



ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ МЕМЛЕКЕТТІК СТАНДАРТЫ

АҚПАРАТТЫҚ ТЕХНОЛОГИЯ

Бағдарламалық құралдарды статикалық талдау

ҚР СТ 34.008-2002

Ресми басылым

**Қазақстан Республикасы Индустрия және сауда министрлігінің
Стандарттау, метрология және сертификаттау жөніндегі комитеті
(Мемстандарт)**

Астана

Алғы сөз

1 ТК 34 «Ақпараттық технологиялар» стандарттау жөніндегі техникалық комитет **ӨЗІРЛЕНІП ЕНГІЗІЛДІ**

2 Қазақстан Республикасы Индустрия және сауда министрлігінің Стандарттау, метрология және сертификаттау жөніндегі комитетінің 2002 жылғы 29 желтоқсандағы № 519 бұйрығымен **БЕКІТІЛІП ҚОЛДАНЫСҚА ЕНГІЗІЛГЕН**

**3 БІРІНШІ ТЕКСЕРУ МЕРЗІМІ
ТЕКСЕРУ КЕЗЕҢДІЛІГІ**

**2007 жыл
5 жыл**

4 АЛҒАШ РЕТ ЕНГІЗІЛДІ

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ МЕМЛЕКЕТТІК СТАНДАРТЫ

АҚПАРАТТЫҚ ТЕХНОЛОГИЯ

Бағдарламалық құралдарды статикалық талдау

Енгізілген күні 2004-01-01

1 Қолданылу саласы

Осы стандарт бағдарламалық құралдарға (ПҚ) статикалық талдау жүргізудің тәртібі мен ережесін белгілейді.

Осы стандарттың ережесі Қазақстан Республикасының Мемлекеттік сертификаттау жүйесінде аккредиттелген сынақ орталықтарында (зертханаларында) міндетті түрде қолданылуға жатады.

Осы стандарт БҚ-дың рәсімдік бағдарламалауды қолдана отырып әзірленген мақсаттарына және қолданылу салаларына қарамастан барлығына таратылады.

2 Нормативтік сілтемелер

Осы стандартта мынадай стандартқа сілтемелер пайдаланылды:

ГОСТ 28195-99 Бағдарламалық құралдардың сапасын бағалау. Жалпы ережелер.

3 Анықтамалар

Осы стандартта мынадай анықтамалар пайдаланылды:

- | | |
|--|--|
| 3.1 БҚ шақырулар графы | Бағдарламалық компоненттері (модульдері) төбелер қызметін атқаратын, ал бағытталған қыры шақыратын компоненттен шақырылатынына бағдарланған бағдарлы граф. |
| 3.2 Басқару графы | Қисынды блоктары төбелері болып табылатын, ал бағытталған қырлары блоктар арасында басқаруды беру бағытына бағдарланған бағдарлы граф. |
| 3.3 Бағдарламаларды динамикалық талдау | Бағдарламаны ЭЕМ орындау нәтижесінде динамикалық сипаттамаларды алу және талдау процесі (бағдарламаны орындау трассасы, тестілеу толықтығы, көлемді уақытша сипаттамалар). |

3.4 Бағдарламаның логикалық блогы	Бір немесе бірнеше операторлардан тұратын және тармақталмайтын бағдарламаның учаскесі.
3.5 Шақырулар және қол жеткізу матрицасы	Бағдарламалық компоненттердің (модульдердің) ерікті парлары арасында шақыру және қол жеткізу қатынастарын сипаттайтын матрица.
3.6 Бағдарламалық құрал (БҚ)	ГОСТ 28195 бойынша.
3.7 Шақыру жолдары	Бастапқы төбесі графтың түбірі, ал ақырғысы – бұтақтың парағы болатын, шақырулар графасындағы қиысатын қырларының дәйектілігі.
3.8 Тестілеу жолы	Бастапқы төбесі графтың кіру төбесі, ал ақырғы төбесі – графтың шығу төбесі болып табылатын, бағдарламалық модульді басқару графасындағы маршрут.
3.9 Бағдарламалық құралдарды сүйемелдеу	Бағдарламалық құралдардың пайдаланушылардағы жұмысқа қабілеттілігі күйін қолдау жөніндегі және бағдарламалық құралдарды жетілдіру жөніндегі қызмет.
3.10 Бағдарламаларды статикалық талдау	Бағдарламаның ЭЕМ орындаусыз бастапқы мәтінін талдау процесі
3.11 Шақыру нүктесі	Бағдарламалық компоненттердің (модульдің) шақыруы тұрған жер
3.12 Сарапшы (сапасы бойынша	Кәсіби және квалиметрикалық құзыреттің талаптарына жауап беретін білікті маман .

4 Жалпы ережелер

Бағдарламаны статикалық талдау:

- БҚ модульдік құрылымын, сондай-ақ жекелеген модульдердің логикалық құрылымын тексеру және осы құрылымдарды бағдарламалық құжаттамаларда көрсетілгендермен салыстыру;
- БҚ динамикалық талдау жүргізу үшін бастапқы деректерді дайындау және БҚ тестілеу жоспарын әзірлеу;
- бағдарламаның конструктивтік сипаттамаларын, бағдарламаны модификациялау мен сүйемелдеудің қарапайымдылық дәрежесін бағалау;
- бағдарламаның пайдаланылмайтын, артық ауыспалы учаскелеріндегі бағдарламаның жетілмегендігінің болуын айқындау;
- бағдарламаның мәтіндік қиындығын, оны әзірлеу мен меңгеруге кеткен шығынды бағалау;

- авторлықты белгілеу және құқықтық дауларды шешу үшін бағдарламаның бірдейлігіне сараптама;

- бағдарламаның сапа деңгейін бағалау кезінде сандық сипаттамаларын анықтау (ГОСТ 28195 сәйкестігін) үшін жүргізіледі.

Статикалық талдау бағдарламаны жобалау сатысынан бастап (ұлғайтылған талдау) бағдарламалық құралдың өмірлік циклының келесі барлық фазаларына жүргізіледі.

5 Статикалық сипаттамалар

Бағдарламалық құралды статикалық талдау мынадай сипаттамаларды алуды көздейді: (графикалық және метрикалық):

- бағдарламалық құралдың модульдік құрылымы;
- жекелеген бағдарламалық модульдің логикалық құрылымы;
- бағдарлама мәтінінің сипаттамалары.

Талданған БҚ-дың модульдік құрылымы мынадай түрде ұсынылады:

- шақыру графтары;
- шақыру жолдарының тізімі;
- шақыру және қолжетімділік матрицасы;
- шақыру нүктелері;
- шақырудың иерархия метрикасы.

Жекелеген бағдарламалық модульдің логикалық құрылымы мынадай түрде ұсынылады:

- басқару графы;
- тестілеу жолдары;
- басқару құрылымының метригі.

Бағдарлама мәтінінің сипаттамалары мыналарды қамтиды:

- бағдарламаның түсініктілігі туралы статикалық деректер;
- мәтіндік Холстед метригі.

5.1 БҚ графикалық, кестелік және матрицалық сипаттамалары

5.1.1 Шақырулар графы

Бағдарламалық құралдың шақыру графтары БҚ модульдік құрылымын қамтиды (А қосымшасы). Графтың төбесі компоненттердің (модульдердің) атауы бар тікбұрыш түрінде, қыры – стрела түрінде бейнеленген.

5.1.2 Шақырулар жолдарының тізімі

Шақыру жолдары бірінші қырдың алғашқы төбесі графтың түбірі болатын, ал соңғысы әрқашанда бұтақтар тәрізді құрылым болмайтын компонент болып есептелетін, яғни бұл бұтақ жапырағы болатын шақыру графтарынан қырлардың жалғасып, тұтасу тізбектілігін көрсетеді (қырдың соңғы төбесі келесі қырдың алғашқы төбесі болып табылады) (А қосымшасы).

Мұндай тізімде тестіленетін шақырулардың барлық жолдары көрсетілген. Сонымен бірге бұл тізімде бағдарламаны сүйемелдеуге арналған қажетті ақпараттар бар:

- компонент шақырулардың тізбектілігі негізінде кейбір модификацияның мүмкіндікті нәтижелерін тікелей көрсетуі мүмкін;
- шақыру жолдарының ұзындығын, сондай-ақ компоненттерді анықтаған кезде модификациялануы мүмкін компоненттің орналасқан жерін және пайда болу жиілігін дереу білуге болады;
- тізім мәтіндік мысалдардың тізбесін жасауға мүмкіндік береді.

