiseks STAMP-keele operatsioonidega toetume lk. 15 toodud näitelegendile KLASS, milles keskmine hinne KH oli kirjeldatud pseudoelemendina. Vaatleme järgnevalt, kuidas STAMP-programmiga saab iga õpilase jaoks arvutada tema keskmise hinde. Vastava programmi tekst on järgnevalt antud sellisel kujul, nagu selle trüüb translaator:

TEKST KHINNE

```
10  LEGL)KLASS
20  2 HARV N2 HINNETE ARV
30  SUMMA N3
40  DEF)KLASS=K
50  LUG)K*110
60  KIND.C)K.HARV=HINNE
70  KIND.E)K.SUMMA=HINNE
80  JAG.2)K.KH=SUMMA,HARV
90  SALV)K
100 M)*50
110 STOP)
```

Selle programmi lausetega 10...4C reserveeritakse kirje KLASS teisel tasemel lisaväljad kahe tööelemendi jaoks ning teatatakse, et järgnevalt võib kirje nime KLASS asemel kasutada lühemat nime K.

Operatsiooniga LUG loetakse sisemällu järjekordne kirje KLASS, kusjuures enne uue kirje lugemist kustutatakse vajaduse korral mälust eelmine antud nimega kirje. Kui kõik kirjed on juba läbi vaadatud, siis antakse juhtimine märgendiga 110 varustatud töölauselale, kus operatsioon STOP lõpetab programmi täitmise.

Operatsioon KIND (= kanda indeksi järgi) eeldab, et esimeseks argumendiks on korduv element ja tulemuseks sama tüüpi mittekorduv element. Põhivariandi (modifikatsioon puudub) korral kantakse tulemuseks teise argumendiga (indeksiga) määratud komponent.

Operatsiooni KIND modifikatsioonide C ja E korral vaadeldakse korduva elemendi kõiki komponente, kusjuures esimesel juhul (C) leitakse nullist erinevate komponentide arv ja teisel juhul (E) nende komponentide summa. Et vaadeldavas näites on tulemusoperand HARV või SUMMA kirje teise taseme ja argument HINNE kolmanda taseme element, siis tulemuse leidmiseks vaadatakse teise taseme iga eksemplari jaoks läbi kõik talle alluvad kolmanda taseme eksemplarid. Seega kokkuvõttes leitakse toodud programmi lausetega 60 ja 70 iga õpilase hinnete arv ja summa.

Keskmise hinde leidmiseks on vaja saadud hinnete summa jagada hinnete arvuga. Et keskmine hinne KH tuleb leida täpsusega 2 kohta peale koma (legendis N1.2), siis kasutame jagamise modifikatsiooni 2, mille korral esimene argument korrutatakse eelnevalt kümne vastava astmega (s.t. arvuga 100) ja seejärel jagatakse teise argumendiga. Jagatis ümardatakse ning kirjutatakse elemendi KH väärtuseks. Lausega SALV)K salvestatakse kirje KLASS sisendfaili ja järgmise operatsiooniga M (= minna) minnakse tagasi töölausesse 50 lugema uut kirjet KLASS.

Programmi KHINNE täitmise tulemusena on sisendfaili ümber kirjutatud kõik kirjed KLASS, mis erinevad lähtekirjetest selle poolest, et neis kõigis on väärtuse omandanud ka

element KH. Märgatava ruumi kokkuhoiu saavutame aga siis, kui sisendfaili laseme kirjutada ainult korraldused elemendi KH väärtustamiseks. Selleks tuleb programmi senine lause 90 asendada lausetega

90 FOP)'A2','KLASS',K.NR

95 FPR)K.PNIMI,ENIMI,'KH',KH

Operatsioonidega FOP (= formeerida operatsioon) ja FPR (= formeerida paranduste rida) genereeritav paranduskirje sisaldab täpselt samasuguseid andmeid, nagu peab sisaldama samasisuline sisestamiskeelne kirjegi: kahe kaldkriipsu järele kirjutatavad andmed esitatakse nüüd operatsiooni FOP operandidena ning ühe kaldkriipsu järele kirjutatavad operatsiooni FPR operandidena. Lausetele 90 ja 95 vastav sisestamiskeelne paranduskirje tuleb antud juhul näiteks järgmise struktuuriga

//A2 KLASS 3A

/PAJU PILLE KH 5,00

/SUSI SULEV KH 3,51

/.....

Niiviisi saadud paranduskirjetes sisalduvate keskmiste hinnete salvestamiseks kirjetesse KLASS tuleb käivitada programm OUT.

Vaadeldud programmi KHINNE võib vajaduse korral modifitseerida ka selliselt, et me elementide KH arvutatud väärtusi järgnevas kasutamiseks kirjetesse KLASS üldse ei paigutagi, vaid laseme klasside kaupa trükkida õpilaste nimekirja koos keskmise hinde näitamisega. Selleks tuleb programmi lülitada laused

45 KTR)'QPILASTE KESKMISED HINDED'

55 KTR)10,'KLASS',K.NR

90 KTR)K.PNIMI,ENIMI,KH

Ühekordselt trükitakse paberi vasakusse serva lauses 45 näidatud tekst. Edasi trükitakse iga klassi kohta alates 10-ndast positsioonist sõna KLASS ja klassi number ning selle klassi iga õpilase kohta järgnevalt üks rida tema nime ja keskmise hindega.

Vaadeldud lihtsas STAMP-programmis oli tegemist ainult ühe kirjega, kus erineva taseme eksemplarid on seotud alluvussuhtega. Mitme kirje korral aga tekib vajadus luua seoseid ka erinevate kirjete vahel.

Näiteks programmis PR1 (vt. lk. 63) kirjeldati elemendid K.KOKKU ja T.SUMMA mõlemad teisel tasemel. Kui mingis töölauses nüüd operandiks on ainult T.SUMMA, siis mõeldakse sellega elementi SUMMA kirje T kõigis teise taseme eksemplarides. Operandide kombinatsiooni

$K.KOKKU=T.SUMMA$

korral aga vaadatakse läbi kirje T kõik teise taseme eksemplarid kirje K iga teise taseme eksemplari jaoks. Näiteks kui operatsiooniks oleks summeerimine, siis saaksime elemendile KOKKU kõigis eksemplarides sama väärtuse.

