

**ЎЗБЕКСТАН РЕСПУБЛИКАСЫ ЖОҚАРЫ ҲӘМ ОРТА АРНАЎЛЫ  
БИЛИМЛЕНДИРИЎМИНИСТРЛИГИ**  
**БЕРДАҚ АТЫНДАҒЫ ҚАРАҚАЛПАҚ МӘМЛЕКЕТЛИК УНИВЕРСИТЕТИ**

---

Математика факультети

**«Әмелий математика хәм информатика» кафедрасы**

**Программаластырыў тийкарлары  
пәнинен**

Лекция тексти

(«Әмелий математика хәм информатика» бакалавр бағдары 1-2-курс студентлери  
ушын)

Дүзген:

Нарымбетов И.

НӨКИС-2007

## М а з м у н ы

Кирисіу	.....
I. Турбо- Паскаль тилинің қурамы	.....
§ I.1. Турбо - Паскаль тилинің алфавити	.....
§ I.2. Программаластырыу анықтамалары хәм сөзлиги	....
§ I.3. Турақлылар, өзгеріушілер	.....
§ I.4. Скаляр мағлыұматлардың беріліу түрлері	.....
§ I.5. Мағлыұматлардың санақлы хәм шегараланған формада беріліуі...	
§ I.6. Мағлыұматлар беріліуінің бірдейлігі хәм сәйкеслігі	.....
§ I.7. Турбо - Паскаль тилинде программа дүзілісі	.....
§ I.8. Программа текстін жазыуда пунктуация қағыйдалары	.....
II. АҢЛАТПА, ОПЕРАНД, ФУНКЦИЯЛАР	.....
§ 2.1. Арифметикалық операциялар	.....
§ 2.2. Салыстырыу операциялары	.....
§ 2.3. Логикалық операциялар	.....
§ 2.4. Стандарт математикалық функциялар	.....
§ 2.5. Операциялардың орынланыу басқышлары	.....
§ 2.6. Арифметикалық аңлатпалар	.....
§ 2.7. Логикалық аңлатпалар	.....
III. ТУРБО - ПАСКАЛЬ ТИЛИНІҢ ОПЕРАТОРЛАРЫ	
§ 3.1. Әпиұайы операторлар	.....
§ 3.1.1. Меншиклестириу операторы	.....
§ 3.1.2. Шәртсиз өтиу операторы	.....
§ 3.1.3. Бос оператор	.....
§ 3.1.4. Процедураны орынлау операторы	.....
§ 3.1.5. Тамамлау операторы	.....
§ 3.2. Мағлыұматларды оқытыу - шығарыу процедуралары	.....
§ 3.2.1. Мағлыұматларды оқытыу процедуралары	.....
§ 3.2.2. Мағлыұматларды шығарыу процедуралары	.....
§ 3.2.3. Мағлыұматларды шығарыу үлгілері	.....
§ 3.3. Қурамалы есаплау процессін шөлкемлестириу операторлары	.....
§ 3.3.1. Шәрт бойынша басқарыу операторы	.....
§ 3.3.2. Вариант бойынша сайлау операторы	.....
§ 3.4. Такрарланыушы есаплау процессін шөлкемлестириу операторлары.	
§ 3.4.1. Цикл параметри бойынша тақирарлау операторлары	.....
§ 3.4.2. Алдағы шәрт бойынша тақирарлау операторлары	.....
§ 3.4.3. Ақырғы шәрт бойынша тақирарлау операторы	.....
IV. ЖӘРДЕМШИ ПРОГРАММА ХӘМ ФУНКЦИЯ ОПЕРАТОРЛАРЫ	
§ 4.1. Функция оперторын шөлкемлестириу	.....
§ 4.2. Процедура оперторын шөлкемлестириу	.....
§ 4.3. Параметрлер механизми	.....
§ 4.4. Идентификаторлардың хәрекетте болыу областы	.....
§ 4.5. Процедура, функция операторларын пайдаланыу өзгешеликлері	....
§ 4.6. Рекурсив жәрдемши программа	.....
§ 4.7. Сыртқы жәрдемши программаны дағазалау	.....
§ 4.8. SYSTEM модулиннің қосымша процедура хәм функциялары	.....
§ 4.8.1. Пограмма орынланыуын басқарыу процедуралары	.....
§ 4.8.2. Мағлыұматлар түрлерін түрлендириуші функциялар	.....
§ 4.8.3. Санақлы мағлыұматларды түрлендириуші функциялар хәм процедуралары	.....
§ 4.8.4. Параметрлер менен жұмыс ислеу функциялары	.....
§ 4.8.5. Адреслер менен ислеуде пайдаланыушы функциялар	.....
§ 4.8.6. Арнаулы процедуралар хәм функциялар	.....
§ 4.8.7. Тосынарлы санларды анықлау функциялары	.....

## КИРИСИҮ

Паскаль алгоритмлик тилинің дәслепки версиясы 1969-1971 жыллар аралығында профессор Никлаус Вирт басшылығында Цюрих (Швейцария) қаласында жоқары техникалық оқыу орнында студенттерди оқыту мақсетінде жаратылған. Биринши механикалық есаплау машинасын ислеп шыққан 1642 ж, француз илимпазы Блезь Паскаль (1623 - 1662) атына қойылған. 1976 жылдан баслап Паскаль тили дүнья жүзи алгоритмлик тил ретінде кабыл етилди.

ЭЕМде қойылған мәселени шешиуді программаластырыу хәм нәтийже алыу, жетерли дәрежеде машақатлы, көп ақыл мийнетин талап етеди. Соның ушында алгоритмлик тиллер жаратылғанда қойылатуғын мақсет: көпшилик ушын үйрениуге аңсат, илим-техниканың хәм халық хожалығының кең тарауында пайдаланыу мүмкиншиликлери, компьютерлердиң көпшилик түрлеринде пайдаланыуы есапка алынады. Усы мақсетте бирнеше мыңлаған алгоритмлик тиллер ислеп шығылған: Фортран, Бейсик, Алгол, PL, СИ, Лого, Рапира, Ада, Пролог, Кобол, Лисп, Симула х.т.б.

Буларға косымша ЖЭЕМ (компьютерлерди) математикалық тәмийнлеу системасы ТУРБО-БЕЙСИК, ТУРБО-ПАСКАЛЬ, ТУРБО-СИ, СИ ++ заманагөй программаластырыу тиллер менен тәмийнленген.

Турбо-Паскаль тилинің басқа тиллерден артықмашылықлары:

1. Тил дүзилисинің эпиуайылығы, оны үйрениуге аңсатлығы, қурамалы алгоритмлерди программаластырыуда ықшамлығы, анық түсиникли программалар дүзиу, программада кеткен кәтиликлерди тез табуу мүмкиншилигине ийе. Операторлар жетерлише- аз санда алынған.

2. Санлы, символлы хәм битли информациялар менен жұмыс ислеуде мағлыұматлар дүзилисин қолайлы шөлкемлестиріу мүмкиншиликлерине ийе. Программада пайдаланылатуғын хәр-бир өзгериушилер қайсы түрдеги мағлыұматларды өзлестиретуғынлығы алдын-ала анықланған болады.

3. Турбо-Паскаль тилинде арнаулы компиляторларды дүзиу методикасы ислеп шығылған. Нәтийжеде, көпшилик миникомпьютерлер ушын пайдаланыу мүмкиншилигине ийе.

4. Компиляторды оптималластырыу қәсийети ықшамлы программалар дүзиуге мүмкиншилик жаратады. Нәтийжеде, Турбо-Паскаль тилинде системалық программалар дүзиу мүмкиншиликлерине ийе.

5. Турбо-Паскаль тилинде структуралық программаластырыу идеясы ислеп шығылды. Бул мүмкиншилик программаны көрсетпели түрде қайта ислеуге хәм әмелий программалар жыйнағын дүзиуге кең мүмкиншилик жаратады.

### Түрлендириу формулалары

### 2- қосымша

Функциялар	Аңлатпалар	Анықланыу области
1. Секанс	$Sc x = 1 / \cos(x)$	$x \neq \pi/2 + \pi * k$ $k = 0, 1, 2, 3, \dots$
2. Косеканс	$Cs x = 1 / \sin(x)$	$x \neq \pi/2 + \pi * k$
3. Тангенс	$Tg x = \sin(x) / \cos(x)$	$x \neq \pi/2 + \pi * k$
4. Котангенс	$Ctg x = \cos(x) / \sin(x)$	$x \neq \pi * k$
5. Арксинус	$Argsinx = \arctg(x / \sqrt{1-x^2})$	$x \in [-1, 1]$
6. Арккосинус	$Argcosx = \pi/2 - \arctg(x / \sqrt{1-x^2})$	$x \in [-1, 1]$
7. Арккотангенс	$Argctgx = \pi - \arctg(x)$	$x \in [-\infty, \infty]$
8. Гипорболалык синус	$Shx = (e^x - e^{-x}) / 2$	$x \in [-\infty, \infty]$
9. Гипорболалык косинус	$Chx = (e^x + e^{-x}) / 2$	$x \in [-\infty, \infty]$
10. Гипорболалык тангенс	$Thx = (e^x - e^{-x}) / (e^x + e^{-x})$	$x \in [-\infty, \infty]$
11. Гипорболалык котангенс	$Cthx = (e^x + e^{-x}) / (e^x - e^{-x})$	$x \neq 0$
12. Гиперболалык арксинус	$Ashx = \ln(x + \sqrt{x^2 + 1})$	$x \in [-\infty, \infty]$
13. Гиперболалык арккосинус	$Achx = \ln(x + \sqrt{x^2 - 1})$	$x \geq 1$
14. Гиперболалык тангенс	$Athx = 0.5 * \ln((1+x) / (1-x))$	$ x  < 1$
15. Гиперболалык котангенс	$Acthx = 0.5 * \ln((1-x) / (1+x))$	$ x  > 1$
16. N санынын А		

тыйкары бойыша логарифми	$\text{LOG } a N = \text{Ln}(N) / \text{Ln}(a)$	$a, N > 0 ; a < 1 .$
17. Корсеткишли функция	$a^x = \text{Exp}(x * \text{Ln}(N) / \text{Ln}(a))$	$a, N > 0 ; a < 1 .$

## I. ТУРБО-ПАСКАЛЬ ТИЛИНИҢ ҚУРАМЫ

Халық тиллеринде информациялар (хабарлар) анық белгилер (символлар) жәрдеминде жазылады хәм оқылады. Программаластырыў тиллериде халық тиллерине уқсас болып, белгилер жәрдеминде **сөзлер** хәм **аңлатпалар** жазылады. Ал, сөз хәм аңлатпаларды бириктириўден операторлар (гәплер) дүзиледи. Белгилер топары **ASCII (American Standart Code for Information InterChange)** коды менен анықланылады. Олардың саны **255** ке тең (1 - косымша).

### § 1.1. Турбо-Паскаль тилиниң алфавити

**Алфавит**-бул, программаластырыў ямаса алгоритмлик тиллерде программаны дүзиўде (жазыўда) пайдаланатуғын анық сандағы белгилер топары. Дүзилген программаны хәм информацияларды ЭЕМ (компьютер) қайта ислеуи ушын, хәрбир алгоритмлик тил (АТ) өз алфавитлик дүзилесине ийе.

Турбо-Паскаль тилиниң алфавитлик дүзилеси :

**А. Хәриплер** -26 баспа ,үлкен хәм жазба, киши латын хәриплери

|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|,

|a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|.

**В. Цифрлар** (10 араб цифлары) 1, 2, 3, 4, 5, 6, 7, 8, 9, 0;

Оналтылық санақ системасындағы цифрлар 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F .

**С. Арнаўлы символлар**

- арифметикалық әмеллер | + | - | \* (көбейтиў) | / (бөлиў) | ;

- қатнас ямаса салыстырыў әмеллери | < | <= | > | >= | <> | = | ;

- бөлиўши ямаса айырыўшы белгилер | . | , | ~ | ' | ` | ( ) | [ ] | { } | | \_ | - бос орын (пробел) | \_

- астыңғы сызық белгиси | .

**Д. Корсеткиш белгилер** | \$ -доллар | # -тор | ^ - тильда | @ - коммерциялық күшикше |.

**Е. 32** үлкен хәм киши орыс хәриплери, түсиник сөзлерди жазыўда пайдаланылады

|А|Б|В|Г|Д|Е|Ж|З|И|Й|К|Л|М|Н|О|П|Р|С|Т|У|Ф|Х|Ц|Ч|Ш|Щ|Ы|Ь|Э|Ю|Я|,

|а|б|в|г|д|е|ж|з|и|й|к|л|м|н|о|п|р|с|т|у|ф|х|ц|ч|ш|щ|ы|ь|э|ю|я|.

### § 1.2. Программаластырыў тилиниң анықламалары хәм сөзлиги

АТлерде **сөз**- бул, алфавит белгилери избе-излигиниң бөлинбей жазылыўы хәм анық мәниге ийе сөзлер. Сөзлер атқаратуғын ұазыйпасына байланыслы “**хызметши**“ хәм “**атама**“ болып екиге бөлинеди.

**Хызметши сөзлер.** Хызметши сөзлердиң атқаратуғын ұазыйпасы белгиле болған, АТ системасы курамына киритилген сөзлер. АТ лерде хызметши сөзлер оператор - деп аталады

(1-кесте). Операторлар орынланыўшы хәм орынланбайтуғын болып екиге бөлинеди.

Турбо-Паскаль тилиниң хызметши сөзлери 1-кесте

Absolute	Else	Interrupt	Procedur e	Unit
And	End	In	Program	Until
Array	External	Implementation	Record	Uses
Begin	File	Label	Repeat	Var
Case	For	Mod	Set	Virtual
Const	Forward	Nil	Shl	While
Constructor	Function	Not	Shr	With
Destructor	Goto	Object	String	Xor
Div	If	Of	Then	
Do	Inline	Or	To	
Downto	Interfase	Packed	Type	

**Орынланыўшы операторлар** - мәселени шешиў алгоритмине сәйкес программада операциялардың орынланыўын әмелге асырады (меншиклестириў, шәртли хәм шәртсиз өтиў, такрарлаў операторлары х.т.б.).

**Орынланбайтуғын операторлар** - берилген мағлыўматлардың түрлерин, қәсийетлерин анықлаўда ямаса дағазалаўда пайдаланылады (мысалы, массив операторы, түсиник сөзлерди жазыў, өзгериўши түрлерин дағазалаў хәм тамамлаў операторлары х.т.б.).

**Атама** - бул, ТП тилинде объектлерге ат койыўда латын хәриплери, цифрлар хәм астынғы сызық белгиси избе-излигинен дүзилген сөз. Атамалар “стандарт” хәм программа дүзиўши тәрәпинен койылатуғын “пайдаланыўшы” болып екиге бөлинеди. Пайдаланыўшы атама АТ лерде идентификатор - деп айтылады.

**Стандарт атамалар.** Стандарт атамалар тилди математикалық тәмийинлеў система курамына киритилген (2 - кесте). Олар мағлыўматлар үстинде операцияларды орынлаўда пайдаланылады. Мысалы, **Arctan(x)** стандарт атама берилген мүйештиң арктангенс функция мәнисин есаплайды.