5.1.3 Шақырулар және қол жеткізу матрицасы

Шақыру және қол жеткізу матрицасында компоненттердің ерікті парлары арасында құрылымның екі негізгі типі туралы ақпарат бар. Шақырулар иерархиясындағы элементтер мынадай өзара байланыстардың біреуінде болуы мүмкін:

- олардың біреуі тікелей екіншісін шақырады;
- шақырулар графында осы пар элементтерінің біреуінде басталып, екіншісінде аяқталатын жол бар, демек, парлар қарайтын элементтер бір-бірінен тікелей шақырылуы мүмкін емес, тек тізбекті шақырулар тізбекшесі арқылы шақырылуы мүмкін.

Шақырулар және қол жеткізу матрицасы мынадай сұрақтарға жауап береді:

- егер А модулін өзгертсе, В модуліне ықпалын тигізуі мүмкін бе;
- А модулі шақыратын модульдердің саны қандай және ол қандай модульдер;
- А модулі қол жеткізген, бірақ оған шақырылмайтын модульдердің саны қандай және ол қандай модульдер;
- қандай модульдер қол жетпейтін болып табылады (ешқашан шақырылмайтын);
- қандай граф төбелері соңғы болып табылады (шақырылмайтын).

Бұл ақпараттар бағдарламалық өнімдерді тестілеу, сүйемелдеу және пайдалану кезінде пайдаланылады.

Шақырулар және қол жеткізу матрицасының мысалы А қосымшада көрсетілген.

Матрицаның жолдары мен бағандарында шақыру иерархиясы компонентінің аты бар және жоғарыдан төмен қарай иерархиялық деңгейлер бойынша реттестірілген. Крес жолға сәйкес келетін компонентті тікелей бағанмен ерекшеленген компонент шақыратынын білдіреді. Жанама шақыру фактісі нүктемен белгіленген (шақырулардың тізбекшесі арқылы қол жеткізіледі). Матрицаның тиісті диагональдық элементі ретінде крес немесе нүкте тұратын, рекурсивтік шақыру жағдайынан басқа, диагональдық элементтер дефиспен белгіленді.

5.1.4 Шақырулар нүктесі

Статикалық талдау процесінде бағдарламаның бастапқы мәтінінде шақырулардың нақты тұрған орны белгіленеді. Бұл мынадай себептерге сүйенеді:

- кез келген компонент басқаны бірнеше рет шақыруы мүмкін, яғни шақыру операторы бастапқы мәтіннің көптеген жолдарында орналасуы мүмкін;
- компонент бастапқы мәтіннің жүзден астам жолында болған жағдайда, тіпті оның осы компонентте орналасқаны белгілі болса да, шақырудың нақты операторын табу өте қиын;
- бағдарламалаудың көптеген тілінде компоненттерді шақыру оның атын көрсетумен іске асырылады, бұл формальды жағынан алып қарағанда ауыспалы атаумен қабылдануы мүмкін.

Шақырулар нүктесінің кестесі (қосымша) үш бағаннан тұрады: шақырылатын компоненттердің атауы, шақырудың әрекет ету саласын белгілеу және шақыру операторларының орналасуы жолдар мен бағандардың нөмірлерімен берілетін шақырулар нүктесінің координаты.

Шақырудың әрекет ету саласы шақырылатын компоненттің компиляция блогында анықталуы немесе анықталмауына байланысты ішкі немесе сыртқы болуы мүмкін. Іздеудің қарапайымдылығын қамтамасыз ету үшін шақырылатын компоненттердің атауы алфавиттік тәртіппен келтірілген.

5.1.5 Басқару графы

Кез келген бағдарлама бір немесе бірнеше логикалық блоктар түрінде көрсетілуі мүмкін. Логикалық блок тармақталмайтын бағдарлама учаскесі (операторлардың тізбектелген жинағы) болып табылады: демек, егер бірінші оператор блогы орындалса, онда осы блоктың қалған операторлары да ізінше орындалады.

Бағдарламаның басқару графы - төбесі логикалық блок болып табылатын, ал қырлары – блоктар арасында басқаруды беру болып табылатын бағдарланған граф (А қосымша).

5.1.6 Тестілеу жолдары

Кіру деректерінің таңдалған жинағына байланысты бағдарлама операторлары әртүрлі тәртіппен орындалуы мүмкін, яғни, бағдарламаның кіру нүктесінен шығу нүктесіне жетудің көптеген тәсілдері бар. Тестілеу жолы басқару графындағы маршрут сияқты белгіленеді, оның бастапқы төбесі графтың төбесі болып, ал ақырғы төбесі - графтың шығу төбесі болып табылады. Бағдарламада тестілеудің мүмкіндікті жолдарының шексіз үлкен саны болуы мүмкін. Бұл жағдайда әрбір тармақты ең болмағанда бір рет жабатын маршруттардың кейбір жинағын орындауға саяды. Жоғарыда аталған талаптарды қанағаттандыратын тестілеу жолдарының санамалануы қосымшада келтірілген, мұнда бірінен кейінгі бірі орындалатын төбелер орындау тәртібін білдіретін сызықшалармен қосылған.

5.2 БҚ сандық сипаттамалары

Осы бөлімшеде ГОСТ 28195 сәйкес бағалау элементтері ретіндегі БҚ сапасы көрсеткіштерінің номенклатурасына енгізілген статикалық сипаттамалар келтірілген (қайта енгізілген көрсеткіштер жұлдызшамен белгіленеді).

Факторы: Сүйемелденуі.

Өлшемі: С2 – Конструкцияның қарапайымдылығы.

С01 – Жобаның жүйелік архитектурасының қарапайымдылығы.

С0103 – Иерархиялық күрделілік:

$$I = \frac{N}{L}, \quad (1)$$

мұнда: N – модульдерді шақыру графындағы төбелердің саны;

L – деңгейлердің саны.

Шақыру графасындағы деңгейдің орташа енін, яғни, бағдарлама әзірлеудің жеке кадамында қабылданатын жобалық шешімдер санын сипаттайды.

С0104 – Құрылымдық күрделілік:

$$S = \frac{D}{N}, \quad (2)$$

мұнда: D - модульдерді шақыру графындағы қырдың саны;

N – төбенің саны.

Құрылымдық күрделілік графтың нысанын сипаттайды: бұтақ (қарапайым құрылым) немесе желі (күрделі құрылым). Қарапайым құрылым (бұтақ) құрылымдалған бағдарлама үшін сипатты.

С0105 - Тестіленілуі:

$$T = \left[\frac{1}{N} \sum_{i=1}^N \frac{1}{P_i} \right]^{-1}, \quad (3)$$

мұнда: N - модульдерді шақыру графындағы шақыру жолдарының саны;

P_i - тестіленілуі i- шақыру жолдары, тең:

$$P_i = \left[\sum_{j=1}^k \frac{1}{A(M_j)} \right]^{-1},$$

мұнда: k - шақырулар жолдарындағы модульдердің саны;

$A(M_j)$ – жолға арналған P_i , модульдің мүмкіндігі M_j , мынадай болып анықталады:

$$A(M_j) = \sum_{x=1}^n \frac{A(M_x)}{C(M_x)},$$

мұнда: $A(M_x)$ – M_j – шақыратын, x -модулдің мүмкіндігі;

$C(M_x)$ – M_x шақыратын барлық модульдердің саны;

n – M_j шақыратын модульдердің саны

$A(M)=1$, егер модуль M ең жоғарғы (бас) модул болып табылатын болса.

Торапқа (модульге) қол жеткізу шақыру бұтағының тарамдалуына байланысты осы модульді шақырудың құрылымдық ықтималдылығын сипаттайды.

Егер тестілеудің көрсеткіші үлкен мәнге ие болса, онда иерархияның төмен деңгейлері модулдерін тестілеу, әсіресе, автоматты тестілеу әдістерін қолдану кезінде қиындай түседі.

C0106 – Шақырулар Энтропиясы:

$$E = \frac{\sum_{i=1}^N L_i \log_2 \left(\frac{L_i}{\sum_{i=1}^N L_i} \right)}{\sum_{i=1}^N L_i}, \quad (4)$$

мұнда: L_i - ұзындығы (модульдер саны) i -шақырулар графындағы жолдар.

Шақырулардың тестіленуі және энтропиясы тиісінше иерархиялық және құрылымдық күрделіліктен өзгеше алғанда бағдарламаның динамикасындағы жай-күйін барынша дәл біліп отыруға мүмкіндік береді.