Olgu meil nüüd elemendi K.KOKKU väärtuse arvutamiseks vaja kokku liita elemendid SUMMA ainult kirje T nendest eksemplaridest, kus elemendi T.NR väärtus ühtib elemendi N väärtusega kirje K vaadeldavas eksemplaris. Selleks on kaks põhimõttelist võimalust, millest esimese võib realiseerida programmiõiguga

210 FIX)K*240,250

220 TVD)T.NR,K.N*240,210

230 S)K.KOKKU=T.SUMMA

240 M)*210

250

Operatsiooniga FIX fikseeritakse igal pöördumisel järjekordne teise taseme eksemplar kirjes K. Edasi kuni märgendini 240 vaadeldaksegi kirje K kõiki teise taseme elemente ainult sellest fikseeritud eksemplarist pärinevatena. Tingimusoperatsiooniga TVD (= võrdne) tähistatakse kirje T need teise taseme eksemplariid, kus elemendi NR väärtus langeb kokku elemendi K.N väärtusega fikseeritud eksemplarist. Tingimuse mõjupiirkonnas kuni märgendini 240 vaadeldakse ainult tähistatud eksemplare (kui selliseid ei leidu, siis antakse juhtimine tagasi märgendile 210).

Operatsiooniga S summeeritakse nüüd elemendid SUMMA üksnes tingimusega tähistatud eksemplaridest ning omistatakse tulemus elemendi K.KOKKU väärtuseks fikseeritud eksemplaris. Tsükliit korratakse nii kaua, kuni kirje K kõik eksemplariid on läbi vaadatud ning operatsioon FIX suunab programmi täitmise märgendile 250.

Kirjeldatud lõigu võib aga programmis asendada ka üheainsa lausega kujul

S)K(N)KOKKU=T(NR)SUMMA

Selle operatsiooni täitmine toimub kahes etapis. Esiteks moodustatakse kirje K iga eksemplari jaoks viitade ahel kirje T nendesse eksemplaridesse, kus elemendi T.NR väärtus on sama, mis elemendil N kirje K vaadeldavas eksemplaris

(viidad moodustatakse ajutise tabeli abil mõlema kirje ühekordse läbivaatamise teel). Teisel etapil võetakse argumente summeerimiseks ainult viidatud eksemplaridest.

Sulgudes olevaid elemente N ja NR nimetatakse vastavalt vasaku ja parema poole viidaelementideks. Viidaelemente võib olla ka mitu, kuid vasakus ja paremas pooles peab viidaelemente olema võrdne arv. Oluline on veel arvestada, et viidaelementid ei pea tingimata olema tulemuse või argumentidega samalt tasemelt.

Tulemusoperandide ja argumentide valikul tuleb silmas pidada järgmisi reegleid. Mitme tulemusoperandi korral peavad need kõik olema sama kirje elementid. Argumentideks võivad olla konstandid, tulemuskirje elementid või mingi teise kirje elementid. Argumente liigitatakse veel ühesteks ja korduvateks. Argument on ühene siis, kui igale tulemusoperandile vastab ainult üks argumenti väärtus. Üheseks argumentiks on:

- 1) konstant;
- 2) esimese taseme element;
- 3) teise taseme element operatsiooni FIX mõjupiirkonnas;
- 4) tulemuskirje element, mille tasemenumber on võrdne või väiksem tulemusoperandi tasemenumbriks;
- 5) viidaga üheselt määratud element, s.t. parema poole viidaelementide hulgas on loetletud kõik eksemplari määravad võtmeelementid.

Muudel juhtudel on tegemist korduva argumentiga. Kõik korduvad argumentid peavad olema ühest kirjest, seejuures kas kõik ilma viidata või määratud sama viidaga.

STAMP-programmi lauseid võib ka kommenteerida. Kommen-
taar eraldatakse muust lauseosast tühikuga. Kui aga lause
lõpeb sulgeva suluga, siis peab kommentaar algama avava su-
luga. Pikema kommentaari võib kirjutada ka eraldi lausena.
Sel korral paigutatakse kommentaar sulgudesse, kuid lõpetava
sulu võib ka ära jätta.

Lõpuks tutvustame veel mõnd STAMP-programmide kirjutami-
sel kasutatavat lihtsustusreeglit.

Olgu A3, A4, A5, A6 legendis K järjestikku kirjeldatud
sama taseme elemendid. Siis operandide loetelu

K.A3,A4,A5,A6

voib asendada samaväärse lühema konstruktsiooniga

K.A3-A6

Kui vasaku poole viidaelementide nimed langevad kokku
parema poole viidaelementide nimedega, siis võib viimased
nimetamata jätta (säilitades aga sulud). Sama tähendusega on
näiteks konstruktsioonid

$T(A,B,C)D=L(A,B,C)E$ ja $T(A,B,C)D=L()E$

Kui mitmes järjestikuses lauses langevad kokku operat-
siooni kood, kirjete nimed ja operandide arv, siis võib need
laused ühendada üheks lauseks, andes muutuvad osad plussiga
eraldatud loeteluna. Näiteks laused

$LAH)T(A,1)B=C,L(D,E)F$

$LAH)T(A,2)B=G,L(D,E)H$

voib asendada ühe samaväärse lausega

$LAH)T(A,1+2)B=C+G,L(D,E)F+H$

Siin operatsiooniga LAH (= lahutada) kantakse tulemuseks
esimese ja teise argumendi vahe.

9. STAMP-KEELE OPERATSIOONID

Nagu eelmises paragrahvis juba nimetatud, on mistahes STAMP-programmi sisuks üldjuhul tulemuskirjete moodustamine mingite lähtekirjete järgi. Võtamegi nüüd vaatlusele tulemuskirjete moodustamiseks kasutatavad operatsioonid.

Käesoleva väljaande maht ei võimalda tutvustada kõiki STAMP-keele operatsioone (neid on üle saja). Seepärast kirjeldame järgnevalt neist ainult kõige olulisemaid ja sagedamini kasutatavaid. Eelmises paragrahvis toodud näiteprogrammi selgitamisel kirjeldatud operatsioonid (FIX, FOP, FPR, M, KIND, KTR) me siin enam lähemalt ei vaatle.

Uue kirje moodustamist sisemällu tuleb süsteemis STAMP alustada kirje avamisest. Avamine tähendab, et paigaldatakse kirje päis ja reserveeritakse mälu esimese taseme eksemplari jaoks. Sealjuures erineb paigaldatav päis legendis kirjeldatust, sest STAMP-programmis on ette nähtud täiendavad lisaväljad tööelementidele, kirjetevahelistele viitadele ja tingimuste rahuldatuse tähistamiseks.

Enamasti genereerib juba translaator avamisoperatsiooni, mis paigutatakse programmis selle lause juurde, kus antud kirjet esmakordselt kasutatakse. Seepärast piisab kirje avamiseks näiteks lausest, kus toimub mingi esimese taseme elemendi väärtustamine. Kui aga kirjel puuduvad esimese taseme elemendid ja teise taseme eksemplaride formeerimine teostatakse tsükliiliselt, siis tuleb vastava tsükli ette kirjutada avamislause, näiteks kujul

AVADA)K

Üheningelisi kirjeid võib sisemälus korraga olla ainult üks. Seepärast sisaldab avamise operatsioon endas ka eelmise kirje kustutamise.