**Турбо-Паскаль тилинде стандарт атамалары 2-кесте**

<b>Abs</b>	<b>False</b>	<b>Odd</b>	<b>Real</b>	<b>True</b>
<b>Arctan</b>	<b>Get</b>	<b>Old</b>	<b>Reset</b>	<b>Trunc</b>
<b>Boolean</b>	<b>Char</b>	<b>Output</b>	<b>Rewrrite</b>	<b>Unpack</b>
<b>Char</b>	<b>Cos</b>	<b>Pack</b>	<b>Round</b>	<b>Write</b>
<b>Cos</b>	<b>Dispose</b>	<b>Page</b>	<b>Sin</b>	<b>Writeln</b>
<b>Dispose</b>	<b>Input</b>	<b>Pred</b>	<b>Sqrt</b>	
<b>Eof</b>	<b>Integer</b>	<b>Put</b>	<b>Sqr</b>	
<b>Eoln</b>	<b>Maxint</b>	<b>Read</b>	<b>Succ</b>	
<b>Exp</b>	<b>New</b>	<b>Readln</b>	<b>Text</b>	

**Пайдаланыўшы атамалар- идентификаторлар.** Программа дүзиўши тәрәпинен программада пайдаланылатуғын объектлерге: таңба, өзгериўши атлары, программа аты, процедура хәм функция операторларының хәрбирине өз алдына ат койылады.

**Идентификаторларға (атамаларға) ат койыў хәм оны жазыў қағыйдалары:**

1. Объектлерге ат койыўда **26** латын хәриплери, астынғы сызық белгиси “\_” хәм **10** араб цифрлары пайдаланылады;

2. Атама латын хәрибинен ямаса астынғы сызық белгиси менен басланылады;

3. Таңбалар цифрдан ямаса хәрип пенен басланады: **121, A2, XYZ44, 333** ;

4. Атамаларды жазыўда екинши символдан баслап хәриплер ямаса цифрлар избе-излигин хәм астынғы сызық белгисин пайдаланыўға болады: **Sum\_S, \_X\_y\_1, Sabc, V\_KoLem, S\_Maidan, X123, X\_1\_Y\_2** ;

5. Объектлерге ат койыўда бос орын, үтир, ноқат хәм басқа арнаўлы белгилер пайдаланылмайды: **X+A, A\$34, S\_@%^, P(X)**;

6. Атамалардың ең үлкен максималъ узынлығын **127** символға дейин пайдаланыўға болады, соның **63** символы мәнили болады. ТП тилинде атамаларға **1** ден **8** ге шекемги символларды пайдаланыў мақсетге муўапық. Мысалы, **NUMBERODAYSINWEEK** (хәптеде кунлер саны ) узын атама болып, ол программада бир неше мәртебе пайдаланылса, онда қәтеликке жол койылыўы мүмкин. Сонлықтан, оны **NODAYS** атыменен белгилеп алыў орынлы болады.

7.. Объектлерге сәйкеслендирип ат койыў, программаны түсиникли оқыўға мүмкиншилик жаратады. Мысалы, **L** (узынлық), **T** (ўақыт), **V** (көлем), **S** (майдан), **Sum** (қосынды) х.т.б. хәриплер менен ямаса одан басланған атамалар менен белгилеп алынады. Шамалардың ең үлкен ямаса ең киши мәнислерин **MAXAB, MINXY** деп белгилеп алыў орынлы болады.

8. Хызметши сөзлерди (1-кесте) пайдаланыўшы атама ретинде қолланыў қадаған етиледі: **IF,CASE,THEN, RESET** х.т.б.

9. Үлкен хәм киши хәриплер менен белгиленген объектлер, мысалы **PasCAL, PASCAL, paSCal, paskal** ушын компьютер ядында бир орында “кетекшелер” ажыратылады.

10. ТП тилинде “\_” (астынғы сызық) объектлерди программада түсиникли белгилеп алыўға мүмкиншилик жаратады. Мысалы, **Max\_XY, Metod\_NEWTON, ALGL\_OPTIM** х.т.б.

11. Программа курамында (бас программада, функция ямаса процедура операторларында) айырым объектлер ушын бирдей атамаларды пайдаланыўға болады. Бул объектлер пайдаланбастан алдын, олар ушын ажыратылған кетекшелерди **<атама >:=0**; операторы жәрдеминде тазалап алыўымыз тийисли. Әмелий мәселелерди шешиўде бул усыл дерлик пайдаланылмайды.

**Атамалардың дурыс жазылыўына мысаллар:**

**Summa, FUNC, METHOD\_GAUSS, Prog1, \_REGISTERa, Sabs, V\_Sabsd, AlfaA, BetaB, Asw00q, FUNCX, METKA1, Prog1\_Sum, R\_REGISTER, S1abs, V\_absd, AaA, BaB.**

**Объектлерди төмендегише белгилеп алыўлар қәте :**

**Max XY** { **Max** хәм **XY** арасында бос орын қалдырылынған };

**X+Y; A\*2; BX1/7** { арифметикалық әмел пайдаланылған };

**%TUR; A?X; X%11** { паскаль алфавитине тийисли емес белгилер койылған };

**5SUM, Baz.2000** { цифр хәм арнаўлы белги “.” койылған };

**QARAQALPAKMA'MLEKETLIKUNIVERSITETI\_1976\_2002** { жүдә узын атама }.

### § 1.3. Турақлылар хәм өзгеріушілер

Объектлер мәніси ямаса мағлыұматлардың берилиуі турақлы хәм өзгеріуші шамалар болып бөлінеді. Объект мәніслери компьютер ядында, кетекшелерде жайласады. Программда, керекли ұақтында усы кетекше адресинен мағлыұматлар алып оқылады ямаса ол жерге қайтадан жазылады.

**Турақлылар.** Программа орынланыуында мәніслери өзгермейтуғын шамалар турақлылар-деп аталады. Турақлы шамалар программда **CONST** (турақлы\ константа) хызметши сөзи менен дағазаланылады:

**Жазылыу үлгиси :** **Const** <атама> = <турақлы мәніси>;

ТП тилинде айрым турақлы шамалар программда жазылыуы бойышша пайдала-нылады (3-кесте).

Турақлы шамалар		3 - кесте
Турақлылар	Түри	Мәніси
True	Boolean	Шын \ Истина
False	Boolean	Өтирик \ Лож
MaxInt	Integer	32676
Pi	Real	3.1415926 ...

**Const** сөзи жәрдемінде бир ямаса бирнеше турақлы шамаларды тәриплеуіге болады. Олар бири екіншисинен ноқатлы үтир менен бөлінип жазылады.

Мысал: **Const** MaxXY = 255; { бир турақлы шама }  
**Const** N=100; ITER=3000; { пүтин санлар }  
**E=1.0E-15; X1=5.908583;** { хақыйқый санлар }  
**BUK1='Г'; BUK2='В'; BUK3='М';** { литерли турақлылар }  
**LOG1=TRUE; LOG2=FALSE;** { логикалық санлар }

**Өзгеріушілер.** Өзгеріуші - деп, программа орынланыуыда мәніслери өзгеретуғын шамаларға айтылады. Турақлы шамаларды **Const** сөзи менен ТП компиляторы сәйкес түри бойынша ажыратып алады. Ал, өзгеріуші шамалар өзлестиретуғын мәнісиниң түрине сәйкес дағазаланылады. Өзгеріуші шамаларды дағазалауға **VAR**(Variable - өзгеріуші-лер\переменные) хызметши сөзи пайдаланылады :

**Жазылыу үлгиси:** **Var** <атама (лар)> : <мағлыұмат түри>;

Программаның **Var** бөлімінде мағлыұматлардың берилиуі түрлерине байланыслы өзгеріуші шамалар ноқатлы үтир менен бөлінип өз алдына дағазаланады. Бир түрдеги мәніслерге ийе өзгеріушілер бири екіншисинен үтир менен бөлінип жазылып дағазаланады.

Мысал: **Var** I, J, N1, Mmax, Mmin, ITER : Integer;  
X, Y, Z, Func, Summa, INTEGRAL : Real;  
L, L1, Bool1, Log1, Log2 : Boolean;  
CH, LITER, CH1, TextA : Char;

**Const** сөзи арқалы турақлы шаманы өзгермели түрде дағазалауға болады.

**Жазылыу үлгиси :** **Const** < атама > : < түри >=< мәніси >;

Мысал : **Const** Epsilon : REAL = 0.000001 ;  
**A : Array [ 1 .. 5 ] Of Integer = ( 1, -7, -85, 0, 17 );**  
**B : Array [ 1 .. 3 , 1 .. 4 ] Of Real = (( 1, -7, -85), (9.7, 0, 17),**  
**( 1.6, -7.4, -8.5), (9.5, 0.7, 1.7 ));**

бул жерде өзгермели турақлы атамасы дәслеп оның түринен кейин берилген мәністи (лерди) өзлестиреди хәм усы атамалы шамалар программда өзгеріуші шама түрінде пайдала-нылады.

**Ескертуі.** ТП тилинде хәрбир объект атына байланыслы мағлыұматлар, аралық есаплаулар хәм алынатуғын ақырғы нәтиже, қандай мәніслерди өзлестиретуғынлығы программа дүзиуші тәрепинен алдын-ала анықланылады. Олар **Const** , **Var** сөзлери арқалы сәйкес мәніслериниң берилиуі түрлери бойынша дағазаланған болыулары тийисли.

### §1.4. Скаляр мағлыұматлардың берилиуі түрлери

Хәрбир объект мәніси мағлыұматлардың берилиуіниң шекленген көплигинде анықланылады. Мағлыұмат түрлери **эпиуайы** (скаляр) хәм **қурамалы** болып бөлінеді. Скаляр мағлыұматлар түрлери “стандарт“, “санақлы“ хәм “шегереланған аралықта “ берилиуілери мүмкин. Программда объектлерди ұлыұмаластырып дағазалауға **TYPE**(түр\тип) хызметши сөзи пайдаланылады.

**Жазылыу үлгиси :** **Type** < түр аты > = < мағлыұмат түри >;

Мысал : **Type** COLOR4 = (Red, Yellow, Green, Blue) ;  
**Metal = (Fe,Ga,Na,Gu,Zn);**

**Стандарт скаляр мағлыұматлардың берилиуі түри.** Олардың топарына, пүтин санлар (**INTEGER**), хақыйқый санлар (**REAL**); символлы ямаса литерли мағлыұматлар (**CHAR**), Буль - логикалық санлар (**BOOLEAN**) хәм көрсеткиш мағлыұматлар (**POINTER**) түрлери киреди. Пүтин түрдеги мағлыұматлар

онлық хәм оналтылық (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F ) санақ системаларында бериледи. Оналтылық санның алдына “\$ (доллор)” белгиси қойылып, ( \$0000 .. \$FFFF ) аралықта бериледи.

Онлық системада берилген ҳақыйқый санлар: өзгермес ноқатлы хәм жылжымалы ноқатлы үлгилерде жазылады. Өзгермес ноқатлы ҳақыйқый санларды жазыўда, санның пүтин хәм бөлшек үлеси “ (ноқат)” арқалы ажыратылынады. Егерде, санларды жазыўда ноқат белгиси қойылмаса, олар пүтин сан болады. Санлар алдынада “ + “ хәм “ - “ белгилери, олардың оң ямаса терис санлар екенлигин анықлайды. Оң санлар ушын “ + “ белгисин қойыў шәрт емес, ал терис санларда “ - “ белгиси қойылыўы шәртли.

Пүтин санлар : -3451 , +567, -35, 6098, -89, 908, 2001, \$AF45 , \$35AC, \$89F.

Ҳақыйқый санлар: Өзгермес ноқатлы 0.98, -34.0067, 4568.9583206, -120.98.

Жылжымалы ноқатлы ҳақыйқый санлар: &mE±p экспоненциал түрде жазылады. Бул жерде m санның мантиссасы , E = 10 хәм p санның тәртиби ( бир, еки, үш ямаса төрт қаналы санлар ), & - “ + “ ямаса “ - “ белгиси.

Мысал :  $0.98 = 0.98E+00 = 98.0E-02 = 0.00098E+3 = 0.0000098E+05 ;$   
 $-34.0067 = -3.40067E+01 = -34006.7E-03 = -0.340067E+2 = -0.00340067E+4 ;$   
 $0.000001 = 0.1E-05 = 1.0E-6 = 10.0E-7 = 0.000000001E+03 .$

Пүтин санлы мағлыўматлар түрин дағазалаў. Пүтин санлы мағлыўматларды өзлестиретуғын объектлерди Type, Var сөзлери менен дағазалаўда мәнислериниң берилиў аралығы есабға алынады (4 - кесте) .

Пүтин санлы мағлыўматлар түри 4-кесте

Түри	Мәнислер аралығы	Яд олшеми ( байт )
Shortint	-128 .. 127	1
Integer	-32768 .. 32767	2
Longint	-2147483648 .. 2147483648	4
Byte	0 .. 255	1
Word	0 .. 65255	2

Мысал: Var X1,Y1, Z1 : Shortint; Iter1, I, J, K : Integer;  
 WORD1, BAIT1 : Word; Bukba, Nomer : Byte;  
 Var DX1,DY1, DZ1 : Longind;

Ҳақыйқый санлы мағлыўматлар түрин дағазалаў. Егерде арифметикалық аңлатпаларда операндлардың биреўи ҳақыйқый сан болса, арифметикалық әмеллердиң (+, -, \*, /) нәтийжесиде ҳақыйқый сан болады (5-кесте).

Ҳақыйқый санлы мағлыўматлар түри 5 - кесте

Түри	Мәнислер аралығы (разрядлар)	Яд олшеми ( байт )
Real	2.9E-39 .. 1.7E38 ( 11-12)	6
Doubly	5.0E-326 .. 1.7E308 ( 15-16)	8
Extended	3.4E-4932 .. 1.1E4932 ( 19-20)	10
Comp	- 9.2E18 .. 9.2E18 ( 19-20)	8
Single	1.5E-45 .. 3.4E38 ( 7-8 )	4

Longint, Single, Double, Extended хәм Comp хызметши сөзлери жәрдеминде объектлерди дағазалаў ушын компьютер 8087 сопроцессоры менен тәмийнленилген хәм оған { \$N+ } директивасы жалғастырылынған болыўы тийисли.

Мысал : Var R1, F1, S1 : Real; Xmax, I, J, K : Single;  
 D1, BT1 : Double; B1, NV1 : Extended;  
 Var DX1, DY1, DZ1 : Comp;

Ескертиў. Програмада киши мәнисге ийе болған өзгериўшилерди ямаса турақлыларды дағазалаўда Shortint, Byte, Word, Integer, Comp, Real хызметши сөзлери пайдаланылады. Компьютерде яд көлеми тежеледи.

Символлы (литерли) мағлыўматлар түри. Олар ASCII код кестесиниң манислери коплиги менен аныкланылады ( 6 - кесте) .

Символлы мағлыўмат түри 6 - кесте

Түри	Мәнис	Яд олшеми(байт)
Char	ASC II код кестеси символлары	1

Мысал : VAR Ch1, BukA, Liter1 : Char ; Xch, FB, ALit : Char ;

Програмада символлы мағлыўматлар мәнислерин меншиклестиргенде, олар тырнақшаға ‘символ’ алынып жазылады ямаса код номерлери алдына ‘#’ ( тор белгиси) қойылады. Белгиниң ASCII кестесинде жайласыў орны сол белгиниң код номерин анықлайды.

Мысал : | ‘+’ | ‘F’ | ‘Я’ | ‘Ы’ | ‘A’ | ‘a’ | ‘@’ | ,

**Логикалық мағлыұматлар түри.** Логикалық мағлыұматлар **TRUE (Шын)** хәм **FALSE (Өтирик)** логикалық санлар менен анықланылады (7-кесте).

**Логикалық мағлыұмат түри 7 - кесте**

Түри	Мәниси	Яд өлшеми(байт)
<b>Boolean</b>	<b>True \ False</b>	<b>1</b>

Мысалы : **Var LOG1, BOOLA2, L, B : Boolean;**  
**ZL1, HBL2, L1, B1 : Boolean;**

**Мағлыұматлардың қурамалы түрлері.** Мағлыұматлардың қурамалы түрлері скаляр мағлыұматлар түрлерінің тәртіплескен көплигінен дүзіледі. Паскаль тилинде тийкарынан төмендеги қурамалы мағлыұмат түрлері пайдаланылады: массивлер (§ 5.1.), қатарлық (§ 5.3.); көплик (§ 5.4.); жазыу (§ 5.5.); көрсеткіш (§ 5.6.); файл (6- бөлім).