C16 – Бағдарламалық модульдің қарапайымдылығы.

C1601 – Модулдің орташа өлшемі:

$$N=N_1+N_2, \quad (5)$$

мұнда: N_1 – бағдарламадағы операторлардың жалпы саны;

N_2 - бағдарламадағы операндтардың жалпы саны.

C1602 – модульдің орташа мәтіндік күрделілігі (Холстед бойынша әзірлеуге күш салу):

$$E = \frac{V}{L}, \quad (6)$$

мұнда: V – бағдарламаның көлемі:

$$Y = N \log_2 Y,$$
$$Y = Y_1 + Y_2.$$

мұнда: - Y_1 бағдарламадағы бірегей операторлардың саны;

Y_2 - бағдарламадағы бірегей операндтардың саны;

L – бағдарлама деңгейі:

$$L = \frac{V^*}{V},$$

V^* - бағдарламаның әлеуетті көлемі:

$$V^* = (2+U) \log_2 (2+U),$$

мұнда: U – бағдарламаның деректерді кіргізу және шығару саны .

Құрылымдық бағдарламаның күрделілігі оның мөлшерімен корреляцияланатындықтан, бұл көрсеткішті құрылымдалмаған компоненттерді бағалау үшін қолданған жөн.

$C1604$ – Орта цируломатикалық сан (Маккейб бойынша күрделілік):

$$C = e - n + 2, \quad (7)$$

мұнда: e – модульдің логикалық құрылым графындағы доғаның саны;

n – графтағы төбенің саны.

Цикломатикалық сан ең болмағанда бір рет модульдің қисынды құрылымы графының әрбір доғасын жабу үшін қажетті мәтіндік жолдардың ең аз санына тең. Бұл көрсеткіш сондай-ақ тестілеу (сынау) жоспарын бағалау үшін пайдаланылуы мүмкін.

$C1605$ – Модул құрылымын басқаратын орта энтропия:

$$E = \frac{\sum_{i=1}^N P_i \log_2 \left(\frac{P_i}{\sum_{i=1}^N P_i} \right)}{\sum_{i=1}^N P_i}, \quad (8)$$

мұнда: N – тестілік жолдардың ең төменгі саны;

P_i – i -тестілік жолдар ұзындығы (логикалық блоктардың саны).

Бұл көрсеткіш мақсаты бойынша $C0106$ ұқсас және жекелеген модуль (барлық БҚ үшін емес) үшін пайдаланылады.

Өлшемдері: $C3$ – Көрнекілігі.

$C09$ – Бағдарлама мәтінін рәсімдеу.

$C0905$ – Мәтіннің түсіндірілуі:

$$C = \frac{N_c}{N}, \quad (9)$$

мұнда: N_c – бағдарламаның бастапқы мәтінінің түсіндірілген жолдарының саны;

N – бағдарламаның бастапқы мәтіні жолдарының жалпы саны

6 Статикалық талдау жүргізудің тәртібі мен ережесі

6.1 Статикалық талдаудың автоматтандырылған әдістері

Статикалық талдаудың қол әдісі арнайы сарапшының бағдарламалардың бастапқы мәтінін жете түрде қарауы болып табылады. Әдістің келесідей кемшіліктері бар:

- уақыттың және адам ресурстарының көп жұмсалуына әкеп соғады;
- жоғары білікті сарапшы (бағдарламашы сияқты сапа сарапшысын да) тартуды қажет етеді;
- бағдарламалардың үлкен көлемінен және күрделілігінен нәтижелердің дәлдігі әрдайым қанағаттанарлықтай емес (бірқатар жағдайларда адамның мүмкіндігі қолдан талдауды қолдануға мүмкіндік бермейді).

Автоматтандырылған құралдар осылайша сарапшыға алынған сипаттамаларды бағдарлама сапасының деңгейіне интерпретациялауға баса назар аударуға мүмкіндік бере отырып, ең еңбек жұмсалымды сипаттамаларды айқындауға немесе көмек етуге қабілетті.

6.2 Бағдарламаның өмірлік циклының жекелеген кезеңдеріндегі статикалық талдаудың ерекшеліктері.

6.2.1 Талдау және жобалау кезеңдері

Осы кезеңдерде статикалық талдауды қолдану, ең алдымен, машинамен өңделетін нысанда нәтижелерді сақтауды қамтамасыз ететін формальды ерекшелігі құралының болуын қажет етеді.

Осы кезеңдердегі статикалық талдау ең алдымен қателіктерді және ауытқуларды анықтау үшін қолданылады, яғни бағалау сипаты бар бағдарламалау кезеңіне қарағанда сапаны басқару құралы болып табылады.

Өлшенетін сипаттамалардың шектеулігіне қарамастан, талдау және жобалау кезеңдерінде статикалық талдауды қолдану бұдан кейінгі кезеңдерде түзетілуі қиын қателерді күні бұрын табуға ықпал етеді.

6.2.2 Бағдарламалау кезеңі

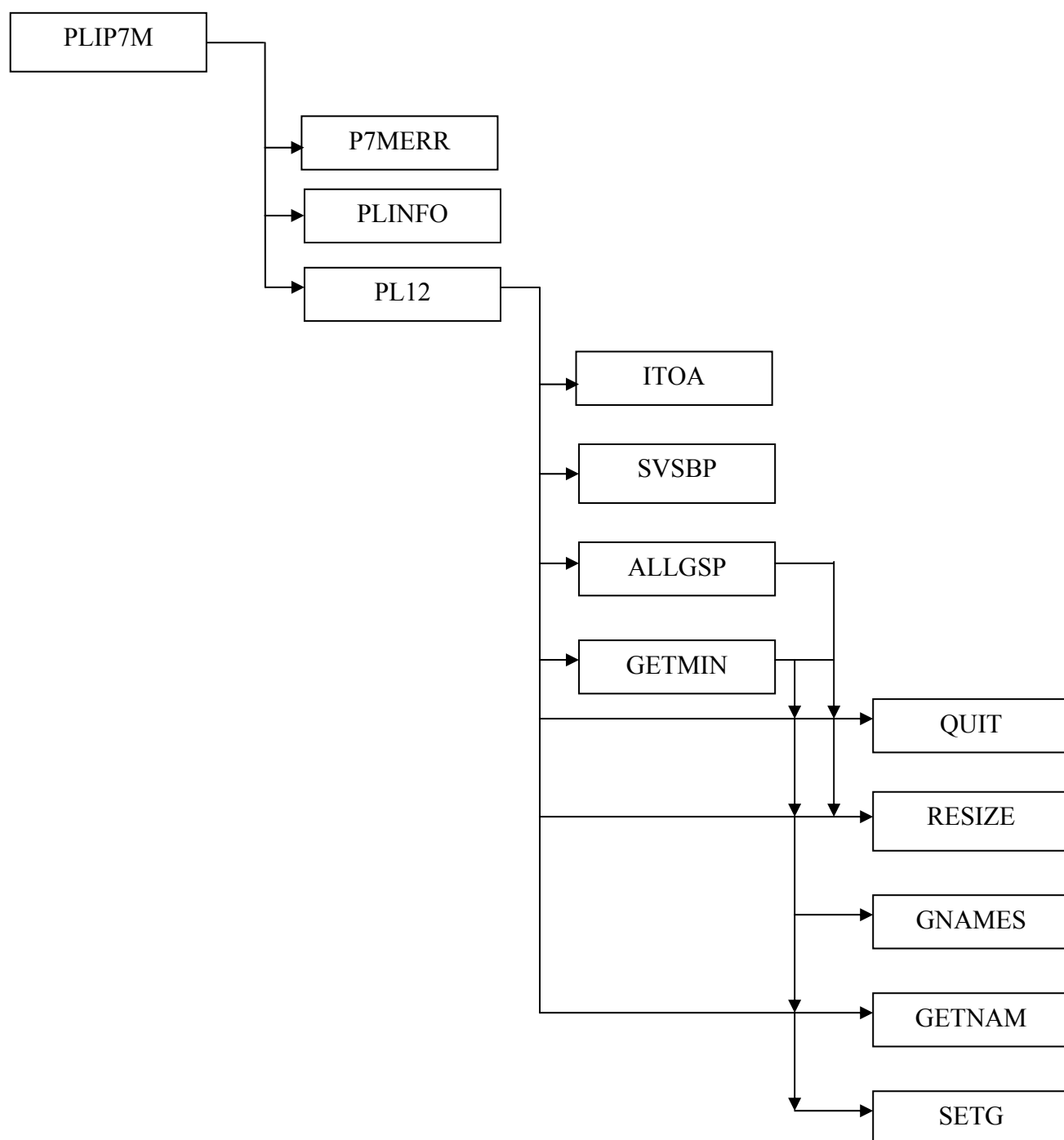
Бұл кезеңде бағдарламаның сапасын бағалау үшін, сондай-ақ құжаттау үшін пайдаланылады (сүйемелдеу құжаттарын әзірлеу кезінде).