Lähtekirjed tuuakse sisemällu operatsiooniga LUG (vt. ka lk. 65), milles konkreetse kirje lugemiseks tuleb ette anda kirje kõigi võtmelementide väärtused. Näiteks lauses

LUG)K.V1,V2,V3=L.A,B,C*510

peavad vasakpoolsed operandid (võtmelementide nimed) olema esitatud legendi järjekorras, nende väärtusi näitavad elementid A, B ja C peavad aga olema kirjes L üheselt määratud. Kui selliste võtmeväärtustega kirje fondis puudub, siis järgmisena täidetakse lause märgendiga 510.

Operatsiooniga LUG saab organiseerida ka tsükli üle näidatud nimega kirjete. Selleks antakse võtmeväärtused ette kas osaliselt või jäetakse hoopis andmata, näiteks kujul

LUG)K*510

LUG)K.V3=L.C*510

LUG)K.V1=L.A*430

Esimese kahe näite korral vaadatakse läbi kogu fail, kusjuures esimesel juhul loetakse järjest sisse kõik kirjed, teisel juhul vaid need kirjed, milles kolmanda võtmelementi väärtus ühtib elemendi L.C väärtusega. Kolmandal juhul vaadatakse vajalike kirjete otsimiseks läbi vaid osa failist, sest kirjed paiknevad seal võtmete kasvavas järjekorras ning otsida tuleb esimese võtmelementi etteantud väärtuse järgi. Suunamine märgendi järgi toimub kõigil juhtudel siis, kui faili läbivaatamisel selgub, et nõutavaid kirjeid failis enam ei leidu.

Kirjest loetakse üldiselt sisse ainult need elemendid, mida programmis on nimetatud. Kui loetava kirje töötlemine lõpeb aga väljastusoperatsiooniga (SALV, VTR), siis sisestatakse kõik elemendid.

Sageli on vaja eristada sisendfaili kirjeid (antud seansis sisestatud) nendest püsifailide kirjetest, mis on varem põhifaili või kollektorisse kantud. Sisendfailist lugemiseks tuleb operatsiooni LUG modifikatsiooniks kirjutada 70, püsifailide jaoks aga 80.

Kirjete lugemisel teeb süsteem ranget vahet konkreetse kirje lugemise (operatsioonis LUG on näidatud kõigi võtmelementide väärtused) ja järjest lugemise vahel (võtmelementide väärtused näitamata või näidatud osaliselt). Järjest lugemisel fikseeritakse pärast iga järjekordse kirje lugemist koht failis ning uut kirjet otsitakse alates sellest kohast. Konkreetse kirje lugemisel otsitakse seda kirjet samuti failis fikseeritud kohast alates. Kui kirjet aga ei leita ning puhvrisse toodud ploki andmete põhjal otsustades võib kirje asuda failis eespool, siis jätkatakse kirje otsimist faili algusest peale.

Olgu näiteks vaja üheaegselt vaadelda mingis failis asuvaid ühiste võtmelementidega (A ja B) kirjeid M ja N, kusjuures igale kirjele M ei tarvitse kirjet N vastata. Et vältida kirje N lugemisel pöördumist faili algusesse, tuleb konkreetse kirje lugemine muuta järjest lugemiseks, kasutades operatsiooni LUG modifikatsiooni 1:

LUG)M*500

LUG.1)N.A,B=M.A,B*430

Kõige lihtsamalt saab elemente väärtustada operatsiooniga K (= kanda) vastava väärtuse ülekandmise teel kas samast või mingist teisest kirjest või siis omistades väärtuseks hoopis konstandi. Ühe operatsiooniga saab väärtustada mitu elementi, kusjuures nii tulemusoperandid kui ka argumendid võivad olla erinevatelt tasemetelt, näiteks

$$K)K.A,B,C=D,L.B,1$$

Seejuures aga peab üldiselt jälgima, et argumendid oleksid üheselt määratud (kui tegemist on korduva argumendiga, siis võetakse neist kandmisel arvesse vaid viimane väärtus). Antud näites peab element $K.D$ olema elemendiga $K.A$ samal või kõrgemal tasemel, element $L.B$ aga omama ühese väärtuse, s.t. olema esimese taseme element või teise taseme element operatsiooni FIX mõjupiirkonnas.

Kui argumendiks on teise või kolmanda taseme element mingist teisest kirjest, siis peame tulemus- ja lähteeksemplarid siduma viitadega (vt. lk. 69), näiteks

$$K)K(E)A,B=L(E)A,4$$

Et argument oleks üheselt määratud, peab element $L.E$ olema vaadeldava taseme ainus võtmelement. Võib aga juhtuda, et mingile eksemplarile kirjes K ei leidu kirjes L vastavat eksemplari. Niisugusel juhul jääb selles eksemplaris väärtustamata ka element $K.B$, järelikult oleks parem elemendid A ja B väärtustada omaette operatsioonidega.

Kui tulemuse ja argumendi arvutüübid on erinevad, siis genereerib translaator tüüpide N , I , D ja R vahel ühtlasi vajaliku teisenduse. Kui aga üks kahest, kas tulemus- või lähteoperand on X -tüüpi, siis teisendust ei määrata.

Teksti saab kanda ainult T-tüüpi elemendi väärtuseks. Sealjuures pikema teksti kandmisel lühemale väljale jäetakse lõpp kandmata, pikemale väljale kandmise korral aga täiendatakse teksti lõpust tühikutega. Analoogiliselt toimub muutuva pikkusega teksti kandmine fikseeritud pikkusega teksti kohale ja vastupidi.

Kirje teise ja kolmanda taseme eksemplarid moodustatakse operatsiooniga FE (= formeerida eksemplarid). Vasakul pool võrdusmärgi loetletakse selle operatsiooni tulemusoperandi- dena elemendid, millele lisatavas eksemplaris omistatakse väärtused, argumentide loetelus aga näidatakse, kust elementide vastavaid väärtusi võtta. Tulemusoperandide hulgas peavad sealjuures tingimata esinema kõik võtmeelemendid, muud elemendid võivad olla antud osaliselt või üldse puududa. Argumentide kohal tohib esitada kas mingi teise kirje suvalise taseme elemente, konstante või siis tulemuskirje esimese taseme elemente.