### § 1.5. Мағлыұматлардың санақлы хәм шегераланған аралықта берилиуі

Мағлыұматлардың стандарт скаляр түрлерде (§1.4) берилиуін менен бирлікте ТП тилинде мағлыұматлар түрін программа дүзиуінің өзи анықлауы мүмкін. Оларға санақлы хәм шегералық аралықта түрлердеги мағлыұматлар киреди. Олар компьютер ядында 1- байтлы орын ийелейди хәм элементлер саны **256** дан үлкен болмайды.

**Мағлыұматлардың санақлы түрде берилиуі.** Санақлы түрде берілген мағлыұматларда өзгеріушілердің өзлестиретуғын мәнислері саны белгили болып “ элементлер саны  $\leq 256$ ”, бири екіншисинен үтир арқалы айырылып жай қаўсырмаға алынып жазылады.

**Жазылыу үлгиси:** 1. **Type<түр аты> = (1-мәнис, 2-мәнис, ... N- мәнис );**

**Var <1 - атама, 2 - атама, ... > : < түр аты >;**

2. **Var <1 -атама, 2 -атама,...> : (1-мәнис, 2-мәнис, ... N- мәнис );**

**Мысал:** **Type COLOR4 = (Red, Yellow, Green, Blue) ;**

**Metal = (Fe, Ga, Na, Gu, Zn); TypI = Integer; TypR = Real;**

**Var Ac, Bc, Cc, Dc : COLOR4;**

**M1, M2, M3 : Metal;**

**Season = (Winter, Spring, Summer, Autumn);**

**I, N, L, K : TypI; X, Y, Z, R : TypR;**

бул жерде **COLOR4** түр аты менен төрт рең **Red** (қызыл), **Yellow** (ашық қоңыр), **Green** (жасыл), **Blue** (көк) мағлыұматлардың санақлы түрде берилиуі дағазаланған, онда **Ac, Bc, Cc, Dc** атамалы өзгеріушілер усы реңлердің биреуін меншиклестиреди. **M1, M2, M3** - атамалы өзгеріушілер, **Metal** түріменен баянланылған санақлы түрдеги **Fe, Ga, Na, Gu, Zn** - элементлердің биреуін өзлестиреди. Екіншіден, мағлыұматлардың санақлы түри **Var** бөлімінде **Season** өзгеріуші атыменен **Winter** (қыс), **Spring** (бәхәр), **Summer** (жаз), **Autumn** (гүз) мәнислердің биреуін меншиклестирди. Ал, **TypI** түр аты менен **I, N, L, K** пүтин санлы өзгеріушілер, **TypR** түр аты менен **X, Y, Z, R** хақыйқый санлы өзгеріушілер дағазаланған.

Санақлы түрдеги мағлыұматларды **Case** операторында (§3.3.2) селектор мәниси ушын пайдаланыуға болады. Салыстырыу хәм логикалық әмеллер бирдей түрде дағазаланған санақлы мағлыұматлар үстинде орынланылады. Санақлы түрлердің элементтері жайласыу тәртіби менен анықланылады. Мысалы, **Winter < Spring** аңлатпа мәниси **True** болады, себеби **Spring** элементи жайласыу тәртіби бойынша, баянлауда **Winter** ден кейин жазылған. Санақлы мағлыұматларды **Succ, Pred, Ord** стандарт функциялардың аргументі ушын пайдаланыуға болады (§ 4.8.2., § 4.8.3.).

**Ескертиу.** Санақлы түрлерде берілген мағлыұматларда оқытыу-шығарыу операторлары орынланбайды. Олардың мәниси мәнислеу операторы жәрдемінде киритилинеди.

**Мағлыұматлардың шегераланған аралықта берилиуі.** Мағлыұматлардың шегераланған аралық түрде берилиуінде объектлердің мәниси төменги хәм жоқары шегаралары бойынша еки турақлы арасындағы шамалар менен анықланылады. ТП компиляторы мәниси аралықта берілген өзгеріуші менен хәрбир әмел орынланғанда өзгеріушінің мәниси аралық ишинде бар ямаса жоқ екенлиги тексерип барады. Төменги хәм жоқары шегараны анықлаушы турақлылар бирдей түрде берілген болады (хақыйқый түр **Real** мағлыұматларды аралықта берилиуінде пайдаланбайды). Биринши турақлының мәниси екіншіден киши болады.

**Жазылыу үлгиси :** 1. **Type<түр аты> = < 1- турақлы > .. < 2- турақлы >;**

**Var <1 - атама, 2 - атама, ... > : < түр аты >;**

2. **Var <1 -атама, 2 -атама,...> : <1- турақлы > .. <2-турақлы >;**

**Мысалы:** **Type DAY = 1 .. 31;**

**Var Di, Df, Md : DAY;**

**ApuaiSan : 2 .. 500;**

Бул мысалда **Di, Df, Md** өзгериушилердин мәнислери **DAY** шегералық түри бойынша анықланып хәм **1 .. 31** аралықта берилген кәлеген бир санды өзлестиреди. Ал, **ApuaiSan** идентификаторы **2** менен **500** арасындағы кәлеген бир санға ямаса эпиуайы санлардың биреуине тең болыуы мүмкин. Аралықтың шегараларын **Const** аркалы алдын-ала беріуге болады.

Мысал: **Const MIN = 127; MAX = 336;**  
**Type Min\_Max = -MIN .. MAX;**  
**Var X, Y, Z : Min\_Max;**

**Ескертиу:** Өзгериушиниң, санақлы хәм аралықта берилиуинде мағлыуатлар курамында болмаған кәлеген басқа мәнисти меншиклестириуи программаны үзилiske түсиреди.

### § 1 6. Мағлыуатлардың берилиуиниң бирдейлиги хәм сәйкеслиги

Программада аңлатпада пайдаланылған операндлар түриниң бирдейлигин ямаса мәнислериниң сәйкеслигин есабға алыу айрықша аҳмийетке ийе. Еки түр бирдейлик бола алады, егерде олар бирдей түрде тәрипленген болса ямаса оларды дағазалауда бирдей түрдеги өзгериушилер (идентификаторлар) атамалары пайдаланылған болса.

Мысал: **Type MAS1, MAS2 = Array [ 1 .. 10, 1 .. 15] Of Real;**

бул жерде **MAS1, MAS2** - атамалы өзгериушилер берилиу түри бойынша бирдейлик бола алады, себеби мағлыуатлар компонент түри бирдей.

Мысал: **Var X1 : Set Of Byte;**  
**Y1 : Set Of Integer;**

бул жерде **X1, Y1** атамалы өзгериушилер бирдейлик бола алмайды, себеби хәр бир өзгериуши өз алдына дағазаланған. Олар белгили аралықтағы пүтин санларды өзлестиреди.

Объект түрлериниң сәйкеслиги, процедура хәм функция операторларын орынлауда ҳақыйқый хәм жалған (формаль) параметрлерлердин мәнислериниң сәйкес келиулері зәрүрли. Мағлыуатлар түрлериниң сәйкеслиги аңлатпаларды есаплауда, салыстырыу эмеллеринде хәм мәнислеу операторында есабға алынады.

1. Еки түрде берилген мағлыуатларды салыстырыуда, сәйкеслик орынланылады, егерде олар арасында төмендеги шәртлердин ең кеминде биреуи орынланса : - еки берилген түр бирдей болса; - еки түрде ҳақыйқый санлар болса; - еки түрде пүтин санлар болса; - бир түр екинши түрдин ишки аралығына жайласған болса; - еки түр тийкарғы түрдин берилиуи аралығы ишиндеги бар болса; - еки түр көплик түрде берилген болып хәм тийкарғы (базалық) түр менен сайкес келсе; - еки түрде бирдей сандағы бирикпелер (компонентлер) менен берилген қатарлық мағлыуатлар болса; - биринши түр қатарлық болып, ал екиншиси қатарлық ямаса символлы түрлер болса; - биринши түр көрсеткишли түр болып, ал екиншиси кәлеген түрдеги көрсеткишли мағлыуатлар болса. Мысалы, **If 'F' = 'X' Then ...;** операторы орынланылады, себеби еки түрде **Char** хызметши сөз менен тәрипленеген. Ал, **If 'B' > 33 Then ... ;** операторы орынланбайды, себеби биринши символлы **Char**, ал екиншиси пүтин сан **Integer** түрге тийисли.

2. Өзлестириу операторында **":="**, еки мағлыуатлар түри арасында сәйкеслик орынланылады, егерде төмендеги шәртлердин ең болмағанда биреуи орынланса: -еки түрде мәнислери бойынша сәйкес келсе ( файллы ямаса курамалы мағлыуатлар түрлеринен басқа); - файл компонентлери курамы жазылыуы бойынша бирдей түрде болса; - еки түрде скаляр түрлерде берилген болса, яғный екинши түрдин мәниси бириншиниң мәнисиниң аралығына тийисли болса; - еки түрде ҳақыйқый түрде берилген болса хәм екинши түрдин мәниси бириншиниң мүмкин болған мәнислери аралығында жайласған болса; - биринши түр ҳақыйқый болып, ал екинши түр пүтин сан болса; - еки түрде қатарлық болса; - биринши түр қатарлық болып, ал екинши символлы (литерли) түрде болса; - еки түрде сәйкес көплик түрге дерек болса, хәм екиншиниң барлық ағзалары, биринши түрдин мүмкин болған мәнислериниң аралығына дерек болса; - еки түрде сәйкес көрсеткишли түрде берилген болса.

Мысал: **Var AR, B : Integer ;**  
**X, Y: Real ; Ch1, Sim : Char ;**  
**Lg1, Lg2, B1 : Boolean;**  
**BEGIN B:=X; B1:=B ; { өзгериушилер сәйкес емес, орынланбайды }**  
**Y:=AR-X; Lg1:= Lg2; { орынланады }**  
**Ch1:=X+Y ; X:=B1 { өзгериушилер сәйкес емес, орынланбайды } .**  
**L2:=B1 ; B1:= Lg1 OR L2 ; { орынланады } END .**

### §1.7. Турбо-Паскаль тилинде программа дүзилиси

Программа - деп, шекли адымда орынланылатуғын операция хәм операторлардың орынланыу избе-излигин бирлестиретуғын, мәселени шешіу алгоритми тийкарында дүзилген инструкцияға айтылады. Программаны дүзиуде алынған нәтийженің дәллиги, орынланыуға кеткен уақыт хәм талап етилетуғын яд көлеми есабға алынады.

Программада ҳәрбир қатар узынлығы 127 белгиден үлкен болмайды. Егерде, одан үлкен болса, ТП компиляторы артығын алып таслайды. Программалар тексти ТП орталығы курамындағы текст редакторы жәрдеминде жазылады.

Программаның жазылыуы синтаксиси басланыуы көринистен хәм блоктардан турады. Блоктар баянлаушы хәм орынланыушы болып бөлинеди. Блок ишинде бирнеше блоктар болыуы мүмкин. Курамында басқа блоктар болмаған **блок-глобаль (бас) блок**, ал басқа блоктар **локаль (толықтырыушы) блоктар** деп айтылады.

Глобаль блок қәлеген программада бар болады. Локаль блоктар, процедура хәм функция операторлары алгоритмге сәйкес, зәрүрликлери бойынша пайдаланылады. Программада пайдаланылатуғын объектлер (мағлыұмат түрлери, турақлылар, өзгериушилер х.т.б.) блоктарға сәйкес глобаль хәм локаль деп айтылады. Объектлер қайсы блокта дағазаланған болса, сол блокта пайдаланылады. ТП компиляторы тәрәпинен программа текстиндеги пунктуация, синтаксислик кәтеликлер хәм семантикалық қағыйдалары тексериледи.

Турбо - Паскаль тилинде программаның ұлыуыма дүзилиси :

1. **Program** < аты > [(INPUT,OUTPUT)]; - программа басы;
  2. **Uses** < модуль (лер), ... >; - китабханалық модуллер;
  3. **Label** < таңба (лар), ... >; - таңбаларды дағазалау;
  4. **Const** < турақлы шама (лар), ... >; - турақлы шамалар;
  5. **Type** < түр аты >; - мағлыұмат түрин анықлау;
  6. **Var** < өзгериуши шама (лар) , ... >; - өзгериушилерди дағазалау;
  7. **Procedure** < аты > [< параметр (лер),...>]; - процедуралық бирлик;
- Begin** < процедура денеси > **End**;
8. **Function** < аты > < параметр (лер) >; - функция операторы;
  - Begin** < функция денеси > **End**;
  9. **BEGIN** < операторлар > **END**. - бас (глобаль) программа;
  10. (\* түсиник сөзлер \*) ямаса { тусиник созлер }.

Мәселени шешіуі алгоритмине сәйкес, бул бөлимлердиң айрымлары программада зәрүрлиги бойынша пайдаланылады. **Uses** бөлими, программа басламасынан кейин жазылыуы шәртли. Басқа қәлеген бөлим программаның тәриплеу бөлимінде бирнеше мәртебе жазылыуы хәм олардың жазылыуы избе-излиги қәлеген тәртипте жайласыуы мүмкин. Программада тийқарғы еске алатуғын бас мәселе, онда пайдаланылатуғын таңбалар, барлық объектлер меншиклестиретуғын мағлыұмат турлерине сәйкес дағазаланған болыулары тийисли.

**I. Программа PROGRAM** < аты > хәм параметрлер **INPUT, OUTPUT** хызметши сөзлери менен басланады. ТП да буларды жазбауға болады. Жазылыуы шәртли болмаған параметрлер **Input** хәм **Output** стандарт файллар болып, олар программаның орынланыуында ТП орталығын компьютерде операцион система арқалы киритиу-шығарыу курылмалары менен байланыстырады.

**Жазылыуы үлгиси :** **PROGRAM** < аты > [ (INPUT,OUTPUT) ] ;

Мысал : **PROGRAM** **METHOD\_MONTE\_CORLO**(INPUT,OUTPUT);

**PROGRAM** **PrimARRAY**; **PROGRAM** **Method\_Newton**;

**II. USES** - бөлими. **Uses** (модуль) хызметши сөзиниң ұазыйпасы ТП орталығына киритилинген программалар комплексинде бар системалық китапханалық модуллерди “ **CRT; DOS; GRAPH; PRINTER; OVERLAY**; “ хәм программа дүзиушилер тәрәпинен дүзилген жеке-китапханалық модуллерди жаңа программада шақырып алып пайдаланыуы ушын қолланылады:

**Жазылыуы үлгиси :** **USES** < 1 -модуль>, < 2 -модуль>, ... ;

Мысал : **Uses** **DOS, CRT, GRAPH**;

**Uses** **MYLIB, MORTIM, CRT**;

**III. LABEL** - бөлими. Мәселе алгоритмине сәйкес программада қәлеген оператор алдына таңба қойыуға болады. Бул программада қәлеген орыннан, усы таңба қойылған операторға шәртсиз өтиу операторы “**GOTO** <таңба>“ арқалы өтиуди әмелге асырады. Таңбаларды тәриплеу **LABEL** (таңба \ метка) хызметши сөзи менен басланып хәм одан кейин таңба атлары үтирлер арқалы айырылып жазылады. Соңғы таңбадан кейин ноқатлы үтир қойылады. Программа ишинде пайдаланылатуғын барлық таңбалар **Label** бөлимінде көрсетилиниуи шәртли. Таңбалар аты цифрлар хәм хәриплерден ямаса оларды бириктириу арқалы қойылады.

**Жазылыуы үлгиси :** **Label** < 1- таңба >, < 2- таңба >, ... ;

Мысал : **LABEL** **МЕТКА1, ТАҢБА2, 131, BLOK1**;

**Label** **Met231, TAA2, Sabs3, Bf12**;

Таңба менен оператор қос ноқат арқалы бөлинип жазылады :

Мысал : **MET23 : UX :=A\*X + Cos(X\*Y-A); 13: If D<0 Then Writeln( ‘ Шешим жок’ );**

Таңба белгисиниң ең үлкен узынлығын 127 символға шекем болады.