6.2.3 Тестілеу және жөндеу кезеңі

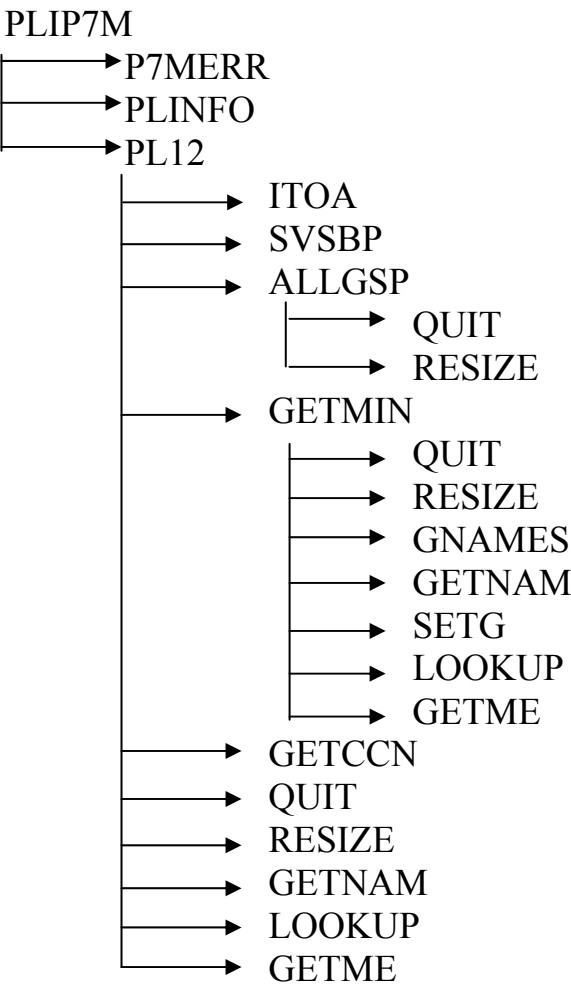
Бұл кезеңде статикалық талдау мәтіндерді дайындау және толықтылығын бағалау кезінде бастапқы ақпарат алу үшін қолданылады.