Ühe operatsiooniga FE saab kirjesse lisada kas ainult teise taseme eksemplare või nii teise kui ka kolmanda taseme eksemplare korraga. Kui teise taseme eksemplarid on olemas, siis saab lisada ka ainult kolmanda taseme eksemplare. Seejuures tuleb teise taseme eksemplari määramiseks ära näidata vastava eksemplari võtmeelementide väärtused.

Olgu näiteks V2 ja V3 vastavalt kirje K teise ja kolmanda taseme võtmeelementid ning A2 ja A3 lihtelemendid. Sellisel juhul operatsioonidega

FE)K.V2,A2=L.B,C

FE)K.V3,A3=L.B,D,E

moodustatakse kirjesse K kõigepealt teise taseme eksemplarid ja seejärel lisatakse vastavad kolmanda taseme eksemplarid. Kokkuvõttes sama tulemuse oleksime võinud saada ka üheainsa lausega

FE)K.V2,A2,V3,A3=L.B,C,D,E

Argumentide iga uue kombinatsiooni korral kontrollitakse, kas selliste võtmeväärtustega eksemplar tulemuskirjes juba leidub ning eitaval juhul see lisatakse. Kui aga formeeritaval tasemel võtmeelemendid legendis üldse puuduvad, siis argumentide iga kombinatsiooni korral lisatakse üks eksemplar. Kui argumendid on arvulised ja nende tüübid ei ühti tulemuskirje legendis antuga, siis sooritatakse eelnevalt ka vajalikud tüübiteisendused.

Operatsioonil FE on veel kolm modifikatsiooni: E, F ja C. Modifikatsioone E ja F kasutatakse korduva elemendi komponentide lahtipakkimiseks erinevatesse eksemplaridesse, kusjuures argumentide loetelus võib olla ainult üks korduv element.

Modifikatsiooni E korral genereeritakse eksemplar ainult sellisel juhul, kui järjekordne komponent üldse esineb, s.t. tema väärtus erineb nullist. Järgnevas näites eeldatakse, et L.C tähendab korduvat elementi, K.V3 aga sama tüüpi ühekordset elementi:

FE.E)K.V2,A2,V3=L.A,B,C

Modifikatsioon F on sisuliselt sama tähendusega, kuid tema korral arvestatakse ka komponente väärtusega null. Täpsemalt niisugusel juhul tulemuskirjes korduva elemendi komponendile vastavas eksemplaris võtmeelemente ei määrata.

Modifikatsiooni C kasutatakse etteantud arvu eksemplari-
de genereerimiseks kirjesse. Nii näiteks operatsiooniga

FE.C)K.V2=1,10

moodustatakse kirjesse K kümme teise taseme eksemplari võt-
meväärtustega 1...10.

Nelja aritmeetilist tehet sooritavad operatsioonid S
(= summeerida), LAH (= lahutada), KOR (= korrutada) ja JAG
(= jagada) on realiseeritud kõigi arvutüüpide jaoks. Tehtest
osa võtavad argumendid viiakse vajaduse korral automaatselt
ühisesse arvusüsteemi. Tehtes kasutatava arvusüsteemi vali-
kul peetakse silmas teisendamise suunda $D \rightarrow I(N) \rightarrow R$, kus-
juures 16-ndarvud loetakse alati õiges arvusüsteemis ole-
vaiks. Kui tulemusoperandi esitusviis vastavas legendis eri-
neb tehtes kasutatud arvusüsteemist, siis enne salvestamist
tulemus teisendatakse.

Summeerimisoperatsioonis võib argumente olla kas üks või
mitu. Argumentide iga kombinatsiooni korral liidetakse kõigi
argumentide väärtused. Kui tulemusele vastab üks argumentide
kombinatsioon, siis saadud summa salvestatakse. Kui aga te-
gemist on korduvate argumentidega, siis saadud osasummad
eelnevalt liidetakse.

Teistes aritmeetilistes tehetes peab argumente alati
olema täpselt kaks. Lahutamisoperatsioonis lahutatakse teine
argument esimesest ja jagamisoperatsioonis jagatakse esimene
argument teiseaga. Korduvate argumentide korral osakorrutised
liidetakse samuti nagu summeerimiselgi. Lahutamis- ja jaga-
misoperatsioonis määrab korduvate argumentide korral tulemu-
se viimane argumendiväärtuste paar.

Sõltumata legendis antud mastaabist tõlgendatakse D, N ja I-tüüpi arve aritmeetilistes tehetes täisarvudena. Seepärast tuleb liitmisel ja lahutamisel ühte argumenti vajaduse korral eelnevalt korrutada või jagada kümne sobiva astmega. Tehetes KOR ja JAG saab seda teha sama operatsiooniga, kirjutades vastava astendaja modifikatsiooninumbriks. Nulliga jagamisel eeldatakse, et null on väärtuse puudumise tunnus ja tulemuseks kirjutatakse samuti null.

Summeerimisoperatsioonideks on veel KSL (= liita summa) ja LM (= liita tulemustele). Operatsioon KSL erineb operatsioonist S ainult selle poolest, et leitud summa liidetakse juurde tulemusoperandi senisele väärtusele. Operatsioonis LM, mis on realiseeritud ainult N või I-tüüpi arvude jaoks, vastab igale argumendile oma tulemusoperand. Kõiki argumente summeeritakse eraldi ja saadud summad liidetakse vastavatele tulemusoperandidele. Näiteks operatsioonidega

$$\text{KSL) } K.S = L.A, B, C$$

$$\text{LM) } K.A, B, C = L.A, B, C$$

liidetakse elemendi K.S senisele väärtusele juurde elementide L.A, L.B ja L.C summad üle argumentide kõigi kombinatsioonide, elementide K.A, K.B ja K.C seniseid väärtusi aga suurendatakse (igäht eraldi) vastavalt elementide L.A, L.B ja L.C summa võrra.

Põhilisteks tingimusoperatsioonideks on TVD (= võrdne), TMV (= mittevõrdne), TS (= suurem) ja TSV (= suurem-võrdne). Kõigis nendes kasutatakse üldiselt kaht argumenti, kusjuures tingimus loetakse rahuldatuks siis, kui näiteks TS korral esimese argumenti väärtus on suurem teise väärtusest.

Tingimusoperatsiooni täitmise käik sõltub sellest, kas argumentide seas leidub korduvaid argumente või mitte. Ühese väärtusega argumentide korral toimub programmis hargnemine: kui tingimus on rahuldatud, siis jätkatakse programmi täitmist tingimusele järgnevast operatsioonist, vastasel korral aga suunatakse märgendi järgi.