**IV. CONST** - бөлими. Бул бөлимінде турақлы шамалардың мәнислери меншиклестири-леди (§ 1.3.).

Мысал : **Const Max :Integer = 100;**  
**G16 = \$12F; { оналтылык системадагы сан}**  
**INT = 166; RE1 =-12.09; LET = ‘Блок\_3’;**  
**Expr = 2.71828; Epsilon =1.0E-09;**

**V. TYPE - бөлими.** Объекттердің қайсы түрдегі мағлыұматларды өзлестіріуі **Type** хызметші сөзі улыұмаластырылып дағазаланады хәм басқа объекттер оның менен байлан-ыстырылады.

Мысал : **TYPE Matr = Array [1..10,1..10] Of Integer;**  
**LBUKVA = ( ‘A’ .. ‘Z’ ); San = 1..100;**

**Var A, B, C : Matr ; Ch1, Sim1 : LBUKVA; X1,Z1 : San;**

**VI. VAR (VARible)- бөлими.** Бул бөлімде программада пайдаланыушы хәрбір атама (идентификатор) өзлеринің меншиклестіретуғын мәніслерине байланыссы дағазаланған болады.

Мысал : **Type Kun = 1 .. 31; Napte =1 .. 7;**  
**АП =(‘Январь’, ‘Февраль’, ‘Март’, ‘Апрель’, ‘Май’, ‘Июнь’, ‘Июль’, ‘Августь’,**  
**‘Сентябрь’, ‘Октябрь’, ‘Ноябрь’, ‘Декабрь’);**

**Int=Integer; Lit= Char;**

**Var I, J, K : Int; I1,J1, K1 : Integer;**

**L1, L2, L3 : Lit; B1, B2 : Char;**

**A1, A2, A3, A4 : Aii; X1, Y2, Z3 : Real;**

**K1, K2, K3, R4 : Kun; B1, B2, B3 : Boolean;**

**H1, H2, H3, H4 : Naptei;**

Мысалда, мағлыұматлардың берилиуі бойынша мағлыұматлар түрлері **Type** хәм **Var** бөлімлерінде дағазаланған. **Int** хәм **Lit** түр атлары жәрдемінде , қайтадан **Integer**, **Char** стандарт мағлыұматлар түрі дағазаланған, **Kun**, **Napte** бойынша шегераланған мағлыұматлар хәм **АП** менен санақлы мағлыұматлар дағазаланған, Бул түр атларына сәйкес хәм оларға қосымша объекттердің өзлестіріуі мүмкін болған басқа объект түрлері **Var** бөлімінде дағазаланған.

**VII-VIII. Функция хәм процедура операторларын шөлкемлестіріу бөлими.** Бул бөлімде жәрдемші программалар ( **FUNCTION**, **PROCEDURE** операторлары) өз алдына программалық бірлік түрінде жазылады (IV-бөлім).

**Жазылыуы үлгиси : 1. PROCEDURE < аты > [ (параметрлер,...)];**  
**< тәриплеу бөлими > BEGIN < процедура денеси > END;**  
**2. FUNCTION < аты > ( параметрлер ) : < түрі >;**  
**< тәриплеу бөлими > BEGIN < функция денеси > END;**

**IX. Бас программа бөлими.** Бас программа - программаның тийкарғы үлесі. Хәқыйқатында, усы бөлімде алдын-ала мағлыұматлар түрлері өзгеріушілер, турақлылар, функция мәніслері х.т.б. объекттер анықланылған болып хәм олар үстінде анық операциялар орынланыуы ақырғы нәтийжени береді. Операторлар бөлімі ”**BEGIN**” (**басланыу\начало**) хызметші сөзі менен басланып ”**END.**” (**ақыры\конец**) хызметші сөзі хәм ноқат пенен тамамланылады.

**Жазылыуы үлгиси : BEGIN < 1- оператор >;**  
**< 2- оператор >; ... END.**

Программада операторлар тил синтаксисине сәйкес қатаң тәртіпте избе-из орынланы-лады. **Begin** хәм **End** сөзлері арифметикалық аңлатпада ашылыушы хәм жабылыушы жай қаусырмалар хызметине сәйкес келеді.

**Ескертиуі.** Кейингі мысалларда **BEGIN** хәм **END** сөзлерін глобаль (бас) блоklar ушын, ал **Begin** хәм **End** сөзлерін локальлы (толықтырыушы) блоklar ушын пайдаланамыз.

**X. Түсиник сөзлер - бөлими.** Түсиник сөзлер (комментария) алгоритмге хәм операторларға программа ишінде түсиник беріуі ушын жазылатуғын сөзлер дизбеги. Буны программаның бос орын (пробель) қалдырып жазыуға болатуғын кәлеген орында жазыуға болады. Түсиник сөзлер дизбеги { **түсиник сөлер** } ямаса (\* ... \*) символлары арқалы шегараланған болады. Түсиник сөз ретінде **ASCII** кодына тийисли кәлеген белгилер комбинациясын пайдаланыу мүмкін.

Мысал : { **SATC ын GAUSS методы менен шешіу** }  
(\* **Транцендент тенлеме Ньютон методы менен шешіу** \*)

Шегаралаушы белгилер арасына бос орын қалдырыуға болмайды. Түсиник сөзлер ишінде екінші мәртебе түсиник сөзді жазыу ушын пайдаланылған белгини қолланыуға болмайды;

Мысал : { **Тапсырма {1}-сызыклы есаплау усылы** } - кате;  
(\* **Тапсырма {1}-сызыклы есаплау усылы** \*) - дурыс;  
{ **Мысал (\* 3 \*) цикллы есаплау усылы** } - дурыс.

Түсиник сөзлер узынлығына шегара жоқ. Компилятор түсиник сөзлерді итибарға алмайды, сонлықтан программаның орынланыуына тәсирин тийгизбейді. Бирақта хәр бір белги компьютер ядында 1 байтлы орын ийелейтуғын ядта сақлау керек. Түсиник сөзлер : -дүзилген программа мақсетин анықлауда; -

пайдаланылған өзгеріушілердің хәм турақлылардың мағанасын түсиндириўде; - программаның логикалық орынланыў избе-излигине түсиник бнриўде; - программада алгоритминиң қыйын түсинилетуғын орынларын түсиндириўде пайдаланылады. Түсиник сөзлерди орынлы пайдаланыў программада кететуғын кәтеликлерди жеңил табыўға хәм екинши адам программаны толық түсиниўге жәрдем етеди.

Программаны орынлаўда { .... } ямаса (\* ..... \*) белгилерин пайдаланып, ўақытша ярым операторлар топарын түсиник сөзлер ретинде қарап, программаның қалған бөлегин орышлап алыў ақырғы нәтийжени тез алыўға мүмкиншилик жаратады.

### § 1.8. Программаны жазыўда пунктуация қағыйдалары

Программада үтир ямаса ноқатлы үтирлердің қойылыўы тәбийғый халық тилиниң қағыйдасына уқсас. Тилге тийисли элементлердің избе - из тәртипте жазылыўында үтир белгиси қойылады **A, B, C, D, E** . Егерде бул элементлерди бир топарға бирлестирсек, олар қаўсырмаларға алынып үтир менен бөлинип (**A,B,C,D,E**) жазылады. Егерде топар элементлери екинши бир топар элементлери менен бириктирилинип ушинши топар дүзетгын болса, онда ол топарлар арасына үтирлер қойылып жазылады(**(A,B,C,D,E)**),(**X,Y,Z**), **U, (P,S,T)**).

ТП тилинде бундай жағдайларда жай қаўсырмалар орнында **BEGIN ... END** операторлары пайдаланылады. Үтирлер ноқатлы үтир менен, топар элементлери - операторлар менен алмастырылынады. (**(A,B,C,D,E)**),( **X,Y,Z** ), **U, (P,S,T)**) топарының ТП тилинде жазылыўы **BEGIN Begin < Операторлар: A; B; C; D; E > End;**

**Begin < Операторлар: X; Y; Z > End; < Оператор: U; >**

**Begin < Операторлар: P; S; T > End END; .**

Ноқатлы үтирди хәм үтирди программада пайдаланыў пунктуация қағыйдалары :

1.Тәриплеў бөлиминде ноқатлы үтир ямаса үтир **Unit, Uses, Label, Const, Type, Var** хызметши сөзлерден кейин қойылмайды. Бул сөзлер менен анықланған объектлер арасына үтир хәм ең соңына ноқатлы үтир қойылады;

2. Ноқатлы үтир **Begin** сөзинен соң қойылмайды. **End** сөзи алдындағы операторға қойыў шәрт емес;

3. Операторлар арасына ноқатлы үтир қойылып жазылыў шәртли ;

4. Ишки тақрарланыўшы (цикли) есаплаў процессинде **End** сөзи бирнеше мәртебе пайдаланылса, онда ноқатлы үтир ямаса ноқат ең соңғы **End** ге қойылады, яғный < оператор> **End End End End [.] [;]** .

5. Егерде **End** сөзи таңбадан соң жазылған болса, онда олардың арасына ноқатлы үтир қоймаўға болады.

Мысал: **< операторлар > ; Goto MET33;**

**MET33: End;**

6. Тақрарлаў операторлары **WHILE, REPEAT**, хәм **DO** сөзлеринен соң ноқатлы үтир қойылмайды. **Then, Else, Until** сөзлеринен алдын хәм соң ноқатлы үтир қойылмайды.

## II. АҢЛАТПАЛАР, ОПЕРАНДЛАР, ОПЕРАЦИЯЛАР, ФУНКЦИЯЛАР

**Арифметикалық аңлатпа** - бул, математикалық аңлатпаның операндлар жәрдемінде мағлыўматлар үстинде орынланылатуғын әмеллердің **AT** лерде жазылыўы. **Операндлар** - турақлы хәм өзгеріуши шамалар, массивлер, функциялар х.т.б. объектлер менен анықланылады. Мысалы, **Y:=FA \* X + B1 - Cos(X) - 56.9**; аңлатпасында **FA, X, B1** хәм **56.9** операндлар; “ + , - , \* “ әмел белгилери; **Cos(X)** операнды косинус функция мәнисин анықлайды. Қаўсырмалар аңлатпаларда ассоциативлик басқарыўды ямаса операциялардың орынланыў тәртибин өзгертиўде пайдаланылады.

ТП тилинде операциялар - арифметикалық, салыстырыў, логикалық (Буль), @ - адресди анықлаў (§5.6), қатарлық мағлыўматлар үстинде (§5.3), көпликлер үстинде (§5.4.) орынланылады. Операциялардың түрине сәйкес, аңлатпалар арифметикалық, салыстырыў, логикалық, қатарлық х.т.б. деп айтылады.

### § 2.1. Арифметикалық операциялар

**Арифметикалық әмеллер** - унарлы хәм бинарлы болып бөлинеди. Биринши жағдайда операция тек бир операндға тийисли болып хәм барлық ўақытта оның алдында жазылады. Екинши жағдайда, еки операнда арасында орынланылатуғын арифметикалық әмел белгиси жазылады.

Мысал : **Унарлы әмеллер** **Бинарлы әмеллер**  
 $-(-X) = X$  |  $+(-X) = -X$  **A + B** | **A - B**  
 $+(+X) = X$  |  $-(+X) = -X$  **A \* B** | **A / B**

Скаляр мағлыўматлар үстинде орынланылатуғын тийкарғы арифметикалық әмеллер (8-кесте );

Арифметикалық операциялар		8 - кесте	
Әмеллер	Мазмұны	Операнд	Нәтийже

	Бинарлы операциялар		
1. + (қосыў)	A + B	Integer Real	Integer Real
2. - (алыў)	A - B	Integer Real	Integer Real
3.*(көбейтiў)	A * B	Integer Real	Integer Real
4. / (бөлиў)	A / B	Integer Real	Integer Real
5. DIV	Тийинди пүтин үлеси	Integer Real	Integer
6. MOD	Тийинди бөлшек үлеси	Integer Real	Integer
7. SHL	Шепке жылжытыў	Integer	Integer
8. SHR	Оңға жылжытыў	Integer	Integer
9. OR	Бит бойынша қосыў	Integer	Integer
10. AND	Бит бойынша көбейтiў	Integer	Integer
11. XOR	Қосыўды бийкарлаў	Integer	Integer
12. NOT	Унарлы операциялар Бийкарлаў	Integer	Integer
13. +	Белгини сақлаў	Integer Real	Integer Real
14. -	Белгини өзгертiў	Integer Real	Integer Real

1-4. Әмеллер : <+(қосыў),- (алыў) , \* (көбейтiў), / (бөлиў)> арифметикалық әмеллердің орынланыўын әмелге асырады.

Мысал :  $X:=A+B/(C-D)$ ;  $Y1:= A/B/C$ ;  $\text{Sin}A:=1-\text{Cos}(A)*\text{Cos}(A)$ ;

5. **DIV** - операциясында операндлар мәніси алдыменен пүтин санға түрлендирип, бөлиў әмелин орынлайды. Нәтиже, тийиндинің пүтин үлесине тең болады. Егерде, бөлиніўши бөлиўшиден кише болса, нәтиже нолге тең болады:

$$\begin{array}{l} 13 \text{ Div } 5 = 2 \quad | \quad 33 \text{ Div } 5 = 6 \quad | \quad 6 \text{ Div } 9 = 0 \\ 30 \text{ Div } 5 = 6 \quad | \quad 9 \text{ Div } 9 = 1 \quad | \quad 18 \text{ Div } 9 = 2. \end{array}$$

6. **MOD** операциясы. Mod пенен операндлар мәніси алдыменен пүтин санға түрлендирилип бөлиў әмели орынланылады. Нәтижеде, тийиндинің қалдық үлесі пүтин сан түрінде алынады. Егерде, операндлар қалдықсыз бөлінсе, нәтиже нолге тең болады.

$$\begin{array}{l} 13 \text{ Mod } 5 = 3 \quad | \quad 6 \text{ Mod } 6 = 0 \quad | \quad 6 \text{ Mod } 5 = 1 \\ 30 \text{ Mod } 5 = 0 \quad | \quad 61 \text{ Mod } 6 = 1 \quad | \quad 16 \text{ Mod } 4 = 0. \end{array}$$

7. Разряд бойынша шепке жылыстырыу операциясы **K SHL N**. Shl операциясында дәслеп **K** санын екилик санақ системасына өткерип, еки байтлы кетекшеге жазады. Еки байтлы кетекшеде жазылған сан **N** орынға шепке жылыстырылынады.

Мысал:  $3 \text{ Shl } 5 = ?$ . 3 саны компьютер ядында 2 байтлы кетекшеге екилик санақ системасында жазылыўы **0000 0000 0000 0011** хәм Shl операциясы орынланғанда хәрбір бит 5 орынға шепке жылыстырылынады **0000 0000 00110 0000** хәм ол **96** ға тең болады.

$$2 \text{ Shl } 7 = 256 \quad | \quad 6 \text{ Shl } 6 = 384 \quad | \quad 13 \text{ Shl } 4 = 208 \quad | \quad -4 \text{ Shl } 5 = -128 .$$

8. Разряд бойынша оңға жылыстырыў операциясы **K SHR N**. Shr операциясы дәслеп **K** санын екилик санақ системасына түрлендирип, еки байтлы кетекшеге жазады. Еки байтлы кетекшеде жазылған сан **N** орынға оңға жылыстырылынады хәм нәтиже онлық системада алынады.