А қосымшасы

Мысалдар



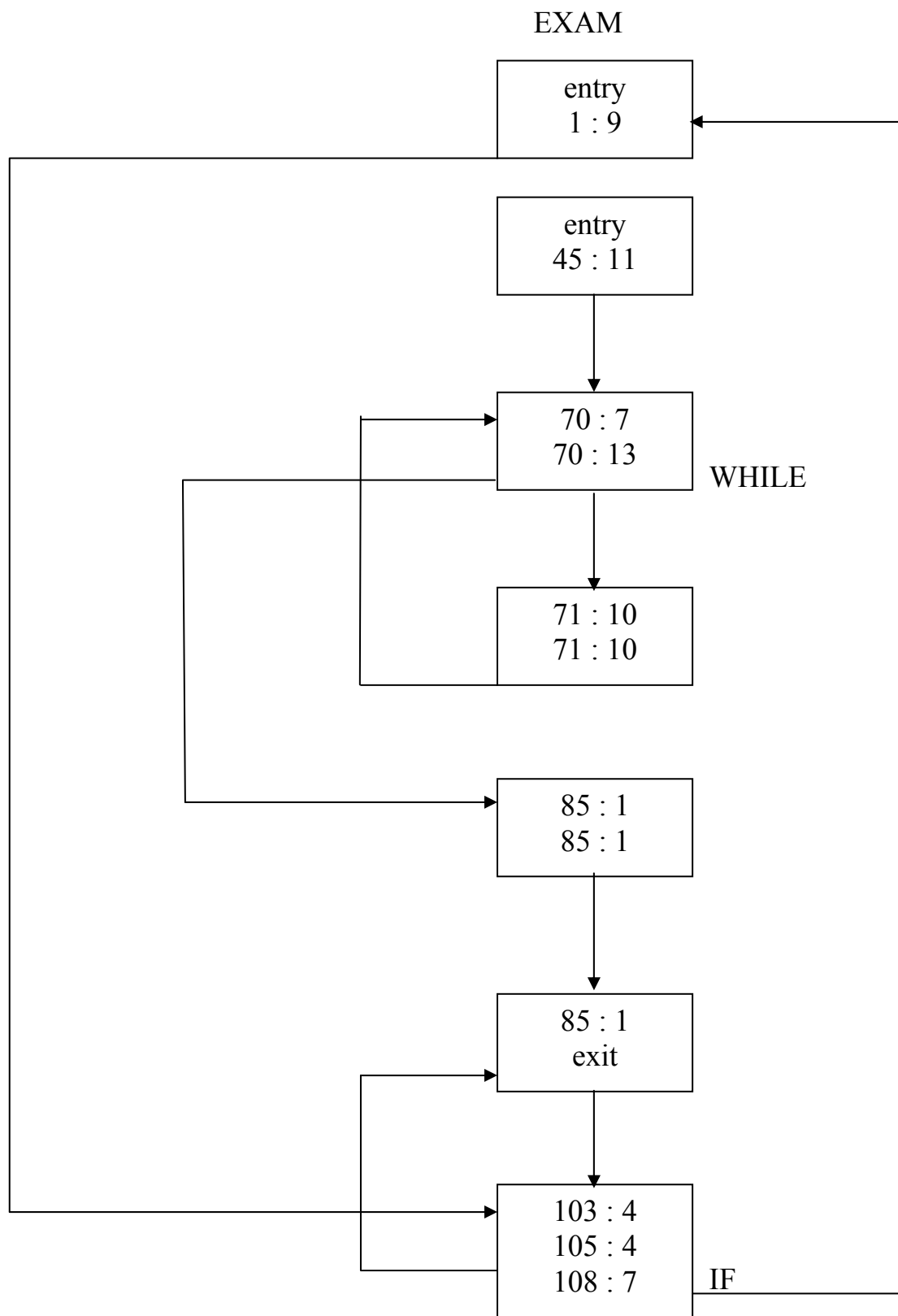
1-сурет – БҚ шақырулар графы мысалы



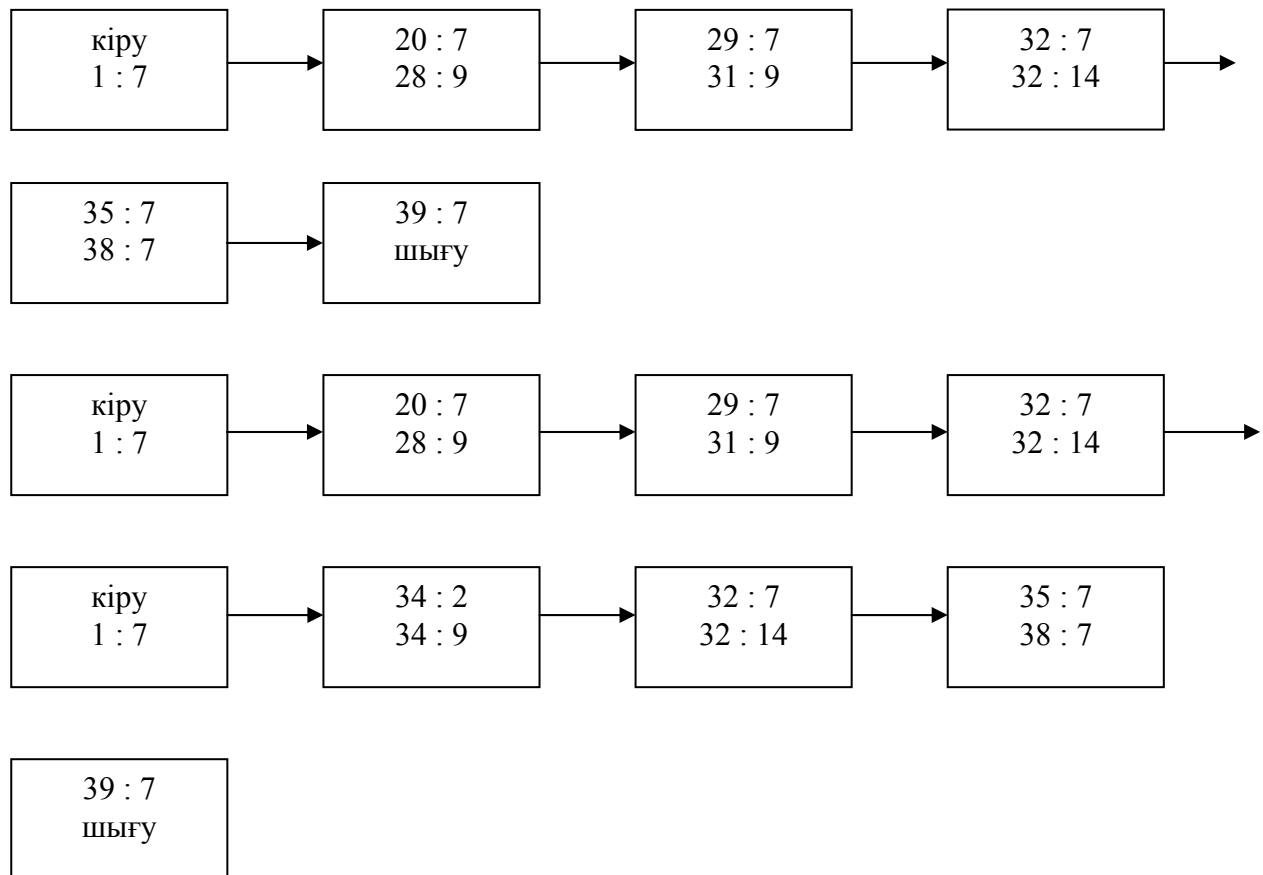
2-сурет – Бағдарламаларды шақыру жолдары мысалы

		1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1
		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6
1	PLIP7M	-	x	x	x	x	.	.	.	.	.	.	.	.	.	.	.
2	P7MERR		-														.
3	PLINFO			-													
4	PL12				-	x	x	x	x	x	x	.	x	.	x		x
5	ITOA					-											
6	SVSBP						-					x					
7	ALLGSP							-		x		x					
8	GETMIN								-	x		x	x	x	x		x
9	GETCCN																
10	QUIT									-							
11	RESIZE											-					
12	GNAMEs												-				
13	GETNAM													-			
14	SETG														-		
15	LOOKUP															-	
16	GETME																-

**3-сурет – Бағдарламалар үшін шақыру және қол жеткізу матрицалары
мысалы**



4-сурет – Басқару графы мысалы (EXAM бағдарламасы)



5-сурет – Бағдарламаны тестілеу жолдары мысалы

1-кесте – PLIP7M бағдарламасындағы шақырулар нүктесі

Шақырылатын компонент	Әрекет ету саласы	Шақыру нүктесі (жол : баған)
P7MERR	сыртқы	28 : 20
PL12	сыртқы	37 : 7
PLINFO	сыртқы	29 : 7

ӘОЖ

МСЖ 35.080

Түйінді сөздер: бағдарламалық құрал, сапа көрсеткіштері, статикалық талдау.



ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ КАЗАХСТАН

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

Статический анализ программных средств

СТ РК 34.008-2002

Издание официальное

**Комитет по стандартизации, метрологии и сертификации
Министерства индустрии и торговли Республики Казахстан
(Госстандарт)**

Астана

Предисловие

1 РАЗРАБОТАН И ВНЕСЕН Техническим комитетом по стандартизации ТК 34 «Информационные технологии»

2 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ приказом Комитета по стандартизации, метрологии и сертификации Министерства индустрии и торговли Республики Казахстан от 29 декабря 2002 г. № 519

**3 СРОК ПЕРВОЙ ПРОВЕРКИ
ПЕРИОДИЧНОСТЬ ПРОВЕРКИ**

**2007 год
5 лет**

4 ВВЕДЕН ВПЕРВЫЕ

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ КАЗАХСТАН

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ
Статический анализ программных средств

Дата введения 2004-01-01

1 Область применения

Настоящий стандарт устанавливает порядок и правила проведения статического анализа программных средств (ПС).

Положения настоящего стандарта подлежат обязательному применению в испытательных центрах (лабораториях), аккредитованных в Государственной системе сертификации Республики Казахстан.

Настоящий стандарт распространяется на все ПС независимо от их назначения и области применения, разработанные с применением процедурного программирования.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующий стандарт:

ГОСТ 28195-99 Оценка качества программных средств. Общие положения.

3 Определения

В настоящем стандарте использованы следующие определения:

- | | |
|----------------------------------|--|
| 3.1 Граф вызовов ПС | Ориентированный граф, в котором вершинами служат программные компоненты (модули), а направленные ребра ориентированы от вызывающей компоненты к вызываемой. |
| 3.2 Граф управления | Ориентированный граф, вершинами которого являются логические блоки, а направленные ребра ориентированы в направлении передачи управления между блоками. |
| 3.3 Динамический анализ программ | Процесс получения и анализа динамических характеристик в результате выполнения программы на ЭВМ (трасса выполнения программы, полнота тестирования, объемно-временные характеристики). |

3.4 Логический блок программы	Участок программы, состоящий из одного или нескольких операторов и не имеющий разветвления.
3.5 Матрица вызовов и достижимости	Матрица, характеризующая отношение вызова и достижимости между произвольными парами программных компонентов (модулей). По ГОСТ 28195.
3.6 Программное средство (ПС)	
3.7 Путь вызовов	Последовательность соприкасающихся ребер из графа вызовов, где начальная вершина есть корень графа, а конечная – лист дерева.
3.8 Путь тестирования	Маршрут в графе управления программного модуля, начальная вершина которого является входной вершиной графа, а конечная вершина – выходной вершиной графа.
3.9 Сопровождение программного средства	Деятельность по поддержанию программного средства в работоспособном состоянии у пользователей и по совершенствованию программного средства.
3.10 Статический анализ программ	Процесс анализа исходного текста программы без ее выполнения на ЭВМ.
3.11 Точка вызова	Местоположение вызова программной компоненты (модуля), задаваемое номером строки и столбца расположения оператора вызова.
3.12 Эксперт (по качеству)	Квалифицированный специалист, отвечающий требованиям профессиональной и квалиметрической компетентности.

4 Общие положения

Статический анализ программ проводится для:

- проверки модульной структуры ПС, а также логической структуры отдельных модулей и сравнения этих структур с приведенными в программной документации;
- подготовки исходных данных для проведения динамического анализа ПС и разработки плана тестирования ПС;
- оценки конструктивных характеристик программы, степени простоты модификации и сопровождения программы;
- определения наличия несовершенств в программе, неиспользуемых участков программы, лишних переменных;

- оценки текстовой сложности программы, затрат на ее разработку и освоение;
- экспертизы идентичности программ для установления авторства и решения правовых споров;
- определения количественных характеристик при оценке уровня качества программы в соответствии с ГОСТ 28195.

Статический анализ проводится, начиная со стадии проектирования программы (укрупненный анализ) и на всех последующих фазах жизненного цикла программного средства.

5 Статические характеристики

Статический анализ программного средства предусматривает получение следующих характеристик (графических и метрических):

- модульная структура программного средства;
- логическая структура отдельного программного модуля;
- характеристики текста программы.

Модульная структура анализируемого ПС представляется в виде:

- графа вызовов;
- списка путей вызовов;
- матрицы вызовов и достижимости;
- точек вызовов;
- метрик иерархии вызовов.

Логическая структура отдельного программного модуля представляется в виде:

- графа управления;
- путей тестирования;
- метрик структуры управления.

Характеристики текста программ включают в себя:

- статистические данные о комментированности программы;
- текстовые метрики Холстеда.

5.1 Графические, табличные и матричные характеристики ПС

5.1.1 Граф вызовов

Граф вызовов программного средства отражает модульную структуру ПС (приложение А). Вершины графа изображены в виде прямоугольников, содержащих имена компонент (модулей), ребра – в виде стрелок.

5.1.2 Список путей вызовов

Путь вызова представляет собой последовательность соприкасающихся ребер из графа вызовов (конечная вершина ребра является начальной вершиной следующего ребра), где начальная вершина первого ребра есть корень графа, а конечная всегда представляет собой

компоненту, которая не содержится древовидной структурой, то есть это лист дерева (приложение А).