Ka korduvate argumentide korral võib programmis toimuda hargnemine, kuid tingimusoperatsiooni põhiülesandeks on sel juhul siiski nende eksemplaride tähistamine, mille jaoks tingimus osutub rahuldatuks. Nimelt reserveerib translaator kirje teisel või kolmandal tasemel ühe baidi, kuhu tingimuse rahuldatuse puhul kirjutatakse vastava biti väärtuseks üks ja mitterahuldatuse korral null. Kui argumendid on sama kirje erinevalt tasemelt, siis tähistatakse suurema tasemenumbri- ga eksemplarid. Programmi järgmiste operatsioonide täitmisel kuni tingimusoperatsiooni esimese märgendiga näidatud lauseni (see välja arvatud) võetakse argumente ainult vaadeldava kirje tähistatud eksemplaridest. Kui tingimus pole rahuldatud ühegi eksemplari jaoks, siis suunatakse tingimusoperatsiooni teise märgendi, kui aga eksemplarid üldse puuduvad, siis viimase märgendi järgi. Seega võib tingimusoperatsioonis olla kuni kolm märgendit, näiteks operatsioon

TV D)K.A,B*470,510,230

tähendab, et kuni lauseni 470 tuleb vaadelda ainult neid kirje K eksemplare, kus elementidel A ja B on võrdne väärtus. Kui kirjes K ühtki sellist eksemplari ei leidu, siis suunatakse märgendile 510, vaadeldaval tasemel eksemplaride puudumisel aga märgendile 230.

Tihti saab tingimusoperatsioonis siiski piirduda ainult ühe või kahe märgendiga, sest vajaliku järjekorranumbriga märgendi puudumisel toimub suunamine viimase olemasoleva märgendi järgi (kui viimases näites märgend 230 ära jätta, siis suunatakse ka eksemplaride puudumisel märgendile 510). Analoogiline printsiip on muide kehtiv üldse STAMP-keele kõigi operatsioonide korral.

Ühe kirje teatud taseme eksemplaridele võib samaaegselt rakendada kuni seitse erinevat tingimust. Seejuures argumentide võtmisel järgnevates operatsioonides arvestatakse ainult nende tingimustega, mille mõjupiirkonnas vastav operatsioon asub. Tingimusoperatsiooni enda täitmisel eelnevaid tingimusi ei arvestata. Näiteks programmilõiguga

440 TS)5,K.A*470

450 TS)K.A,0*480

460 S)K.B=A

470 S)K.C=A

saadakse elemendi K.B väärtuseks viiest väiksemate positiivsete elementide K.A summa, elemendi K.C väärtuseks aga kõigi positiivsete elementide K.A summa.

Senivaadeldud tingimusoperatsioone on loomulik nimetada ja-tingimusteks, sest mitme niisuguse tingimusoperatsiooni ühises mõjupiirkonnas peavad iga konkreetse eksemplari jaoks argumentide võtmisel olema samaaegselt rahuldatud kõik tingimused. Leidub aga olukordi, kus mitmest tingimusest piisab ainult ühe rahuldatusest. Sellisel juhul tuleb kasutada nn. või-tingimusi operatsiooni koodidega vastavalt VTVD, VTMV, VTS ja VTSV.

Kui ja-tingimused osutuvad üksteisest sisuliselt sõltumatuteks, siis teatud taseme eksemplaridele rakendatud järjestikustes või-tingimustes arvestatakse iga järgmise korral ka eelmiste või-tingimustega. Et või-tingimuste korral on eksemplaride tähistamiseks vastavas baidis eraldatud ainult üks bitt, siis peavad mitme järjest kirjutatud või-tingimuse mõjupiirkonnad (esimesed märgendid) kokku langeda. Et suunamine või-tingimuste mitterahuldatus korral toimub teise märgendi järgi, siis kirjutatakse kaks märgendit üksnes viimasele neist. Näide või-tingimuste kasutamise kohta on toodud järgmises paragrahvis.

Kõigi tingimusoperatsioonide kirjutamisel saab kasutada ühte erandit. Nimelt võib kirjutamata jätta teiseks argumendiks oleva nulli. Selline lihtsustusreegel on vajalik peamiselt väärtuse puudumise kontrollimisel D, R ja T-tüüpi elementide korral.

Tulemuskirje trükkimiseks etteantud trükikirjelduse järgi kasutatakse operatsiooni VTR (= trükkida kirje). Tavaliselt on trükikirjeldusel sealjuures sama nimi, mis trükitava kirjelgi. Kui aga nimed erinevad (kirje trükkimiseks on koostatud mitu trükikirjeldust), siis tuleb operatsioonis VTR lisaks trükitava kirje nimele näidata tekstikonstandina ka trükikirjelduse nimi, näiteks

VTR)TK='TKB'.

Lisaks tingimusoperatsioonidele ja kohustuslikule suunamisele M saab lausete täitmise järjekorda muuta veel operatsiooniga MMUUT (= minna muutumise järgi) ja EX (= täita programmilõik). Operatsioonil MMUUT võib sealjuures olla üks

või mitu ühese väärtusega argumenti. Tavaliselt on nendeks mingi järjest sisse loetava kirje võtmelemendid, kusjuures igale argumendile vastab üks märgend. Näiteks operatsiooni

MMUUT)K.A,B,C*230,230,170

esmakordsel täitmisel jäetakse meelde elementide K.A, K.B ja K.C väärtused ning täidetakse vahetult järgmine lause. Igal järgmisel täitmisel kontrollitakse, kas mingi argumendi väärtus pole eelmise korraga võrreldes muutunud. Kui leitakse esimene selline argument (antud juhul järjekorras A, B, C), siis antakse juhtimine vastava järjekorranumbriga märgendile, säilitades ühtlasi argumentide väärtuste uue seisuga. Kui kõigi argumentide väärtused jäävad samaks, siis võetakse täitmisele järgmine lause.

Operatsiooni EX kasutatakse sellisel juhul, kui teatud komplekti järjest kirjutatud lauseid on vaja täita programmi mitmes erinevas kohas. Operatsioon on ilma argumentideta ja kolme märgendiga, näiteks

EX)*180,240,110

Esimene märgend näitab täidetava lauserühma algust ja teine esimest mittetäidetavat lauset. Kui programmi töö jõuab teise märgendiga näidatud lauseni, siis järgmisena tuleb täitmisele võtta kolmanda märgendiga näidatud lause. Kui aga kolmas märgend puudub, siis pärast lauserühma täitmist antakse juhtimine vahetult operatsiooni EX järel paiknevale lausele.

Vaatleme lõpuks veel mõningaid kändmisoperatsioone, mis paljudel juhtudel võimaldavad üldse loobuda tingimusoperatsioonide kasutamisest.