Мысал:  $96 \text{ Shr } 5 = ?$ . 96 саны компьютер ядында 2 байтлы кетекшеге екилик санақ системасында жазылыўы **0000 0000 0110 0000** хәм Shr операциясы орынланғанда хәрбір бит 5 орынға оңға жылыстырылынады. **0000 0000 0000 0011** хәм ол **3** ге тең болады .

$$| \quad 256 \text{ Shr } 7 = 2 \quad | \quad 96 \text{ Shr } 5 = 3 \quad | \quad 13 \text{ Shr } 4 = 0 \quad | \quad 7 \text{ Shr } 7 = 0 \quad | .$$

9.Бит бойынша қосыў (**OR**) операциясы. Or операциясы санлардың екилик системада жазылыўы бойынша төмендеги кесте тийкарында қосыў әмелин орынлайды. Нәтиже онлық системада алынады

$$| \quad 1 \text{ Or } 1 = 1 \quad | \quad 0 \text{ Or } 1 = 1 \quad | \quad 1 \text{ Or } 0 = 1 \quad | \quad 0 \text{ Or } 0 = 0 \quad | .$$

Мысал:  $12 \text{ Or } 22$  аңлатпасында нәтиже неге тең болады ?

$$\begin{array}{l} 0000 \ 0000 \ 0000 \ 1100 \ \text{Or} \ 0000 \ 0000 \ 0001 \ 0110 \quad ( \ 12 \ \text{Or} \ 22 = 30 \ ) ; \\ 0000 \ 0000 \ 0001 \ 1101 \ \text{Or} \ 0000 \ 0011 \ 0011 \ 1110 \quad ( \ 29 \ \text{Or} \ 64 = 93 \ ) . \end{array}$$

10. Бит бойынша көбейтiў (**AND**) операциясы. And операциясы операндалар үстінде төмендеги кестеге сәйкес көбейтiў әмелин орынлайды.

$$| \quad 1 \text{ And } 1 = 1 \quad | \quad 0 \text{ And } 1 = 0 \quad | \quad 1 \text{ And } 0 = 0 \quad | \quad 0 \text{ And } 0 = 0 \quad | .$$

Онлық системада берілген санлар операция орынланған ўақытта екилик системаға түрлендириледі. Нәтиже онлық системада алынады .

Мысалы: Егерде  $A=12$  хәм  $B=22$  болғанда,  $A$  **And**  $B$  операциясы нәтийжеси неге тең болады?  $A$  хәм  $B$  компьютер ядында 4 байтлы орын ийелеп хәм олар екилик системада төмендегише жазылады

0000 0000 0000 1100 **And** 0000 0000 0001 0110 ( 12 **And** 22 = 4 );  
 0000 0000 0001 1101 **And** 0000 0011 0011 1110 ( 29 **And** 64 = 0 ).

**11. Битли XOR операциясы.** **Xor** операциясында ( **Or** операциясын шығарып таслаушы ) нәтийже төмендеги кесте тийкарында алынады.

| 1 **Xor** 1 = 0 | 0 **Xor** 1 = 1 | 1 **Xor** 0 = 1 | 0 **Xor** 0 = 0 |.

Мысал: 12 **Xor** 22 аңлатпасында нәтийже неге тең болады?

0000 0000 0000 1100 **Xor** 0000 0000 0001 0110 ( 12 **Xor** 22 = 26 ),  
 0000 0000 0001 1101 **Xor** 0000 0011 0011 1110 ( 29 **Xor** 64 = 93 ).

**12-13. Унарлы операцияларда** ( + ) белгиси операнд белгисин өзгериссиз қалдырады, ал ( - ) белгиси операнд белгисин қарама-қарсы белгиге өзгертеди

Мысал: | +(-706) = -706 | +( +45 ) = 45 | +( 103 ) = 103 | -( -256 ) = 256 |.

**14. Арифметикалық бийкарлау (NOT) операциясы.** **Not** операциясы пүтин санның екилик коды бойынша инверсия орынлайды. Яғный, 0 санын 1 ге ямаса 1 санын 0 ге өзгертеди

Мысалы: | **Not** 0 = -1 | **Not** 73 = -74 | **Not** (-73) = 72 | **Not** (-189) = 188 |.

## § 2.2. Салыстырыу операциялары

Салыстырыу (қатнас) операциялары еки операндлар ямаса аңлатпалардың мәнислерин салыстырыуда нәтийженің “**True\Шын**” ямаса “**False\Өтирик**“ екенлигин тексереди (9-кесте). Салыстырыу операциялары АТ лерде шәрт бойынша өтиу операторларында жий қолланылады. Объектлердің мәнислерин салыстырыуда мағлыұматлардың скаляр түрлери, санақлы түрлери, массивлер, көпликлер х.т.б. пайдаланылады. Нәтийже, барлық ўақытта Буль санына тең болады.

Салыстырыу операциялары 9- кесте

Операциялар	Жазылыуы	Нәтийжелер
Тең	$A = B$	Егер $A=B$ болса, мәниси <b>True</b>
Тең емес	$A \diamond B$	Егер $A \diamond B$ болса, мәниси <b>True</b>
Үлкен	$A > B$	Егер $A > B$ болса, мәниси <b>True</b>
Үлкен хәм тең	$A \geq B$	Егер $A \geq B$ болса, мәниси <b>True</b>
Киши	$A < B$	Егер $A < B$ болса, мәниси <b>True</b>
Киши хәм тең	$A \leq B$	Егер $A \leq B$ болса, мәниси <b>True</b>
Дереклик	$A \text{ In } B$	Егер $A \in B$ болса, мәниси <b>True</b>

**Ескертиу.** Егерде шәртлер орынланбаса, онда салыстырыу операциясы нәтийжелери **False** болады.

## § 2.3. Логикалық операциялар

ТП тилинде логикалық бийкарлау **Not**, логикалық қосыу **Or**, логикалық көбейтиу **And** хәм **Xor** (**Or** - операциясын бикарлаушы) операциялары пайдаланылады. Логикалық аңлатпа нәтийжеси Буль санларына (**True**, **False**) тең болады (10-кесте):

Логикалық операциялар 10 - кесте

Операнд А	Операнд В	<b>Not A</b>	<b>A Or B</b>	<b>A And B</b>	<b>A Xor B</b>
<b>True</b>	<b>True</b>	<b>False</b>	<b>True</b>	<b>True</b>	<b>False</b>
<b>True</b>	<b>False</b>	<b>False</b>	<b>True</b>	<b>False</b>	<b>True</b>
<b>False</b>	<b>True</b>	<b>True</b>	<b>True</b>	<b>False</b>	<b>True</b>
<b>False</b>	<b>False</b>	<b>True</b>	<b>False</b>	<b>False</b>	<b>False</b>

## § 2.4. Стандарт математикалық функциялар

Математикалық элементар функциялар ( синус, косинус, квадрат корен, натураль логарифм х.т.б.) есаплау жумысларында кеңнен пайдаланылады. Олар АТ лерде стандарт функциялар деп аталады (11-кесте).

Стандарт функцияларды жазыу қағыйдалары:

1. Функция аты алты хәриптен үлкен болмайды;

2. Функция аргументи оның атамасынан кейин жай қаўсырмаға алып жазылады;

3. Функция аргументи турақлы, өзгериуши шамалар, арифметикалық аңлатпалар хәм екинши бир стандарт функциялар болыуы мүмкин;

4. Тригонометриялық функциялар аргументі радиан өлшемінде беріледі. Егер аргумент X градуста берілген болса, оны радиан өлшемге өткеріу  $X' = ( \text{Pi} * X ) / 180$  формуласы менен, ал радианлық өлшемнен градуслық өлшемге өткеріу  $X = ( 180 * X' ) / \text{Pi}$  формуласы менен анықланылады.

5. Егер функцияның аргументі функцияның анықлануы облыстына кирмесе, мысалы ушын, терис санның логарифмин табыу, терис саннан корень шығару компьютерде есаплау процессин үзиліске түсіреді. Компилятор бул кәтелик ҳаққында мәлимлеме береді.

#### Стандарт функциялар

#### 11- кесте

Математикалық жазылыуы	ТП тилинде жазылыуы	Аргумент түри	Функция түри
1. $y =  x $	Abs(x)	Integer Real	Integer Real
2. $y = \text{Argtg } x$	Argtan(x)	Integer Real	Real
3. $y = \text{Cos } x$	Cos(x)	Integer Real	Real
4. $y = e^x$	Exp(x)	Integer Real	Real
5. $y = \{x\}$	Frac(x)	Integer Real	Real
6. $y = [x]$	Int	Integer Real	Integer
7. $y = \text{Ln } x$	Ln(x)	Integer Real	Real
8. $y = 3,14159\dots$	Pi	Real	Real
9. $y = \text{Sin } x$	Sin(x)	Integer Real	Integer Real
10. $y = x^2$	Sqr(x)	Integer Real	Integer Real
11. $y = \sqrt{x}$	Sqrt(x)	Integer Real	Integer Real

**Ескертиу:** ТП орталығына элементар функциялар толығыменен киритилинбеген ( Arccos, Log<sub>a</sub>N х.т.б.). Олардың мәнислери келтириу ямаса түрлендириу формулалары жәрдемінде анықланылады (2 - қосымша).

### § 2.5. Операциялар орынланыу басқышлары

Аңлатпаларда операциялардың орынланыу басқышлары төмендеги кестедеги тәртіб бойынша әмелге асады (12-кесте).

#### Операциялардың орынланыу тәртіби

#### 12- кесте

Операциялар	Басқышлар	Категориясы
1. @, Not	Биринши	Унарлы
2. *, Div, Mod, And, Shl, Shr	Екинши	Көбейтиу
3. +, -, Or, Xor	Үшинши	Қосыу
4. >, >=, <, <=, <>, =, In	Төртинши	Салыстыруу

Операциялардың орынланыуында төмендеги қағыйдалар есабға алынады:

1. Аңлатпада операндлары арасында әмеллер орынланғанда дәслеп жоқарғы басқыштағы операциялар орынланылады;
2. Аңлатпада операндлар арасында бир басқыштағы әмеллер орынланғанда, есаплау алдыменен шеп тәрәптеги операциядан басланылады;
3. Жай қаусырма ишинде жазылған аңлатпа өз алдына операнд деп қаралады хәм оның ишиндеги жазылған операндлар, сәйкес басқыш тәртіблери бойынша алды менен орынланылады ;
4. ТП тилинің стандарт функциялар хәм процедуралар мәнислерин есаплау басқа операциялардан бурын орынланады.

### § 2.6. Арифметикалық аңлатпалар

Арифметикалық аңлатпа - нәтийжелери пүтин ямаса ҳақыйқый санларға тең болады. Ең әпиуайы арифметикалық аңлатпаларға : белгиге ийе болмаған мәнислери пүтин ямаса ҳақыйқый санлар болған турақлы хәм өзгеріуши шамалар; элементлериниң мәнислери пүтин ямаса ҳақыйқый түрдеги массивлер; нәтийжеси пүтин ямаса ҳақыйқый санлар болған функциялар мәнисин есаплаулар киреди

Мысал :  $E := 8.906$ ;  $N1 := 255$ ;  $A := X$ ;  $C2 := FZX1$ ;  
 $X := \text{SUM}[I]$ ;  $FX := \text{Argtan}(X)$ ;

Өзгеріушилер хәм массив элементлериниң мәнислери арифметикалық аңлатпада есапланыуға шекем анықланған болыулары тийисли. Қурамалы арифметикалық аңлатпаларды бирнеше әпиуайы аңлатпаларға бөлип алып, оларды бирнеше мәртебе избе-из орынлауға хәм жай қаусырмааларға бөлип алып жазыуға болады.

Мәселен, операндлар мәнислери пүтин санларда берілген өзгеріушилердиң пайдаланыуда арифметикалық аңлатпаның мәниси пүтин сан болады. Мейли N хәм K пүтин турақлы ямаса пүтин түрдеги

аңлатпа болсын, онда  $A:=N+K$ ;  $A:=N-K$ ; хәм  $A:=N*K$ ; аңлатпалардың мәнислери пүтин санға тең болады. Ҳақыйкый түрдеги өзгериўшилер менен берилген арифметикалық аңлатпаның мәниси барлық ўақытта ҳақыйкый санға тең болады. Арифметикалық әмел белгилери хәм математикалық операциялар ( корень, функция мәнислерин есаплаў х.т.б ) жәрдемінде жазылған арифметикалық аңлатпалар нәтийжеси ҳақыйкый санға тең болады. Бөлиў әмели орынланғанда нәтийже, барлық ўақытта ҳақыйкый сан болады.

Арифметикалық аңлатпаларды жазыў қағыйдалары:

1. Аңлатпада барлық белгилер массив индекслери квадрат қаўсырмаларға, дәреже көрсеткишлери **Sqr** функциясы жәрдемінде сызықлы (бир қатарда) жазылады. Егерде, дәреже көрсеткиши ҳақыйкый сан болса  $a^x$ , ол **Exp(x\*Ln(a))** түринде жазылады

Математикалық жазылыўы :

ТП тилинде жазылыўы :

$$(a^2 + b^3) / (c + d),$$

$$(a*a + \text{Sqr}(b)*b)/(c+d);$$

$$A_i * B_i * \text{Cos}^2 x,$$

$$(A[I]*B[I])* \text{Sqr}(\text{Cos}(x));$$

$$A^5 = A * A * A * A * A;$$

$$A * \text{Sqr}(\text{Sqr}(A)).$$

2. Көбейтиў белгиси “\*” еки операндлар арасына қойылыўы шәртли:

$$2(A+B),$$

$$2*(A+B);$$

$$| X_i Y_i,$$

$$X[I]*Y[I];$$

3. Функция аргументлери қаўсырмаға алып жазылады:

$$\text{LnCos } x, \quad \text{Ln}(\text{Cos}(x));$$

$$| \quad \text{Ln}| x |, \quad \text{Ln}(\text{Abs}(X));$$

4. Аңлатпаны бирнеше қатарға болып жазыўға болады:

Мысал:  $Y:=A[I]*X+B$ ; аңлатпасын төмендегише жазыўларға болады

а)  $Y:=$

$$\frac{A[I]*X+B}{X+B};$$

в)  $Y :=$

$$\frac{A[I]*X}{X+B};$$

с)  $Y:=A[I]*X$

$$+ B;$$

5. Аңлатпаның қәте жазылыўлары :

$$A*-B;$$

$$A/B(-C);$$

$$A/(BC)$$

Аңлатпаның дурыс жазылыўлары :

$$A*(-B);$$

$$A/B*(-C);$$

$$A/(B*C) \text{ ямаса } A/B/C;$$

6. Берилген арифметикалық аңлатпада әмеллердің орынланыўын мысал менен түсіндиремиз:

**F :=**

$$A*\text{Sqr}(X)+b/x*c + y - \text{Sin}(x)/(a-b);$$

$$1 - \text{Sqr}(X); \quad 2 - A*\text{Sqr}(x); \quad 3 - b/x; \quad 4 - b/x*c; \quad 5 - (2+3); \quad 6) (5+y);$$

$$7 - \text{Sin}(x); \quad 8 - (a-b); \quad 9 - \text{Sin}(x)/(a-b); \quad 10 - (6-9).$$

7. Аңлатпаларда объект атларын ( идентификаторларды ) хәм функция атамаларын үлкен ямаса киши латын хәриплери менен жазыўда бирдей нәтийже алынады.

Мысал:  $Y:=A+C/(B-D)$ ;  $Y:=a+c/(B-D)$ ;  $y:=a+c/(b-d)$ ; аңлатпаларда нәтийжелер өз-ара тең болады.

8. Аңлатпада хәрбир ашылған қаўсырмаға, оған сәйкес жабыўшы қаўсырма қойылады.

## §2. 7. Логикалық аңлатпалар

Логикалық аңлатпа - логикалық операндлар, логикалық операция белгилери хәм қаўсырмалар жәрдемінде жазылады. Логикалық аңлатпаларға: логикалық турақлылар хәм өзгериўшилер, элементлери логикалық санлар болған массивлер, логикалық түрдеги функция көрсеткишлери хәм салыстырыў әмеллери киреди (10 -кесте) .