В таком списке представлены все пути вызовов, которые подлежат тестированию. Кроме того, этот список содержит полезную информацию для сопровождения программы:

- на основании последовательности вызовов компонент можно непосредственно указать возможные результаты некоторой модификации;
- можно немедленно узнать длину пути вызовов, а также местонахождение и частоту появления компонент, что важно при определении компонент, которые могут быть модифицированы;
- список дает возможность составления перечня текстовых примеров.

5.1.3 Матрица вызовов и достижимости

Матрица вызовов и достижимости содержит информацию о двух основных типах структур между произвольными парами компонент. Элементы в иерархии вызовов могут находиться в одной из следующих взаимосвязей:

- одна из них непосредственно вызывает другую;
- в графе вызовов существует путь, начинающийся на одном из элементов данной пары и заканчивающийся на другом, т.е. элементы рассматриваемой пары не могут быть вызваны друг из друга непосредственно, а лишь через цепочку последовательных вызовов.

Матрица вызовов и достижимости позволяет ответить на следующие вопросы:

- если изменить модуль А, то может ли это как-то повлиять на модуль В;
- каково число модулей, вызываемых модулем А и что это за модули;
- каково число модулей, достижимых модулем А, но не вызываемых на него, и что это за модули;
- какие модули являются недостижимыми (никогда не вызываемыми);
- какие вершины графа являются конечными (не содержащими вызова).

Эта информация используется при тестировании, сопровождении и эксплуатации программных продуктов.

Пример матрицы вызовов и достижимости приведен в приложении А.

Строки и столбцы матрицы содержат имена компонент иерархии вызовов и упорядочены по иерархическим уровням сверху вниз. Крестик обозначает, что компонента, соответствующая строке, непосредственно вызывает компоненту, специфицированную столбцом. Точкой обозначен факт косвенного вызова (достижение реализуется через цепочку вызовов). Диагональные элементы обозначены дефисом за исключением случая рекурсивного вызова, при котором в качестве соответствующего диагонального элемента матрицы будет стоять крестик или точка.

5.1.4 Точка вызовов

В процессе статического анализа фиксируются точные местоположения вызовов в исходном тексте программы. Это обуславливается следующими причинами:

- любая компонента может вызвать другую много раз, т.е. оператор вызова может располагаться во многих строках исходного текста;
- в случае, когда компонента содержит более ста строк исходного текста, очень трудно найти конкретный оператор вызова, даже если известно, что он находится в этой компоненте;
- во многих языках программирования вызов компоненты реализуется просто указанием ее имени, которое с формальной точки зрения может быть воспринято именем переменной.

Таблица точек вызовов (приложение) состоит из трех столбцов: имен вызываемых компонент, обозначения области действия вызова и координат точек вызовов, задаваемых номерами строк и столбцов расположения операторов вызова.

Область действия вызова может быть внутренней или внешней в зависимости от того, была ли вызываемая компонента определена в блоке компиляции или нет. Для обеспечения простоты поиска имени вызываемых компонент перечисляются в алфавитном порядке.

5.1.5 Граф управления

Любая программа может быть представлена в виде одного или нескольких логических блоков. Логический блок представляет собой участок программы (последовательный набор операторов), не имеющий разветвлений: т.е. если выполняется первый оператор блока, то последовательно выполняются и остальные операторы этого блока.

Граф управления программы представляет собой ориентированный граф, вершинами которого являются логические блоки, а ребрами – передача управления между блоками (приложение А).

5.1.6 Пути тестирования

Операторы программы могут исполняться в различном порядке в зависимости от выбранного набора входных данных, т.е. существует много способов достижения точки выхода из точки входа программы. Путь тестирования определяется как маршрут в графе управления, начальная вершина которого является вершиной графа, а конечная вершина – выходной вершиной графа. Программа может иметь бесконечно большое число возможных путей тестирования. В этом случае тестирование сводится к исполнению некоторого набора маршрутов, покрывающего каждую ветвь хотя бы один раз. Перечисление путей тестирования, удовлетворяющих указанным выше требованиям, приводится в приложении, где вершины, представляющие исполнимые друг за другом блоки, соединены стрелками, обозначающими порядок исполнения.

5.2 Количественные характеристики ПС

В настоящем подразделе приводятся статические характеристики, включенные в номенклатуру показателей качества ПС в качестве оценочных элементов в соответствии с ГОСТ 28195 (звездочкой обозначаются вновь вводимые показатели).

Фактор: Сопровождаемость.

Критерий: С2 - Простота конструкции.

С01 - Простота системной архитектуры проекта.

С0103 - Иерархическая сложность:

$$I = \frac{N}{L}, \quad (1)$$

где: N - количество вершин в графе вызовов модулей;

L - количество уровней.

Иерархическая сложность характеризует среднюю ширину уровня в графе вызовов, т.е. количество проектных решений, принимаемых на отдельном шаге разработки программы.

С0104 - Структурная сложность:

$$S = \frac{D}{N}, \quad (2)$$

где: D - количество ребер в графе вызовов модулей;

N - количество вершин.

Структурная сложность характеризует форму графа: дерево (простая структура) или сеть (сложная структура). Для структурированной программы характерна простая структура (дерево).

С0105 - Тестируемость:

$$T = \left[\frac{1}{N} \sum_{i=1}^N \frac{1}{P_i} \right]^{-1}, \quad (3)$$

где: N - количество путей вызовов в графе вызовов модулей;

P_i - тестируемость i -го пути вызова, равная:

$$P_i = \left[\sum_{j=1}^k \frac{1}{A(M_j)} \right]^{-1},$$

где: k - количество модулей в пути вызовов;

$A(M_j)$ - доступность модуля M_j , принадлежащего пути P_i , определяется следующим образом:

$$A(M_j) = \sum_{x=1}^n \frac{A(M_x)}{C(M_x)},$$

где: $A(M_x)$ - доступность x -го модуля, вызывающего M_j ;

$C(M_x)$ - количество всех модулей, которые вызывает M_x ;

n - количество модулей, которые вызывают M_j ;

$A(M)=1$, если модуль M является самым верхним (головным) модулем.

Доступность узла (модуля) характеризует структурную вероятность вызова этого модуля, зависящую от разветвленности дерева вызовов.

Если показатель тестируемости имеет малое значение, то затрудняется тестирование модулей нижних уровней иерархии, особенно при применении автоматизированных методов тестирования.

C0106 - Энтропия вызовов:

$$E = \frac{\sum_{i=1}^N L_i \log_2 \left(L_i / \sum_{i=1}^N L_i \right)}{\sum_{i=1}^N L_i}, \quad (4)$$

где: L_i - длина (количество модулей) i -го пути в графе вызовов.

Тестируемость и энтропия вызовов в отличие соответственно от иерархической и структурной сложности позволяют более точно предсказывать поведение программы в динамике.

C16 - Простота программного модуля.

C1601 - Средний размер модуля:

$$N=N_1+N_2, \quad (5)$$

где: N_1 - общее количество операторов в программе;

N_2 - общее количество операндов в программе.

C1602 - Средняя текстовая сложность модуля (усилия на разработку по Холстеду):

$$E = \frac{V}{L}, \quad (6)$$

где: V - объем программы:

$$Y=N\log_2 Y,$$

$$Y=Y_1+Y_2.$$

где: - Y_1 количество уникальных операторов в программе;

Y_2 - количество уникальных операндов в программе;

L - уровень программы:

$$L = \frac{V^*}{V},$$

V^* - потенциальный объем программы:

$$V^*=(2+U)\log_2(2+U),$$

где: U - число входных и выходных данных программы.

Этот показатель целесообразно применять для оценки неструктурированных компонентов, поскольку сложность структурной программы коррелирует с ее размером.

C1604 - Среднее цикломатическое число (сложность по Маккейбу):

$$C = e - n + 2, \quad (7)$$

где: e - число дуг в графе логической структуры модуля;

n - число вершин в графе.

Цикломатическое число численно равно минимальному количеству текстовых путей, необходимых для покрытия каждой дуги графа логической структуры модуля хотя бы один раз. Этот показатель может быть использован также для оценки плана тестирования (испытаний).

C1605 - Средняя энтропия управляющей, структуры модуля:

$$E = \frac{\sum_{i=1}^N P_i \log_2 \left(P_i / \sum_{i=1}^N P_i \right)}{\sum_{i=1}^N P_i}, \quad (8)$$

где: N – минимальное количество тестовых путей;

P_i – длина (количество логических блоков) i -го текстового пути.