Operatsioonidega KEN (= kanda näidatud element) ja SEN (= liita näidatud elemendile) saab indeksi etteantud väärtuse järgi valida sobiva lähte- või tulemusoperandi. Olgu näiteks kirje K teatud eksemplaris elemendi K.N (indeksi) väärtuseks arv kaks. Siis lausetega

KEN)K.E=N,A,B,C

SEN)K.A,B,C=M,N

kantakse vaadeldavas eksemplaris elemendi K.E väärtuseks indeksile järgnevas argumentide loetelus teisel kohal oleva elemendi K.B väärtus ning elemendi K.M väärtus liidetakse juurde samuti teisel kohal oleva tulemusoperandi K.B väärtusele. Operatsioonis SEN on lubatud ka indeksi väärtus null, mille korral tulemusoperandide väärtusi ei muudeta.

Kahe argumendiga operatsioonides KMIN (= kanda miinimumi järgi) ja KMAX (=kanda maksimumi järgi) leitakse argumentide kombinatsioon, kus teine argument omab vähima/suurima väärtuse. Tulemuseks kantakse esimese argumendi (kuni 8-baidine element) väärtus sellest kombinatsioonist. Kantavaks ja võrreldavaks argumendiks võib olla ka sama element, näiteks

KMAX)K.M=A,A

Kui elemendi väärtus sõltub mingi teise elemendi väärtuse kuulumisest teatud vahemikku, siis on sobiv kasutada nelja argumendiga operatsiooni KVAH (= kanda vahemiku järgi). Näiteks lause

KVAH)K.A=L.A,B,C,K.V

täitmisel omistatakse elemendi K.A väärtuseks elemendi L.A väärtus sellest argumentide kombinatsioonist, kus on rahuldatud tingimus $L.B \leq K.V \leq L.C$.

10. PROGRAMMINÄITEID

Seitsmendas paragrahvis (lk. 60) vaatlesime väljundtabeli näitena aruande koostamist lao sissetulekute kohta hankijate ja materjali liikide lõikes. Sama valdkonnaga piirdumeka kolmes järgnevas STAMP-programmi näites, kus kokkusurutud kujul on püütud illustreerida põhilisi programmeerimisvõtteid ja operatsioonide mitmesuguseid kasutusviise.

Esimene näiteprogramm on mõeldud lähteandmete kontrollimiseks, kusjuures põhirõhk asetatakse tingimusoperatsioonide kasutamise selgitamisele. Teises programmis tutvustatakse andmetest sõltuva tsükli organiseerimist ning tulemuskirje koostamise üldiseid põhimõtteid. Kolmanda näitena tuakse samadel andmetel põhinev suhteliselt lihtne programm jooksvalaoseisu kogumiseks.

Kõigi toodavate näidete põhilisteks lähteandmeteks olgu laodokumendid, mis erinevalt lk. 58 näidatud väljundkirjest SSORT sisestatakse nüüd vastavalt legendile LDOK:

LEG LDOK

1 LADU X4-K

KUUP X6-K KUUPAEV

VT N1-K VEA TUNNUS (O-KORRAS,1-VIGANE)

2 DOKN X6-K DOK. NR

OP T2 LIIKUMISOPER.

KUKU X4 KUST VQI Kuhu

3 ART T10 MATERJALI ARTIKKEL

SORT N1

KOGUS I3.2

Nagu sellest legendist nähtub, sisestatakse iga lao tarvis kõik andmed antud päeval toimunud liikumiste kohta üheainsa kirjena LDOK. Sealjuures antakse veatunnuse VT väärtuseks esialgu null, s.t. sisestamisel eeldatakse andmete õigsust. Liikumise iseloomu näitab iga dokumendi korral element OP, mille võimalikest väärtustest huvitagu meid esimeses ja teises näiteprogrammis ainult sissetulekut hankijatelt tähistav operatsiooninimi SH. Elemendi KUKU tähendus sõltub üldiselt sellest operatsioonist, kuid meid huvitaval juhul olgu KUKU väärtuseks hankija siffer.

Lähtekirjeid LDOK üldiselt pikaajaliseks säilitamiseks andmebaasi ei koguta. Lisaks mitmesuguste vahetute aruannete moodustamisele kasutatakse neid nähtavasti vaid mõningate teiste kirjete (näiteks laoseisu) täiendamiseks. Et sealjuures lähtedokumentides esineda võivad vead ei kanduks edasi teistesse kirjetesse, tuleb pärast sisestamist üldiselt alati kõigepealt kontrollida lähteandmete õigsust vähemalt põhinäitajate osas.

Järgmisel leheküljel ongi toodud näiteprogramm KONTLDOK, milles kontrollitakse kirjete LDOK elementide KUKU, ART ja SORT väärtusi. Elemendi KUKU väärtusi vaadeldakse sealjuures ainult juhul, kui tegemist on operatsiooniga SH. Sisestatud sifrite usutavuse kontrollimisel võetakse aluseks varem kirjena HANKIJAD andmebaasi sisestatud hankijate nimekiri, mis olgu määratud legendiga:

LEG HANKIJAD

2 HANK X4-K HANKIJA SIFFER

HNIM T20 NIMI

TEKST KONTLDOK

- 10 LEGK)ARTIKLID,HANKIJAD
 20 LEGL)LDOK
 30 2 HANK X4
 40 3 T N1
 50 DEF)LDOK=D
 60 LUG)ARTIKLID+HANKIJAD
 70 LUG)D*270
 80 K)D(ART)T=1,ARTIKLID()
 90 TVD)D.T*110
 100 KTR)D.LADU,KUUP,DOKN,'PUUDUB ART.',ART
 110 VTS)D.SORT,3*150
 120 VTVD)D.SORT*150,150
 130 KTR)D.LADU,KUUP,DOKN,'VALE SORT',SORT
 140 K)D.T=0
 150 TVD)D.OP,'SH'*200
 160 K)D(KUKU)HANK=HANKIJAD(HANK)HANK
 170 TVD)D.HANK*200
 180 KTR)D.LADU,KUUP,DOKN,'PUUDUB HANKIJA',KUKU
 190 K)D.T=0
 200 TVD)D.T*240
 210 FOP)'S','LDOK',D.LADU,KUUP,1
 220 FPR)D.DOKN-KUKU,ART-KOGUS
 230 KUST)D.T*250
 240 TMV)D.VT*70
 250 K)D.VT=0
 260 SALV.60)D*70
 270 STOP)

Elemendi ART väärtuste kontrollimiseks eeldame, et kõigi esineda saavate artiklite loetelu on andmebaasis juba olemas (meil läheb ka kolmandas näiteprogrammis laoseisu moodustamisel iga materjali artikli jaoks vaja tema ühiku hinda). Seepärast olgu iga lubatava artikli jaoks kirjesse ARTIKLID juba sisestatud vastav eksemplar, kusjuures kirje struktuur on määratud legendiga:

LEG ARTIKLID

2 ART T10-K

HIND N3.2

Kontrollitavat kirjet LDOK tähistame lk. 87 toodud programmis lühidalt tähega D ning viime sinna ühtlasi juurde kaks ajutist tööelementi HANK ja T. Neist esimene osutub vajalikuks hankija sifri kontrollimisel, teine aga vigu sisaldava eksemplari äramärgimisel.