Логикалық аңлатпаларды жазыў қағыйдалары :

1. Логикалық аңлатпаларды жазыўда логикалық операция менен операнд арасында ең кемінде бир бос орын қалдырылып жазылады ;

2. Әпиўайы логикалық аңлатпалар бир логикалық операнд **True**, **False** арқалы, ямаса алдында бийкарлаў **Not** операциясы болған бир логикалық операнд жәрдемінде жазылады. Мысалы, егерде **B**, **F**- логикалық түрдеги өзгериўшилер, ал **R** хәм **C** пүтин ямаса ҳақыйкый түрдеги өзгериўшилер болсын, онда әпиўайы логикалық аңлатпалардың жазылыўы

$$F:=A; \quad B:=\text{TRUE}; \quad F:=R>C; \quad A:=\text{Not}(\text{TRUE}); \quad F:=\text{Not}(R<=C);$$

3. Қурамалы логикалық аңлатпаларда, логикалық операциялар жәрдемінде жалғас-тырылынған бирнеше әпиўайы аңлатпалар арқалы ямаса бириниң ишинде екиншиси жайласқан бирнеше қаўсырмалар жәрдемінде жазыўға болады:

$$\text{Мысал: } Y:= B \text{ And } (B \text{ Or } C) \text{ Or } (X<Y);$$

$$Y:= (Y \text{ Or } ((X>=Z)) \text{ And } (\text{Not}((B \text{ And } C) \text{ And } Y))$$

$$Y:= A \text{ And } (\text{True}); \quad Y:=A \text{ Or } (R>=C)+0.5 \text{ And } (\text{Not } L);$$

бул жерде **Y**, **A** хәм **L** - логикалық түрдеги өзгериўшилер, ал **R** хәм **C** пүтин ямаса ҳақыйкый санлы мағлыўматлар.

3. Берилген  $B:=L \text{ And } ((M \text{ Or } X) \text{ And } (X \text{ And } L))$ ; - логикалық аңлатпа мәниси төмендеги избекте орынланылады:

$$1- M \text{ Or } X;$$

$$2 - (X \text{ And } L);$$

$$3 - (1) \text{ AND } (2);$$

$$4 - B:= L \text{ And } (3);$$

Мысалы, егерде  $A=12$ ,  $C=10$ ,  $D=7$  болса,  $\text{Not}(((A>15) \text{ Or } (C=D+3) \text{ And } ((A=2) \text{ And } (D<8))))$ ; логикалық аңдатпаның мәнісі неге тең болады? Бул жерде  $A>15$  -мәнісі өтирик,  $C=D+3$  мәнісі шын, олай болса  $(A>15) \text{ Or } (C=D+3)$  -True болады.  $A=2$  аңдатпа мәнісі өтирик, ал  $(D < 8)$  мәнісі True, онда  $(A=2) \text{ And } (D < 8)$  False болады. Шеп тәрәптеги аңдатпа мәнісі шын, ал оң тәрәптеги - өтирик. Сонлықтан  $\text{Not}(\text{False And True})$  мәнісі True болады.

**Алған билимди тексеріуге арналған сораулар ?**

1. Операндлар не мақсетте пайдаланылады ?
2. Алгоритмлик тиллерде арифметикалық аңдатпаның алгебралық аңдатпадан айырма-шылығы неде ?
3. Операндлар үстінде қандай арифметикалық әмеллер орынланады ?
4. Div хәм Mod операцияларының айырмашылығы неде ?
5. Shl хәм Shr операцияларының айырмашылығы неде ?
6. Разряд бойынша битли Or, And, Xor хәм Not операцияларының логикалық Or, And, Xor хәм Not операцияларынан айырмашылығы неде ?
7. ТП тилинде қандай салыстырыу операциялары пайдаланылады ?
8. ТП тилинде қандай стандарт математикалық функциялар пайдаланылады ?
9. Стандарт функциялар қырамына киритилинбеген математикалық функциялар қалай пайдаланылады ?
10. Арифметикалық хәм логикалық аңдатпалардың мәніслери қандай санларға тең болады ?
11. Арифметикалық хәм логикалық аңдатпалардың орынланыу өзгешелиги неде ?

**Өзлестирілген билимди беккемлеуге арналған шынығыулар.**

2.1. Аңдатпалардың мәнісин анықлаң :

- a)  $1+19 \text{ Mod } 5$                       b)  $(1+19) \text{ Mod } 5$                       c)  $3 \text{ Div } (5+15 \text{ Mod } 4)$   
d)  $(1+25 \text{ Div } 5) \text{ Mod } 2$             e)  $1+25 \text{ Div } 5 \text{ Mod } 2$             f)  $(15-4 \text{ Div } 7) \text{ Mod } 2$   
g)  $X:=8+6 \text{ Mod } 4+2$ ;  $Y:=X \text{ Div } 5+5$ ;  $Z:=(Y*5 \text{ Div } 2)$             k)  $\text{Mod } 2*2$ ;  $Z = ?$

2.2. Битли операциялар менен жазылған аңдатпа мәніслерин анықлаң :

- a)  $17 \text{ Shl } (3 + 17 \text{ Shr } 2)$                       b)  $(1+16 \text{ Shl } 4) \text{ Shr } 2*2$   
c)  $X:=8+3 \text{ Shr } 2$ ;  $Y:=X \text{ Shl } 5$ ;  $Z:=(Y \text{ Shl } 3) \text{ Shr } 2$ ;  $Z = ?$   
d)  $(17 \text{ or } 25) \text{ And } 13$                       e)  $(17 \text{ And } 3) \text{ Xor } (5 \text{ or } 22)$   
f)  $X:=(22 \text{ Or } 12) \text{ And } (29 \text{ Or } 64)$ ;  
 $Y:=5+X \text{ Xor } 25/5$ ;  $Z:=\text{Not}((X \text{ Or } Y) \text{ And } (X \text{ Xor } Y))$ ;  $Z = ?$

2.3. ТП тилинде жазылған арифметикалық аңдатпаларды алгебралық аңдатпа көринисинде жазың :

- a)  $a*x+B/c*X+D/(-c)+a*b/(c*x)$                       b)  $x+1/(\text{Sqr}(x)+1/((a*x+b)+1/(\text{Sqr}(ax+b))))$   
c)  $\text{Abs}(x+\text{Sqr}(x)+\text{Sin}(\text{Sqrt}(\text{Abs}(x))))$                       d)  $(x+\text{Sqrt}(x)+\text{Sin}(\text{Sqrt}(\text{Abs}(x))))$

2.4. Стандарт функцияларды пайдаланып аңдатпаларды ТП тилинде жазың :

- a)  $\sqrt{b + \sqrt{b^2 + 2a^2}}$  ;                      b)  $(x^2 + 1 + \sqrt{x + x^3})^{1/2} + 1$  ;  
c)  $\lg \frac{x^2+1}{x^2+2} \text{ ctg} \sqrt{|1-a|}$  ;                      d)  $1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$  ;  
e)  $\sqrt[3]{1 + \sin^2 x + tg^2 y}$  ;                      f)  $\sin \frac{x}{y} - \sqrt{(\ln x + \cos x)^2 + 1}$  ;  
g)  $\sin \frac{x}{y} - \sqrt{(\ln x + \cos x)^2 + 1}$  ;                      k)  $\sin(x\sqrt{x} + |y|) - 3 \lg(x^2 + 5)$  ;

2.5. Аңдатпаларда операциялардың орынланыу ізбе-излигин анықлаң :

- a)  $A*\text{Sqr}(X)+B/X*G - X*\text{Exp}(A*B)/A-B$                       b)  $A*B/A*A/C*B$   
c)  $X+X/(X+Y)*z/(X+Z)-(X-\text{Sin}(y))/\text{Cos}(x)$                       d)  $X*X/X-1+(-6.3*X-\text{Sqr}(X)+4.3)$

2.6. Аңдатпаларда пайдаланылған артық қаўсырмаларды алып таслаң :

- a)  $(X*Y)/Z$                       b)  $(((a+b)/c)*d)$                       c)  $(x+(i-j))+((b/c+5))*(b+c)$                       d)  $((a/x)*c)$

2.7. Аңдатпаның мәніслерин есаплаң :

1. Берилген  $a=3$ ,  $b=4$ ,  $c=5$  хәм  $d=2$  ;

- a)  $Y:=a*b/\text{Sqr}(c)$ ;                       $Y=?$                       b)  $Y:=c/a/b*d$ ;                       $Y=?$   
c)  $Y:=a*a*a*\text{Sqr}(b)*\text{Sqrt}(c)$ ;                       $Y=?$                       d)  $Y:= -h*b+c*d/a$ ;                       $Y=?$

2. Берилген  $i = 2$ ,  $j = 3$ ,  $k = 4$ ,  $l = 2$ ,  $r = 3$  хәм  $S = 7$  ;

- a)  $a:=i*j/r*\text{Sqr}(r)$ ;                       $a=?$                       b)  $a:=i*\text{Sqr}(r) / i*l$ ;                       $a=?$   
c)  $a:=j/i*j$ ;                       $a=?$                       d)  $a:=k*k / i / j / l$ ;                       $a=?$

2.8. Егерде  $I=1$ ,  $J=2$ ,  $K=3$ ,  $I=4$  хәм  $M=5$  болса, онда төмендеги логикалық аңдатпаның мәніслерин анықлаң :

- a)  $(J>K) \text{ Or } (I=M-1)$  ;                      c)  $(I+J)<(K*I) \text{ And } (\text{Not}(M>3) \text{ Or } (K<2))$   
b)  $(J>K) \text{ Or } (I=M-1)$  ;                      d)  $\text{Not}((J-3)=-1) \text{ Or } (\text{Not} (M<=50) \text{ And } (K=i+1))$

2.9. Логикалық аңдатпада операциялардың орынланыу ізбе-излигин анықлаң :

- a)  $L \text{ And } (\text{Not}(M \text{ Or } (X=Y+2)) \text{ And } Z)$                       b)  $A \text{ Or } (R<=C+0.5) \text{ And } (\text{Not}(\text{False}))$   
c)  $\text{Not}((A-D/Z)*B>=A) \text{ And } (X \text{ Or } (\text{Not } Y))$                       d)  $A \text{ Or } (R<=C+0.5) \text{ And } (\text{Not}(\text{False}))$

2.10. Логикалық аңдатпаның математикалық жазылыуын ТП тилинде жазың :

- a)  $(J > K) \vee (L \wedge M)$       b)  $(A+B-C)/2 < 3.6 \vee (L1 \wedge B1) \wedge \vee$  шын  
 c)  $(J \leq K) \wedge (L \vee M)$       d)  $B1 \wedge \wedge \vee$  Отирик  $\wedge (A+B-C)/2 < 3.6$

2.11. Логикалық аңдатпалардың мәнісін анықтаң

- a)  $X := \text{False}; Y := \text{Not } X \text{ And } \text{False}; Y = ?$   
 b)  $A := -0.3E02; B := 7.3E-2; Y := \text{False Or } (A+B \leq 0.6); Y = ?$   
 c)  $X := \text{True}; Y := \text{True}; Z := \text{False}; A := 8.4E-3; B := -0.001E05; S := 6.1;$   
 $Y := X \text{ Or } (\text{Not } Y \text{ And } (\text{Not } Z \text{ Or } (S - 6.3/A + B > 0)))$ ;

### III. ТУРБО ПАСКАЛЬ ТИЛИНИҢ ТИЙКАРҒЫ ОПЕРАТОРЛАРЫ

**Операторлар** - жәрдемінде тилдің структуралық қурылысына сәйкес, мәселені шешіу алгоритми тийкарында программа тексти жазылады. Операторлар **әпиуайы, оқытыу-шығарыу** хәм **қурамалы** болып үш топарға бөлинеди. ТП тили ең аз сандағы операторларға ийе.

#### § 3.1. Әпиуайы операторлар

Қурамында басқа операторлар пайдаланылмаған оператор - **әпиуайы оператор** деп аталады. Оларға **мәнісләу (меншиклестириу), шәртсиз өтиу, процедураны орынлау, бос** хәм **тамамлау** операторлары киреди.

##### §3.1.1. Мәнісләу операторы

**Жазылыу үлгиси:**  $\langle \text{атама} \rangle := \langle \text{аңлатпа} \rangle$ ; ямаса  $V := A$ ;

бул жерде **V (атама)** - объект ямаса өзгеріуши аты (идентификатор);  $:=$  - мәнісләу оператор белгиси; **A** - аңлатпа. **Аңлатпа** - турақлы хәм өзгеріуши шамалар, қатарлық, символлы (литерли) мағлыұматлар түри, арифметикалық хәм логикалық аңлатпалар, функциялар көрсеткіши арқалы арифметикалық, логикалық операциялар хәм қаўсырмалар жәрдемінде тутастырылып жазылыуы менен анықланылады (§2.6, §2.7, §5.1, §5.2, §5.3).

**Ұазыйпасы:** Алды менен **A** аңлатпасының мәніси есапланылып, соңынан **V**- идентификаторы **A** ның мәнісин меншиклеиди.

Мәнісләу операторы  $:=$ , математикалық теңлик  $=$  белгисинен өзгеше. Мысалы,  $X = X * X + A * X - C$  түриндеги математикалық жазылыу кәте, ал мәнісләу операторы арқалы  $X := X * X + A * X - C$  жазылыуы дурыс. Хәқыйқатында,  $A=1$ ;  $C=2$  хәм  $X$  өзгеріушиниң дәслепки мәніси **3** ге тең болса, онда арифметикалық әмеллер орынланғаннан кейин  $X$  тиң мәніси **10** ға тең болады ( $X := 3 * 3 + 1 * 3 - 2 = 10$ ). Нәтижәде,  $X$  ушын ажыратылынған кетекшеде **3** орнына **10** саны жазылады.

Мәнісләу операторында, аңлатпаның мәніси шептен оңға әмеллердиң орынланыу басқышы бойынша есапланылады (12 - кесте, § 2.5).

Мысаллар : 1.  $EPS := 0.00001; V1 := -573.46901;$   
 $M := \text{SUMMA}; DRT := 'a * x^3 - c * x + d * 5'; SinA := \text{Sqrt}(1 - \text{Sqr}(\text{Cos}(A)));$   
 $R2 := \text{Sqr}(X) * \text{Sqrt}(\text{Abs}(Y)); Y := a * \text{Sqr}(X) + b * X + c; R := R1/R2/R3 + R;$

2. Мейли  $B1 := \text{False}; B2 := \text{True}$ ; логикалық санлар берилген болсын, онда  $L := B2 \text{ Or } B1$  орынланғанда  $L = \text{True}$ , ал  $\text{Log} := B1 \text{ And } B2$  орынланғанда  $\text{Log} = \text{False}$  болады.

3. Мейли объектлер (өзгеріушилер) төмендегише дағазаланған болсын :

$\text{Var } I, J, K : \text{Integer}; A, B, C : \text{Real};$   
 $L1, B1 : \text{Boolean}; Ch1, Ch2 : \text{Char};$

a) Төмендеги мәнісләу операторлары орынланылады:

$I := (I * J) \text{ Div } (J + I); J := I \text{ Mod } K; A := A - K * C / B;$   
 $Ch1 := 'P'; Ch2 := '+'; K := J * K \text{ Div } I; L1 := (L1 \text{ Or } B1) \text{ And } (\text{Not } B1);$

в) Төмендеги меншиклестириу операторлары орынланбайды, себеби объект түрлериниң дағазаланыуы сәйкес емес:

$I := A * J / K; Ch1 := A + B; A := \text{True}; L1 := (A * B + C) - 4.8;$   
 $L1 := L1 \text{ And } (\text{Not } A) + C; Ch2 := 'Pascal';$

**Ескертиу.** 1. Мағлыұматлардың файллы түрлеринен басқа, барлық түрлердеги шамалар арасында мәнісләу операторы орынланылады;

2. Мәнісләу операторында  $"V := A;"$  V атама хәм A аңлатпасының нәтижеси бирдей түрде дағазаланған болыуы хәм мағлыұматлар түрлериниң бирдейлиги хәм сәйкеслиги есабға алынады (§ I. 6).