Этот показатель по назначению аналогичен C0106 и используется для отдельного модуля (а не для всего ПС).

Критерий: C3 – Наглядность.

C09 – Оформление текста программы.

C0905 - Комментированность текста:

$$C = \frac{N_c}{N}, \quad (9)$$

где: N_c – количество комментированных строк исходного текста программы;

N – общее количество строк исходного текста программы.

6 Порядок и правила проведения статического анализа

6.1 Автоматизированные методы статического анализа

Ручной метод статического анализа, заключается в детальном просмотре исходного текста программ специальным экспертом. Метод имеет следующие недостатки:

- приводит к большим затратам времени и человеческих ресурсов;

- требует привлечения эксперта высокой квалификации (как программиста и как эксперта качества);

- точность результатов не всегда удовлетворительна из-за больших объемов и сложности программ (в ряде случаев человеческие возможности не позволяют применить ручной анализ).

Автоматизированные средства способны вычислить или помочь в определении самых трудоемких характеристик, позволяя таким образом эксперту сосредоточить внимание на интерпретации полученных характеристик в уровень качества программы.

6.2 Особенности статического анализа на отдельных этапах жизненного цикла программы

6.2.1 Этапы анализа и проектирования

Применение статического анализа на этих этапах требует прежде всего наличия инструментального средства формальной спецификации, обеспечивающего хранение результатов в машинообрабатываемой форме.

Статический анализ на этих этапах применяется прежде всего для выявления ошибок и аномалий, т.е. является средством управления качеством в отличие от этапа программирования, где имеет оценочный характер.

Применение статического анализа на этапах анализа и проектирования, несмотря на ограниченность измеряемых характеристик, способствует раннему обнаружению ошибок, трудноисправимых на последующих этапах.

6.2.2 Этап программирования

На этом этапе статический анализ используется для оценки качества программы, а также для документирования (при разработке документов сопровождения).

6.2.3 Этап тестирования и отладки

На этом этапе статический анализ применяется для получения исходной информации при подготовке и оценке полноты тестов.

Приложение А

Примеры

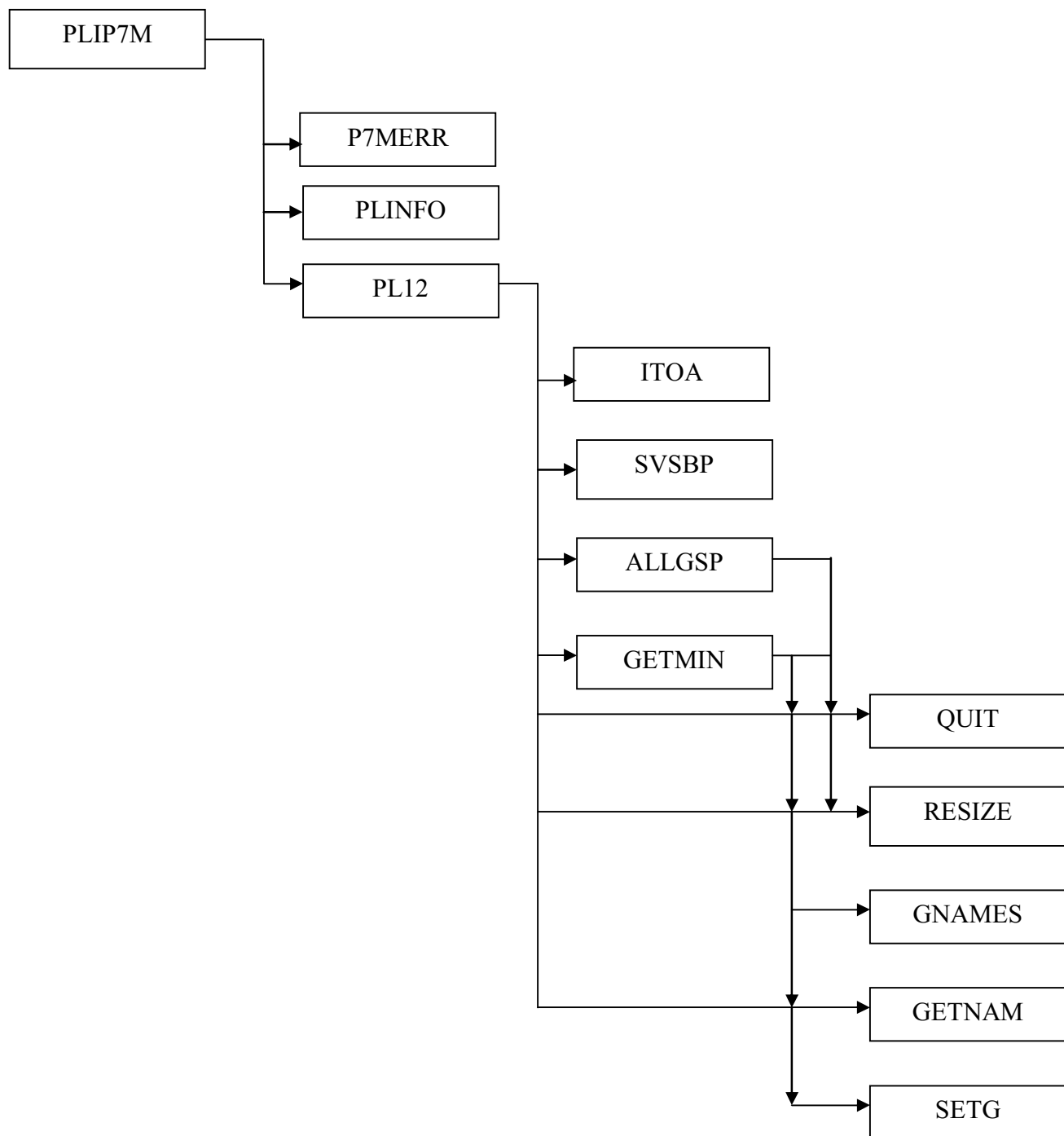


Рисунок 1 – Пример графа вызовов ПС

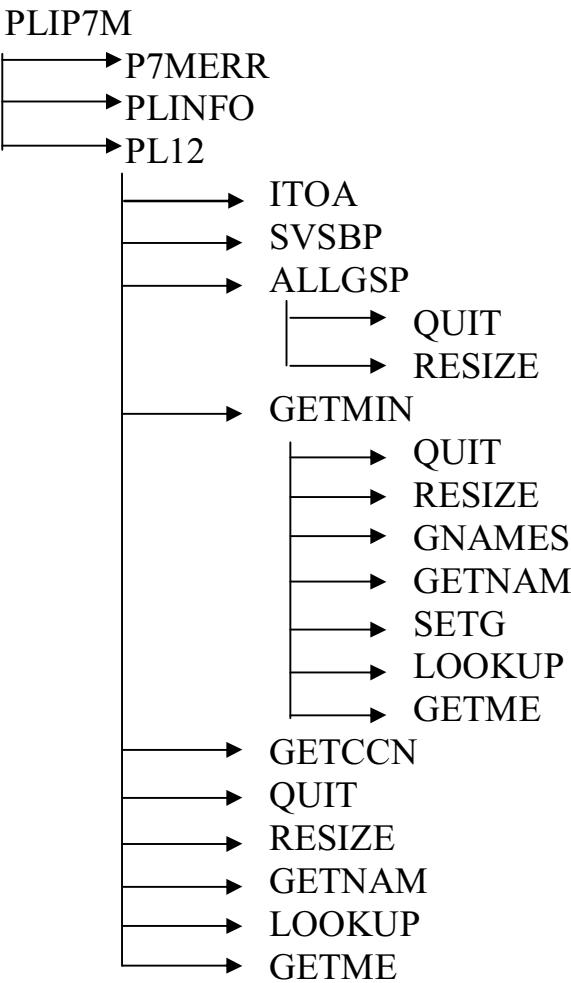


Рисунок 2 – Пример путей вызовов программы

												1	1	1	1	1	1
1												1	2	3	4	5	6
1	PLIP7M	-	x	x	x	x	.	.	.	.	.	.	.	.	.	.	.
2	P7MERR	-															.
3	PLINFO			-													
4	PL12				-	x	x	x	x	x	x	x	.	x	.	x	x
5	ITOA				-												
6	SVSBP						-			x		x					
7	ALLGSP						-			x		x					
8	GETMIN							-		x		x	x	x	x	x	x
9	GETCCN								-								
10	QUIT									-							
11	RESIZE										-						
12	GNAMEs											-					
13	GETNAM												-				
14	SETG													-			
15	LOOKUP														-		
16	GETME																-