Programmi täitmine algab ühekordsete kirjete ARTIKLID ja HANKIJAD lugemisega. Et lugemisoperatsioonis pole kirjenimede järel näidatud märgendit, siis ükskõik kumba kirje puudumisel lõpetatakse programmi täitmine tõrkega.

Seejärel alustab programm kirjete D läbivaatamist. Kõigepealt kontrollitakse lausetega 80...100 kirje D artiklite õigsust, s.t. kas nad leiduvad lubatavate artiklite senises loetelus. Kirje D nendes kolmanda taseme eksemplarides, kus elemendi ART väärtus sisaldub kirjes ARTIKLID, kantakse lausega 80 elemendi T väärtuseks arv üks. Kui mingisse eksemplari jäi õigsuse tunnuseks arv üks kandmata, siis lause 100 trükib seda eksemplari määravad võtmeelemendid koos vastava teatega ning elemendi ART vigase väärtusega.

Elemendi SORT väärtus peab igas eksemplaris olema vahemikust 1...3 (sellise eelduse tegime juba legendis SSORT). Et element on N-tüüpi, siis vigase väärtuse korral osutub rahuldatuks üks kahest tingimusest: kas $SORT > 3$ või siis $SORT = 0$. Või-tingimusi kasutatakse programmis sellepärast, et kui mingis kirjes D ei leidunud ühtki eksemplari, kus elemendi SORT väärtus oleks olnud suurem kolmest, siis tuleb ikkagi veel kontrollida ka lausega 120 esitatud tingimuse rahuldatus. Kui ka selle lause täitmisel ei leita ühtki tingimust rahuldavat eksemplari, siis täidetakse järgmisena teise märgendiga näidatud lause 150. Nendesse eksemplaridesse aga, kus oli rahuldatud vähemalt üks nimetatud kahest tingimusest, kantakse elemendi D.T väärtuseks vea tunnusena null ja trükitakse ühtlasi vastav teade.

Lausetes 160...190 vaadeldakse tingimuse 150 toimel ainult kirje D neid teise taseme eksemplare, kus elemendi OP väärtus on SH. Kui kirje D elemendi KUKU mingile väärtusele vastavat eksemplari kirjes HANKIJAD ei leidu, siis jääb element D.HANK väärtustamata. Et laused 180 ja 190 asuvad kahe tingimuse mõjupiirkonnas, siis vastav teade antakse ainult nende eksemplaride kohta, kus need kaks tingimust on korraga rahuldatud, s.t. elemendi OP väärtus on SH ja elemendi HANK väärtus null. Veatumus ($D.T=0$) kirjutatakse kõigisse vastavale teise taseme eksemplarile alluvatesse kolmanda taseme eksemplaridesse.

Lausega 200 kontrollitakse, kas programmi eelneva osaga avastati vigu kirjes D. Kui vigu esines (s.t. leidub eksemplare, kus $D.T=0$), siis jaotatakse antud kirje kaheks kir-

jek. Vigastest eksemplaridest formeeritakse nimelt operatsioonidega FOP ja PPR uus kirje, kus veatunnus VT saab väärtuse üks, need vigased eksemplarid aga kustutatakse lähtekirjest operatsiooniga KUST (= kustutada eksemplar). Kirje D järelejäänud eksemplarid salvestatakse operatsiooniga SALV, kandes eelnevalt esimesele tasemele veatunnuse väärtuseks nulli (et VT on võtmelement, siis selle kirje võti erineb sama lao ja kuupäeva jaoks moodustatud vigaste eksemplaridega kirje võtmest).

Operatsiooni SALV modifikatsioon 60 on sisuliselt sama tähendusega nagu sisestamisoperatsioon P, s.t. vigadeta kirjet programm OUT nüüd püsifaili ei kanna. Seevastu aga veatunnusega kirjed kannab programm OUT püsifaili, sest lausega 210 formeeriti ju operatsioon S. Pärast avastatud vigade parandamist saab need kirjed seega järgmises seansis uuesti töösse lülitada.

Kui lause 200 täitmise käigus selgus, et kirjes vigu ei leidunud, siis kontrollitakse lausega 240 veel elemendi VT väärtust. Selle erinemise korral nullist (tegemist võis olla parandatud kirjega) salvestatakse sama kirje juba veatunnusega null.

Teise näitena vaatleme järgmisel leheküljel esitatud programmi SSORT, mis hankijate nimekirja ja eelmise programiga kontrollitud laodokumentide järgi koostab iga lao jaoks leheküljel 58 kasutusele võetud legendile SSORT vastava väljundkirje. Ka selles programmis kasutame sagedamini tarvitatavate kirjenimede SSORT ja LDOK asemel vastavalt lühemaid nimesid S ja D.

TEKST SSORT

10 LEGK)SSORT,HANKIJAD,LDOK
 20 DEF)SSORT=S,LDOK=D
 30 LUG)HANKIJAD
 40 LUG)D.VT=0*170
 50 MMUUT)D.LADU,KUUP*120,70
 60 K)S.LADU=D.LADU
 70 K)S.VKUUP=D.KUUP
 80 TVD)D.OP,"SH"*110,40
 90 FE)S.HANK,ART=D.KUKU,ART
 100 SEN)S(HANK,ART)S1-S3=D(KUKU,ART)KOGUS, SORT
 110 M)*40
 120 TMV)S.HANK*130,160
 130 S)S.KOKKU=S1-S3
 140 K)S(HANK)HNIM=HANKIJAD()HNIM
 150 VTR)S
 160 M)*60
 170 EX)*120,160
 180 STOP)

Programmi SSORT töö algab ühekordse kirje HANKIJAD lugemise-
 misega. Seejärel hakatakse järjest sisse lugema neid kirjeid
 D, kus veatunnuse VT väärtus on null. Operatsioon MMUUT es-
 makordsel täitmisel jäetakse meelde elementide LADU ja KUUP
 väärtused ning avatakse lausega 60 kirje S. Et sama lao kir-
 jed D on kuupäevade kasvavas järjekorras (KUUP on ju võtme-
 element), siis saame lause 70 täitmisel elemendi S.VKUUP
 lõplikuks väärtuseks viimase kuupäeva, mille kohta veel lao-
 dokumente sisestati.