3. **Real** түрдеги атама V, нәтижеси путин сан **Integer** болатуғын A мәнісин меншиклестиреди (хәқыйқый санға түрлендиреди). Ал, керисинше **Integer** түрдеги V атама, мәніси **Real** болған A аңлатпа мәнісин меншиклестире алмайды. Бул жағдайда программа үзиліске түседи.

##### § 3.1.2. Шәртсиз өтиу операторы

Программаның орынланыу процессинде, бөліуішینیң мәніси нолге тең болса, операндлар жүда киши санға көбейтилине (бул санның тәртібинің жоғалыуына алып келеди), ямаса дискреминант мәніси нольден киши болса х.т.б. жағдайларда АТ лерде **шәртсиз өтиу операторы " GOTO "** жәрдемінде, буның келип шығыу себебин тексеріу операторына басқарылыу өткерилинеп хәм шешим қабыл етиледі.

**Жазылыу үлгиси: Goto < таңба >;**

бул жерде, **Goto** - хызметши сөз, көрсетилинген “ **таңбаға өт \перейти к метка** ”. **Таңба** - өтилетуғын оператор алдына қойылған атама.

**Уазыйпасы:** Мәселени шешіу алгоритмине байланыслы программаның орынланыу ізбе-излигин өзгертиу.

**Goto** операторын пайдаланыу мүмкиншиликлери :

1. Таңба қойылған оператор " : " белги менен ажыратылынып жазылады :

**Metka : X := - B/(2 \* A) ;**

**127 : Writeln( 'Хакыйкый санлар коплагинде шешими жок ' ) ;**

2. Таңбалар қайсы блокта көрсетилинген (**Label** бөлими) болса, олар сол блок ишинде пайдаланылады ;

3. **Goto** жәрдемінде бир блоктан екинши блок ишине кириуе болмайды ;

4. **Goto** жәрдемінде блок ишинен, бас программаға өтиуе болады ;

5. Операторларға қойылған хәрбир таңба программаның **Label** бөлиминде алдын-ала көрсетилинген болыулары тийисли ;

6. Программада мүмкиншилиги болғанша **Goto** операторы аз пайдаланылады. Себеби, программаның орынланыу ізбе-излигин логикалық тексеріу қыйынласады ;

7. **Procedure, Function** хәм бас программада бирдей атамалы таңбаларды пайдаланыуға болады;

8. Айырым операторлардың алдына таңба қойыуға болады. Оған **Goto** менен өтилиуі шәртли емес. Бирақта, олар **Label** бөлиминде көрсетилинеди ;

9. Таңба ушын узынлығы **127** ден үлкен болмаған цифрлар, хәриплер ямаса хәрипли - цифрлар ізбе-излиги пайдаланылады.

### § 3.1.3. Бос оператор

**Бос оператор** - деп, хеш қандай әмел орынламайтуғын операторға айтылады. Ол программаның тил синтаксисине байланыслы оператор жазыу мүмкин болған кәлеген орында, ең кемінде бир бос орын “ | ” қалдырылып, ноқатлы үтир “ ; ” қойылып жазылыу менен анықланылады. Программада хәрбир оператордан кейин, егерде оның жазылыуы қағыйдасы бойынша ноқатлы үтир қойылатуғын болса, онда бул белгини бир мәртебе қойыу жеткиликли. Бос оператор - программаның түсиникли оқылыуы ушын пайдалана-нылады.

Мысал: **Program Prim\_PUSTO;**

**Label Met1, 22, M11;**

**< тәриплеу бөлими >;**

**BEGIN < операторлар > If A=0 Then GoTo Met1;**

**{ блок ақырына өтиу } < операторлар >**

**Met1: ; { таңбадан кейин бос оператор жайласқан }**

**< операторлар > If X<0 Then GoTo M11;**

**22: A:=Sqrt(X); R:=2.05; G:=A \* R; ; { бос оператор }**

**K:=Sqr(G+A); < операторлар >**

**If K>=1001 Then GoTo M11 Else GoTo 22;**

**M11: { бос оператор } ; END.**

### § 3.1.4. Процедураны орындау операторы

**Жазылыу үлгиси : < процедура аты > [( хакыйкый параметрлер дизими)];**

**Уазыйпасы:** Программа дүзиуіш тәрәпинен мәселени шешіу алгоритмине байланыслы, бирнеше мәртебе такрарланып пайдаланылатуғын программа үлеси өз алдына процедура операторы менен шөлкемлестирелинеди. **Procedure < аты > (IV бөлим)** ямаса **стандарт процедуралар** программа ишинде бирнеше мәртебе шақырылып орынланады хәм алынған нәтийже кейинги есаплауға өткерилинеди.

Мысал: **Program Prim\_PROCEDURE;**

**< тәриплеу бөлими >**

**Procedure PV1; Begin < операторлар > End;**

**Procedure PP2( X,Y : Integer; Var ZZ : Real);**

**Begin < операторлар > End;**

**BEGIN < операторлар >**

**PV1; { PV1 процедурасын орындау } < операторлар >**

**PP2(A,B,C); { PP2 процедурасын орындау }**

< операторлар > END.

### § 3.1.5. Тамамлау операторы

**Жазылыу үлгиси:** End [ ; ] [ . ]

бул жерде End “акыры\конец” хызметши сөз.

**Ұзыйпасы:** Бас программаның, жәрдемши программалық блоктың ямаса программалық модулдин ең акырғы операторы ретинде пайдаланылады.

End орынланбайтуғын оператор. Программада хәрбир блоктан кейин "End;" койылады. Ал, бас программаның тамамланыуы "End." аркалы анықланылады. Егерде бирнеше End лер избе-из жазылса, онда " ; ямаса . " белгилери ең акырына койыу жеткиликли.

### § 3.2. Мағлыұматларды оқытыу-шығарыу процедуралары

Объектлердин (берилгенлердин) дәслепки мәнислерин программада, компьютер ядына киритпей турып нәтийже алыу мүмкин емес екенлиги бизге белгили. Мағлыұматларды киритиу операциясы - бул, сыртғы курылмадан берилген шамалардың мәнислерин искерлик ядға өткериу. Шығарыу операциясы - бул, киритиуге кери процесс болып, яғный қайта исленген мағлыұматларды искерлик ядтан сыртғы шығарыу курылмаларына өткериу болып табылады. Киритиу - шығарыу терминалы (дисплей-экран, клавиатура), алфавитли - цифрли баспаға шығарыу курылмасы принтер, ийилмели дискет, катты диски - винчестер х.т.б. лар сыртғы курылмалар топарына киреди.

Мағлыұматларды компьютер ядына киритиу - шығарыу операциялары Read, Readln, Write, Writeln операторлары жәрдеминде орынланады. ТП тилинде бул операторлар - процедура деп айтылады.

#### § 3.2.1. Мағлыұматларды оқытыу процедурасы

**Read** процедурасының ұзыйпасы: санлы, символлы, қатарлық х.т.б мағлыұматларды искерлик ядға киритиу.

**Жазылыу үлгиси:** Read(X1, X2, ... , XN);

Read(FL, X1, X2, ... , XN);

бул жерде X1,X2,...,XN - өзгериушилердин (объектлердин) атамалары; FL- оқытылатуғын файл менен байланыстырыушы файл көрсеткиши аты (VI-бөлим). X1, X2, ... , XN өзгериушилердин (параметрлердин) мәнислерин оқытыуда клавиатура жәрдеминде хәр бир мағлыұмат арасында ең кеминде бир бос орын қалдырылып териледи. Read процедурасында киритилинетуғын мағлыұматлар терилип болғаннан кейин "Enter" түймеси басылады. Өзгериушилердин мәнислери ТП тил синтаксисине сәйкес келиуи тийисли. Мысалы, K өзгериушиси Integer түрде дағазаланған болып, ал киритилген мағлыұмат мәниси Real түринде болса, онда оқытыуда кәтелик хаққында мәлимлеме сөз экран бетинде пайда болады : I/O error XX бул жерде, XX-кәтелик коды.

Мысал: Program Prim\_BBOD;

Var K:Integer; I: Real; Ch: Char;

BEGIN Read(I,K,Ch); < операторлар > END.

{программаны орышлау ушын, CTRL + F9 түймелери биргеликте басылады}

1. -28.617 -170 'P' "Enter басылады"

2. -170 -28.67 'P' "Enter басылады"

Биринши вариантта мағлыұматлардың оқытылыуы дурыс, ал екнши вариант кәте, себеби пүтин сан орнына хақыйкый сан киритилип турыпты. Егерде бир программа ишинде бирнеше Read процедурасы пайдаланылса, онда мағлыұматлар сәйкес избе-изликте, хәрбир Read параметрлери, бириниң кейнинен екншиси киритилип барылады.

Мысал: Program Prim\_BBOD;

Var A, B, SUM1 : Integer;

C,D,SUM2 : Real;

BEGIN Read(A,B); SUM1:=Sqr(A+B/A);

Read(C,D); SUM2:=C\*D - SUM1 END. { Ctrl + F9 }

1170 36 "Enter басылады"

-0.739 67.78 "Enter басылады"

ReadLn процедурасының Read ден өзгешелиги, параметрлер дизиминдеги кейинги мәнис киритилип болғаннан кейин, курсор өзлигинен тез екнши қатарға өтиледи ("Enter" түймесин басыу шәрт емес).

#### §3.2.2. Мағлыұматларды шығарыу процедурасы

**Write** процедурасының ұзыйпасы: санлы, символлы, қатарлық түриндеги х.т.б. мағлыұматларды экран бетине ямаса қағазға басып шығарыу.

**Жазылыу үлгиси:** Write(' S1', Y1, ' S2', Y2, ... , ' Sn', YN);

Write(FL, Y1, Y2, ... , YN);

бул жерде **S1, S2, ... , Sn** лер қатарлық мағлыұматлар ( ASCII коды символлар избе-излиги); **Y1, Y2, ... , YN** - **Integer, Real, Char, Boolean** (4-, 5- кесте ) х.т.б. түрлердеги объектлер атамасы; **FL** - файллы өзгеріуши аты. Өзгеріушилер мәнислери принтерде шығарылынатугын болса, онда **Uses Printer** ; модули программаға киритилинип **FL** орнына **Lst стандарт файл** аты жазылады.

**Writeln** процедурасының **Write** процедурасынан өзгешелиги, параметрлердің мәнислери шығарылып болғаннан кейин, курсор өзлигинен таза қатарға тез өтип жайласады. **Writeln** операторы параметрлерсиз жазылса, онда курсор таза қатарға өткерилинеди.

**Ескертуі.** **1.Input** хәм **Output** файллары менен **Eoln** хәм **Eof** функциялары байланыстырылған жағдайда информацияларды избе-из файлдан оқыўға ямаса файлға жазыўға болады (**VI**-бөлим ).

**2. Write[ln]** операторда, қысқа узынлықтағы аңлатпаларды шығарыў процедурасы параметри орнында жазып, нәтийжесин алыўға болады.

**1-программа.** Берилген : а)  $X = (a+c-b)^2 - \text{Arctg}(a/b) * (a-c)/b$  , арифметикалық ;

б)  $B = \text{Not} ( \text{Log1 And} ( \text{Not B1} ) \text{ Or B2} ) \text{ Xor} ( P \leq R )$  , логикалық аңлатпалардың мәнислерин есаплаў программасы :

**Program Prog1\_Read\_WRITE;**

**Uses Printer;**

**Var A, B, C, X : Real;**

**B : Boolean;**

**Const B1=True; B2=False; Log1=True; R=-6.7; P=15.8;**

**BEGIN Write('A, B, C = ? '); Read( A, B ); Readln( C );**

**Write('A= ', A, ' B=', B, ' C=', C );**

**Writeln( ' 1 - косылыушы =', Sqr(A + C-B));**

**Writeln( ' 2 - косылыушы =', Arctan(A/B) \* (A-C)/B);**

**X:= Sqr(A+C-B) - Arctan( A/B) \* (A-C)/B;**

**Writeln( LST, ' X мәниси =', X );**

**B:=( Log1 And ( Not B1 ) Or B2 ) Xor ( P<= R);**

**Writeln(' B1=', B1); Writeln(' B2=', B2: 6); Writeln(' B=', Not B); END.**

### §3.2.3. Мағлыұматларды шығарыў үлгилери

**Writeln** ямаса **Write** параметрлери бойынша мағлыұматларды шығарыў майданына жайластырыўды анықлаўшы үлги белгилери:

**I, P, Q** - пүтин санлы аңлатпалар; **R** - хақыйқый сан түрдеги аңлатпа;

**B** - логикалық аңлатпа; **Ch** - символлы аңлатпа;

**St** - қатарлық түрдеги аңлатпа; **#** - цифрлар;

**\*** - " + " ямаса " - " белгилери; **&** - бос орын (пробел) .

**а) I үлгиси.** **I** - шамасының онлық системадағы мәнисин курсор турған орыннан бир бос орын қалдырып шығарады:

**I:= 134; Write( I ); &134**

**I:= -5671; Write( -I ); &5671**

**I:= 17; Write( -I\*5,I ); & -178517**

**б) I : P үлгиси.** **I** - шамасының онлық системадағы сан мәнисин оң шетки орыннан баслап **P** кеңликтеги майданға шығарады:

**I:= 134; Write( I :4 ); &134**

**I:= 37134; Write( I:10 ); &&&&&37134**

**I:= 134; Write( I\*2:6); &&&&268**

**с) R үлгиси.** **R** - шамасының мәнисин шығыў майданы енине **18** символлар жайласатуғын онлық жылжымалы ноқатлы үтир түрінде шығарады:

егерде  $R \geq 0$  болса, шығарыў үлгиси **\_#.#####E\*##**

егерде  $R < 0$  болса, шығарыў үлгиси **\_-#.#####E\*##**

**R:=715.432; Write( R ); &7.1543200000E+02**

**R:=-7154.2; Write( R ); &-7.1543200000E+04**

**R:=-97.9145E+1; Write( R ); &-1.9791450000E+01**

**д) R:P үлгиси.** **P** майданның шетки оң тәрәпинен баслап цифрлардың онлық системада берилиўиниң жылжымалы ноқатлы үлгисинде **R** шама мәниси шығарылынады (**P** -шығарылатуғын символлар саны).

Егерде  $R \geq 0$  болса, онда **&\_#.## ... #E\*##** форматы пайдаланады. Бул жерде шығарылатуғын символлардың ең қысқа узынлығы **7** ге тең болады. Егерде  $R < 0$  болса, онда **&\_#.## ... #E\*##** шығарылатуғын орынның ең қысқа узынлығы **8** символ. Онлық ноқаттан кейин ең кеминде бир цифр шығарылады:

**R:= 511.04; Write( R :15 ); &5.110400000E+02**

R:= -.000151104; Write( R :15 ); &-5.11040000E-03  
 R:= 46.78; Write( R :12 ); &-4.67800E+01

e) R:P:Q үлгиси. P майданға шетки оң тәрәптен баслап R диң мәнислери онлық көринисте турақлы ноқатлы үлгиде шығарылады. Онлық ноқаттан соң Q ( 0<=Q <= 24) санының бөлшек үлесин анықлаушы цифрлар шығарылады. Егерде Q=0 болса, онда бөлшек үлес хәм онлық ноқат шығарылмайды. Егерде Q>24 болса, онда натийжени шығарыўда жылжымалы ноқатлы R:P форматы пайдаланылады.