Рисунок 3 – Пример матрицы вызовов и достижимости для программы

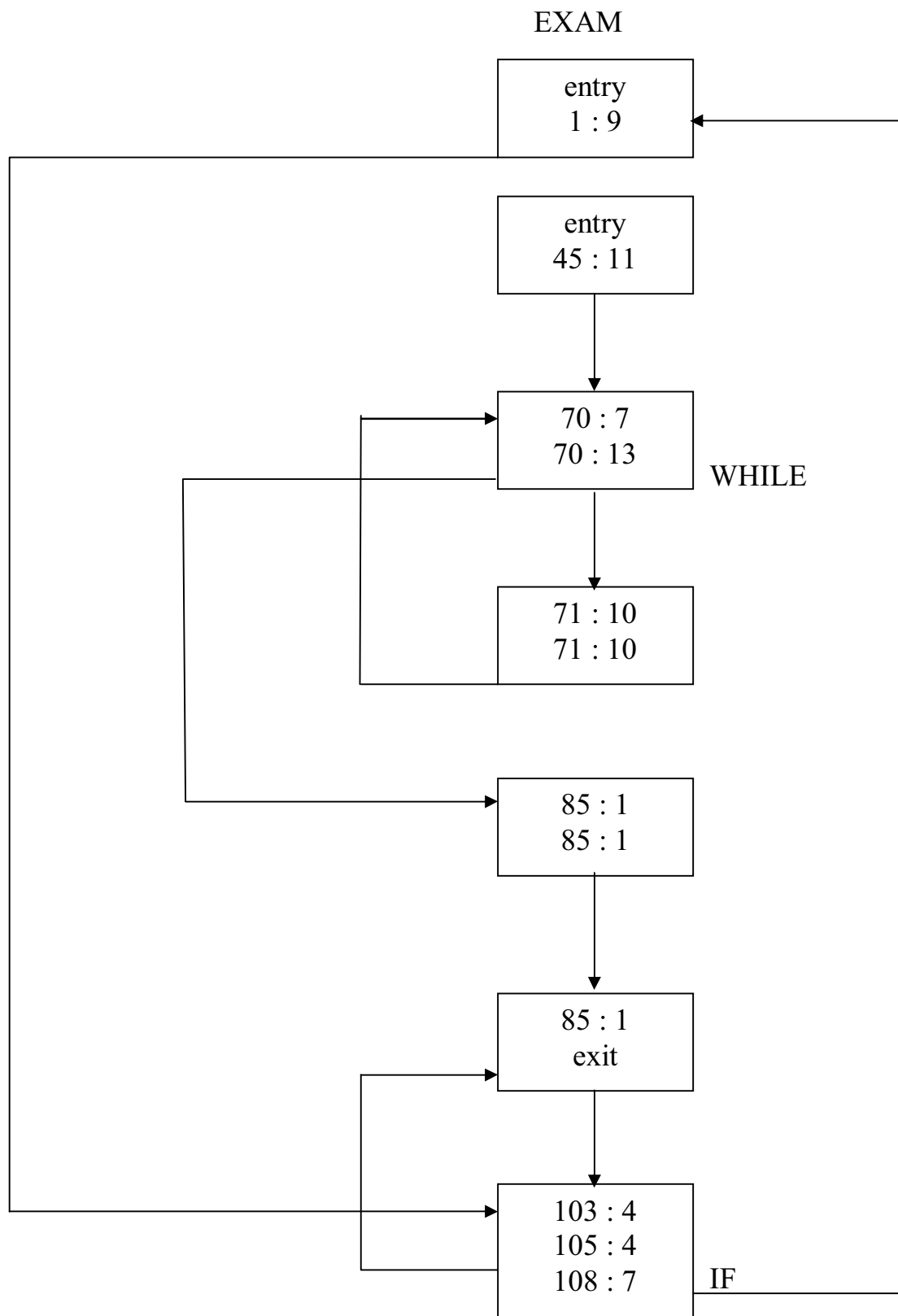


Рисунок 4 – Пример графа управления (программа EXAM)

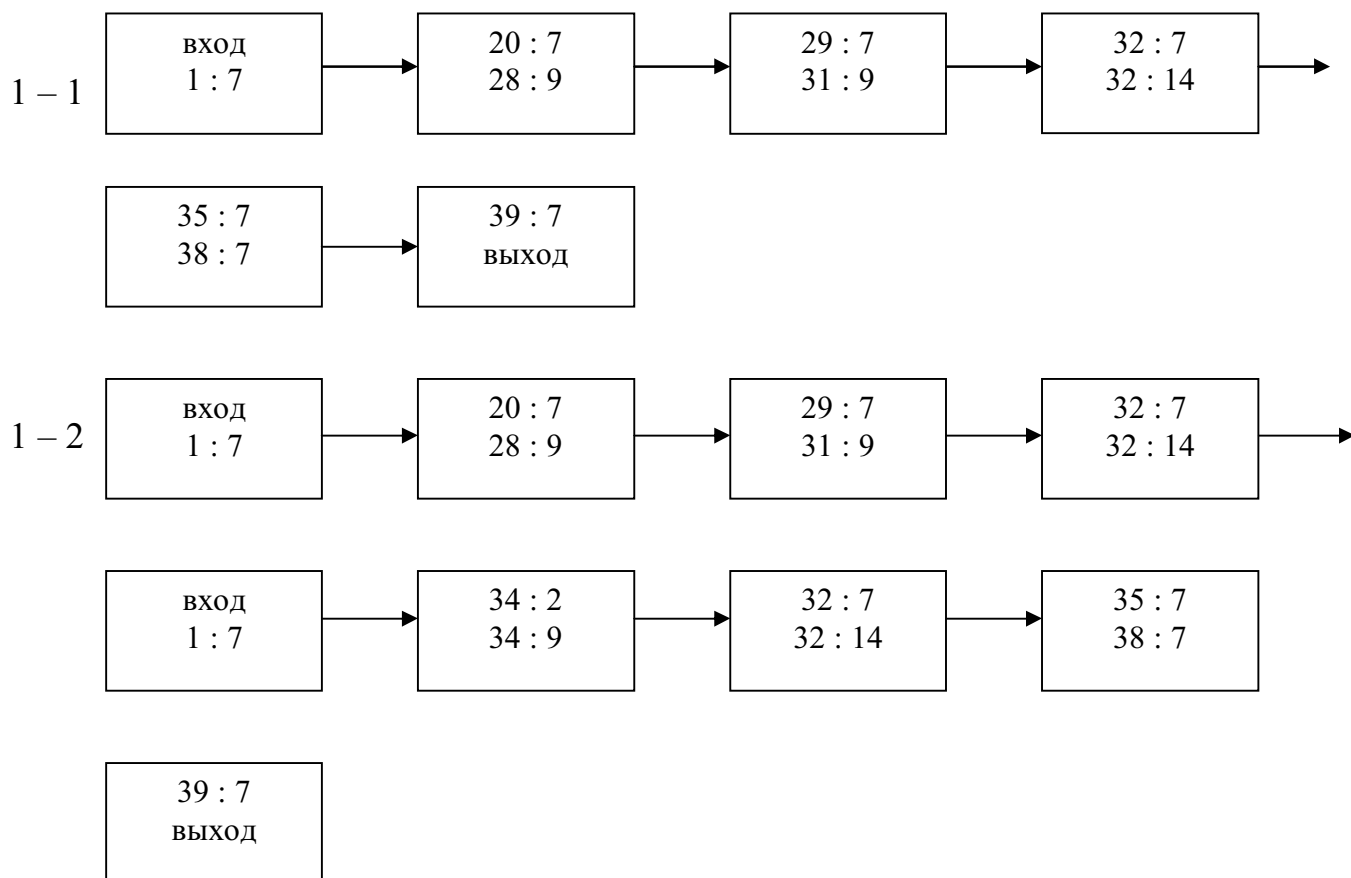


Рисунок 5 – Пример путей тестирования программы

Таблица 1 - Точки вызовов в программе PLIP7M

Вызываемая компонента	Область действия	Точка вызова (строка : столбец)
P7MERR	внешняя	28 : 20
PL12	внешняя	37 : 7
PLINFO	внешняя	29 : 7

УДК

МКС 35.080

Ключевые слова: программное средство, показатели качества, статический анализ.