Edasi kontrollitakse lausega 80, kas antud kuupäeval üldse hankijatelt midagi saabus (kas esineb operatsiooninime SH). Eitaval juhul minnakse lugema uut kirjet D. Kui aga kirjes operatsiooninimega SH eksemplare leidis, siis lisatakse kirjesse S vastavad eksemplarid hankija šifri ja artikliga (juhul kui neid seal varem polnud). Operatsiooniga SEN liidetakse nüüd vastava sordi senisele kogusele juurde kirjes D näidatud kogus.

Kui lauses 50 selgus, et ladu muutus (s.t. eelmise lao kõik kirjed on läbi vaadatud), siis tuleb asuda kirje S lõpliku formeerimise juurde. Kõigepealt kontrollitakse lausega 120, kas kirjesse S on üldse eksemplare formeeritud. Kui leidis kas või ükski teise taseme eksemplar, siis on see tingimus rahuldatud ja täitmisele tuleb järgmine lause. Et järgmistes operatsioonides mitte teha ülearust tingimuse rahuldatuse kontrolli, pole tingimuse mõjupiirkonda jäetud ühtegi operatsiooni.

Kirje S moodustamine lõpeb elementide KOKKU ja HNIM väärtustamisega ning saadud kirje trükkimisega (lk. 60 esitatud trükikirjelduse SSORT järgi). Seejärel alustatakse uue kirje S moodustamist, kusjuures arvestatakse, et esimene uue lao kirje D on juba sisse loetud.

Kui kõik kirjed D on läbi vaadatud (sunnamine lausest 40), siis enne operatsiooni STOP tuleb trükkida ka viimasele laole vastav kirje S. Et ka selles kirjes võivad eksemplarid puududa, tuli lausest 120 eksemplaride puudumisel suumata märgendi 60 asemel märgendile 160. Vastasel korral oleks programm jäänud tsüklisee.

Programmis SSORT võib kirje S koostamisel selgelt eristada kolme faasi: kirje avamine (lause 60), andmete kogumine (laused 70...100) ja lõpptöötlus (laused 120...150). Olgu märgitud, et kogumise etapi igas operatsioonis on argumentideks tsükliliselt uuendatava kirje D elemendid. Programmi kiiruse huvides on siin soovitatav vältida selliste operatsioonide sattumist tsükliisse, mida võib täita ka väljaspool tsüklit. Kui näiteks antud programmis oleksiid lausetele 130 ja 140 vastavad operatsioonid paigutatud lausete 100 ja 110 vahele, siis saaksime ju ikkagi sama lõpptulemuse, kuid neid operatsioone tarbetult korrates.

Viimase näitena vaatleme veel lao hetkeseisu näitava kirje SEIS koostamist, mis olgu määratud legendiga (mida küll programmis täiendame veel ühe tööelemendiga):

LEG SEIS

1 LADU X4-K

2 ART T10-K

KOGUS I4.2

SUMMA I7.2

Järgmisel leheküljel esitatud programmi SEIS koostamisel on arvestatud, et kõik sissetulekut tähistavad operatsioonid algavad tähega S, laost väljaminekusse puutuvad operatsioonid tähega V ja muid algustähti operatsiooni nimetustes ei tohi esineda. Operatsiooninime esitähe eraldamiseks kanname lauses 105 selle nime ühetähelise elemendi OP1 väärtuseks. Nüüd saame lausetega 110 ja 120 anda väljaminevale kogusele negatiivse väärtuse (tõsi küll, vigased operatsiooninimed loetakse kõik sissetulekuoperatsioonide nimedeks).

Täiendavat kommenteerimist vajab programmis SEIS eeskätt vast lause 160, kus kontrollitav tööelement TUN saab olla nullist erineva väärtusega ainult juhul, mil antud lae jaoks leitud vähemalt üks kirje LDOK. Kui selliseid kirjeid ei leitud, siis pole ka mõtet salvestada muutmata jäänud kirjet SEIS.

TEKST SEIS

```
10 LEGK)ARTIKLID
20 LEGL)SEIS
30 1 TUN N1
40 LEGL)LDOK
45 2 OP1 T1
50 3 SUMMA I6.2
60 DEF)LDOK=D,SEIS=S
70 LUG)ARTIKLID
80 LUG)S*180
90 LUG)D.LADU,VT=S.LADU,0*160
100 K)S.TUN=1
105 K)D.OP1=OP
110 TVD)D.OP1,'V'*130
120 LAH)D.KOGUS=0,KOGUS
130 KOR.2)D(ART)SUMMA=KOGUS,ARTIKLID( )HIND
140 FE)S.ART=D.ART
150 LM)S(ART)KOGUS,SUMMA=D( )KOGUS,SUMMA*90
160 TMV)S.TUN*80
170 SALV)S*80
180 STOP)
```

INDEKS

- argument - 63
elemendi nimi - 16
element - 13
fail - 32
fail SISE - 39
failikirjeldus - 36
fond - 36
formaadikirjeldus - 57
indeks 33, 51
infokandja kirjeldus - 36
kataloog - 34
kirje - 12
kirje LEGEND - 19
kirje nimi - 15
kirje päis - 14
kirje TEKST - 61
kirje TNT - 35
kirje TRYKL - 51
kirjeldav lause - 61
kollektor - 33
kommentaär - 71
komponent - 18
kood - 21
korduste arv - 18
lahendusseanss - 41
legend - 14
lisaelement - 19
muutuv pikkus - 18
märgend - 42, 61
omadus - 17
operatsiooni kood - 63
pilt - 16
programm IN - 45
programm OUT - 47
põhifail - 32
reakirjeldus - 51
sisendfail - 34
sisestamiskeel - 23
STAMP-programm - 61
tase - 12
tasemenumber - 15
tellimus - 41
trükikirjeIdus - 50
tulemusoperand - 63
tööfail - 35
töölause - 63
töörežiim - 48
tüübinimi - 16
veerukirjeldus - 52
viidaelement - 70
võti - 14
võtmeelement - 13

S i s u k o r d

1. Süsteemi struktuurist	3
2. Kirje ja legend	12
3. Andmete sisestamine	23
4. Failid ja fondi kirjeldamine	32
5. Tellimus	41
6. Kirjete viimine andmebaasi	45
7. Väljundtabelite kirjeldamine	50
8. STAMP-programmid	61
9. STAMP-keele operatsioonid	72
10. Programminäiteid	85
Indeks	95