R:= 511.04; Write( R :10 :6 ); &511.040000  
 R:= -146.078; Write( R : 7 :3 ); &-46.078  
 R:= Pi; Write( R : 20 :16); &3.1415926535397932

f) Ch үлгиси. Курсор жайласқан орыннан баслап Ch символлы өзгериўши мәниси шығарылады:

Ch := 'x'; Write( Ch ); &x  
 Ch := 'A'; Write( Ch,'+',Ch); &A + A

h) Ch : P үлгиси. Шетки оң позициядан баслап P кеңликтеги майданға Ch тың мәниси шығарылады.

Ch :='x'; Write( Ch : 4 ); &&&x  
 Ch :='A'; Write( Ch : 8 ); &&&&&& A

i) St үлгиси. Курсор турған орыннан баслап қатарлық мағлыўмат St шығарылады (қатарлар узынлығы, берилген қатар узынлығына сәйкес келиўи тийисли \$5.3)

St:=' ВЕДОМ1'; Write( St ); &ВЕДОМ1  
 St:=' A'; Write( St, ' +abs' ); &A +abs

j) St : P үлгиси. Шетки оң орыннан баслап P кеңликтеги майданға қатарлық мағлыўматлар шығарылынады:

St:= 'ЭВМ'; Write( St : 6); &&&ЭВМ  
 St:= ' abcdef '; Write( St :7, St :4 ); &abcdef abcd  
 St:=' ВЕДОМ1'; Write( St ); &ВЕДОМ1

k) B үлгиси. Курсор турған орыннан баслап B логикалық аңлатпаның мәнислери True ямаса False шығарылады:

B := TRUE; Write( B ); &TRUE  
 B := False; Write( B ); &False

n) B : P - үлгиси. Шетки оң орыннан баслап P кеңликтеги майданға B логикалық аңлатпаның мәнислери True ямаса False шығарылады:

B := TRUE; Write( B : 8 ); &&&&TRUE  
 B := False; Write( B : 10); &&&&&False

2-программа: Берилгенлерди хәм есаплаў нәтийжелерин Write хәм Writeln операторларын пайдаланып мағлыўматларды шығарыў үлгилери бойынша дүзилген программа :

```

Program Prog2_Writeln;
Var I : Integer; { путин сан }
    R,R1,R2 : Real; { хакыйкый санлар }
    Ch,Ch1 : Char; { символлы мағлыўматлар }
    St : String; { қатарлық мағлыўматлар }
    B : Boolean; { логикалық санлар }
BEGIN Writeln('1234567890123456789012345678901234567890');
      { Read(I,R,Ch,St,B); }
      I:=2001; R:=5679.04308324; R1:=5.9876E-05; R2:=-9.9848E+04;
      Ch:=' Я '; Ch1:=' I '; B:=True;
St:='КАРАКАЛПАК МАМЛЕКЕТЛИК УНИВЕРСИТЕТИНЕ 25 жыл.';
Write('I=',I); Writeln(-I,'+',Sqr(I):18, i:12);
Write('R=',R); Writeln('R:20',R:20); Writeln('R:17:9',R:17:9);
Write('Pi =',Pi:50); Writeln('E^x=',Exp(1.0):23:21);
Write('Ch=',Ch); Writeln('Ch1:20',Ch1:20);
Write('St=',St); Writeln('St : 35',St:35);
Write('B= ',B); Writeln('B:10',B:10);
Writeln('B : 3',(Not B):3); END.

```

*Ескертиў.* Егерде өзгериўшилер мәнислери ямаса аланатуғын нәтийжелер мәнислери жетерли дәрежеде үлкен ямаса жүдә киши санлар болса Longind, Donly, Extend (4,-5- кестелер) сөзлери арқалы идентификаторлар дағазаланады.

### § 3.3. Қурамалы есаплаў процессин шөлкемлестириў операторлары

Қурамалы оператор - деп, қурамында мәселени шешиў алгоритмине сәйкес орынланыўшы бирнеше операторлардың программада операторлық қаўсырма Begin .. End сөзлери менен шегараланған үлесине

айтылады. Курамлы операторлар **бириктирилген, шәрт бойынша баскарыу, вариант бойынша сайлау** хәм **такрарланыушы** болып төрт топарға бөлинеди.

**Бириктирилген операторлар** - бул, бирнеше операторлардың бири екиншисинен ноқатлы үтир аркалы айрылып **Begin** хәм **End**; операторлық қауысырмаға алып жазылыуы менен анықланылады. Курамлы оператордың ишинде екинши бир курамлы операторларды шөлкемлестириуіге болады.

**Жазылыу үлгиси: Begin < 1 - оператор; >**

**Begin < операторлар > End;**

**< 2- оператор; > ... < N-оператор > End; ....**

**Ескерту:** Курамлы оператордың ишине, сырттан шәртсиз өтиу операторы аркалы кириуіге болмайды;

**3-программа.** Үшмүйешликтің үш тәрепи **a=12.8, b=7.7, c=6.9** бойынша, оның майданын ГЕРОН формуласы жәрдеминде есаплау программасы.

**Program Prog3\_SabcGERON;**

**Var A,B,C, P, Sabc : Real;**

**BEGIN Writeln( ' A, B , C манислерин киритин ');**

**{ курамалы операторлар блогы } Readln( A, B, C);**

**P:=0.5\*(A+B+C) ; Sabc:=Sqrt(P\*(P-A)\*(P-B)\*(P-C)) ;**

**Writeln( ' A =', A:10:6, ' B =', B:10:6, ' C =', C:10:6);**

**Writeln( ' Sabc майдан =', Sabc :10:6); END.**

### § 3.3.1. Шәрт бойынша баскарыу операторы

Программаның орынланыуын **баскарыу операторы** қойылған шәрттің (Буль аңлатпасының) мәнисине байланыслы программада айырым оператордың, операторлар топарының ямаса блоктың ўақытша орынланыуын ямаса орынланбауын тәмийнлейди.

**Жазылыу үлгиси : A. If <шәрт> Then <оператор>;**

**B. If <шәрт > Then <1-оператор> Else <2-оператор>;**

бул жерде **If** (егерде \ если), **Then** (онда \ то), **Else** (ямаса \ иначе) хызметши сөзлер.

**Шәрт** - бул, логикалық аңлатпа (§2.3.). Ол әпиуайы хәм курамалы болып бөлинеди. Шәрттің курамалығы **And, Or, Not, Xor** логикалық хәм салыстырыу операциялары "**>**", "**>=**", "**<**", "**<=**", "**=**", "**<>**" жәрдеминде шөлкемлестириледі.

**Ўазыйпасы:** **If** операторының **A** үлгисинде (шәртли оператордың қысқа формасы): егерде, шәрт мәнисі **True** болса, онда **Then** сөзинен кейинги **<оператор>** орынланады; егерде **False** болса, онда **If ...** ден кейинги биринши оператор орынланады. **If** операторының **B** жазылыуында ( шәртли оператордың толық формасы), егерде аңлатпа мәнисі **True** болса **< 1- оператор >**, ал **False** болса, онда **< 2- оператор >** орынланады.

**If** операторы курамында екинши мәртебе **If** операторы қатнасса, онда ишлей бирлескен шәртли операторлар хаққында сөз болады хәм ол төмендегише жазылады:

**If < 1-шәрт > Then Begin If < 2-шәрт > Then < 1-оператор >**

**Else < 2-оператор > End Else < 3-оператор >;**

Ишлей бирлескен шәртли операторларда хәрбир **Else** ушын одан бурын сәйкес **Then** жазылады. Усындай усылда, шәртли операторларды бирнеше мәртебе ишлей жайластырыуға болады.

Төменде **If ...** операторын қолланыу әпиуайы мысаллар менен түсиндирилген.

**1. If A <> 0 Then V:=5.7/A Else V:=5.7;** бул жерде шәрт орынланғанда **A<>0** болса, онда **V** атама **5.7/A** мәнисти өзлестиреди; Ал, егерде **A=0** болса, **V=5.7** болады.

**2. If B1 <> 0 Then Begin B2 <> 0 Then V:=B2/B1**

**Else V:=-1.0 End Else V:= Abs(B2-B1);**

Бул оператор орынланғанда: егерде **B1 <> 0** хәм **B2 <> 0** болса, онда **V=B2/B1** болады; ал **B1 <> 0** хәм **B2=0** болса, онда **V=-1.0** болады. Ал, **B1=0** болса, онда **V=Abs( B2-B1)** болады.

**3. Шәртли операторды пайдаланыуда** ТП системасының семантикалық қағыйдасы сақланылады. Мысалы, **If A1 Then If A2 Then OP1 Else OP2** ; туринде шәртли операторды шөлкемлестириу кәте болады. Бул оператордың дурыс жазылыуы :

**If A1 Then Begin If A2 Then OP1 End Else OP2;**

**4. a) If A>B Then Y:=Sqrt(A-B) Else Y:=Sqrt(Abs(A-B));**

**b) If X<Y Then Max:=Y Else Max:= X;**

**c) If A<B Then Begin Writeln( ' A < B ' ); D:=B-A; Write( ' D=' ,D) End Else Writeln( ' A >= B D:=', A-B) ;**

**d) If X<0 Then Sgn:=-1 Else If X=0 Then Sgn:=0 Else Sgn:=1;**

**e) If (A=B) And (C=D) Then**

**Begin Write( ' НОРМА ' ); F:=0 End**

**Else Begin Writeln( ' НОРМАДАН АРТЫК ' ); F:=100 End;**

f) **If Color = Red Then Begin Xx:= cos(Y); Yy:= Alpha+Beta;  
Color:=Yellow End Else Begin x:=Sin(Y); Color:=Blue End;**

*Ескертуу.* Ишлей бирлескен шэрт бойынша өтиу операторларын бирнеше мәртебе пайдаланыу максетге мууапык емес, себеби программаны қайта ислеуде хэм анализлеуде қыйыншылық тудырады.

**4-программа.**  $a_i x^2 + b_i x + c_i = 0$  квадрат теңлемени  $a_i, b_i, c_i$  коэффициентлер саны **100** шекем өзгергенде шешимлерин анықлау программасы.

```

Program Prog4_KBT;
  Const Iter=100;
  Label M1,M2,KON,11;
  Var A,B,C : Real; D,X1,X2 : Real; S: Integer;
BEGIN S:=1;
  11: Writeln('A B C= ?'); Readln(A,B,C); D:=Sqr(B) - 4*A*C;
  If D < 0 Then Goto M1 Else If D = 0 Then Goto M2;
  X1:=(-B+Sqrt(D)) / (2*A); X2:=(-B-Sqrt(D)) / (2*A);
  Writeln('Шешим X1=',X1:12);
  Writeln('Шешим X2=',X2:12); Goto KON;
  M2: Writeln(' Шешимлер X1=X2 ', -B / (2*A)); Goto KON;
  M1: Writeln(' Шешимге ийе емес ');
  KON: S:=S+1; If S<=100 Then GoTo 11; END.

```

### § 3.3.2. Вариант бойынша сайлау операторы

Вариант бойынша сайлау операторы **Case** селектор (**жалғастырыушы**) - деп, аталыушы аңлатпа хэм сайлап алынатуғын тураклылар дизимине сәйкес операторлар избе-излиги көринисинде жазылады. **Case** операторы **If** операторының ұлыұмаласған түри.

**Жазылыу үлгиси:** **Case** < аңлатпа - селектор > **Of**

< 1-дизим > : < 1-оператор ; >

< 2-дизим > : < 2-оператор ; >

-----

< N-дизим > : < N-оператор > [ **Else** < оператор > ] **End;**

бул жерде **Case** ( болған жағдайда \ в случае); **Of** ( дан \ из) хызметши сөзлер.

**Узыйпасы:** **Case** операторында дәслепп аңлатпа-селектор мәниси есапланылады. Селектордың өзлестирген мәниси бойынша есаплау варианты сайланады. Егерде, селектор өзлестирілген мәнис вариантылар дизимине сәйкес келмесе, онда **Else** сөзинен кейинги оператор орынланады. Егерде **Else** сөзи **Case** операторында пайдаланылмаса, онда **End** сөзинен кейинги биринши оператор орынланады.

**Селектор - аңлатпасының өзлестиретуғын мәнислери:** ( -32768 . . 32767) аралықтағы пүтин санлар менен, буль санлары менен, литерли, санақлы хэм аралықта берілген мағлыұматлар түрлерименен анықланылады. Хақыйқый санлар хэм қатарлық мағлыұматлар селектор мәнислери бола алмайды. Сайланатуғын тураклылар дизими бири екиншисинен үтир менен айрылған кәлеген сандағы дизимлер ямаса шегераланған аралықларда бериледи. Вариантта жазылған опертторлар қурамалы болса, **Begin** хэм **End** сөзлери менен шегераланылады. Аралық шегараларын анықлаушы белги " .. " еки тураклы шамалар арасына қойылады. Вариантды анықлаушы тураклылар түри барлық ұақытта селектор аңлатпасы түриниң дағазаланыуына сәйкес келеди. **Case** операторына мысаллар:

1. Селектор мәнислери пүтин санға тең болса :

```

Begin Read(I);
  Case I Of
    1 : Y:=A*Sqr(X)+B*X-D;
    2 : Y:=Sin(X);
    3 : Y:=0;
    4 : Y:=Ln(X);
    5 : Y:=Abs(X)
  Else Writeln('Y=',Sqr(X)) End; ... End;

```

2. Селектор мәнислери шегераланған аралық түрде берілген болса :

```

Case S Of
  1..10 : Writeln('сан ',S:3,' 1-ден 10 аралыкта');
  11..20 : Writeln('сан ',S:3,' 11-ден 20 аралыкта');
  21..40 : Writeln(' сан ',S:3,' 21-ден 40 аралыкта');
  Else Writeln(' сан',S:3,'40>болған аралык'); End;

```

3. Селектор мәниси санақлы турдеги мағлыұматлар менен берілген болса :

```

Var SEASON:(Winter,Spring,Sumer,Autum);

```

```

BEGIN < операторлар >
  Case SEASON Of
    Winter : Writeln(' Winter - кыс ');
    Spring : Writeln(' Spring - бахар ');
    Summer : Writeln(' Summer - жаз ');
    Autum : Writeln(' Autum - гуз '); End;

```

4. Селектор мәніси символлы мәнісге тең болса :

```

Var Ch : Char; Log : Boolean;
Begin Read( Ch); ...
BEGIN < операторлар; > Case Ch Of
  ' Y ', ' Д' : Log := True;
  ' N ', ' H' : Log := False; End; ... End;

```

5. If B Then < 1-оператор > Else < 2-оператор >; шәрт бойынша өтиү операторы Case аркалы төмендегише дүзиледи:

```

Case B Of
  True : < 1- оператор >;
  False : < 2- оператор > End;

```

- Ескертуу:* 1. Case операторында қойылған таңбалар программада Label бөлиминде тәрипленбейди;  
 2. Goto операторы менен Case операторы вариантларына өтиүге болмайды;  
 3. Case операторы курамында Read [ln] хәм Write [ln] процедураларын пайдаланыўға болады.  
 5-программа. Компьютер ядына киритилинген гәп ишинде неше мәртебе А,Е, Р хәм С хәриплери хәм “|\_” бос орын белгиси бар екенлигин анықлаў программасы. (гәптин хәрбир хәриби ең соңғы символ “**ноқат**” белгиси болғанға шекем избе-из қатар бойынша киритилинип барылады).

```

Program Prog5_Case;
Label M1;
Var VUKBA : Char;
    Ch_A, Ch_E, Ch_P, Ch_C, C_ : Integer;
BEGIN Ch_A:=0; Ch_E:=0; Ch_P:=0; Ch_C:=0; Ch_:=0;
M1: Readln( VUKBA);
    Case VUKBA Of
      'A', 'a' : Ch_A:=Ch_A+1;   'E', 'e' : Ch_E:=Ch_E+1;
      'P', 'p' : Ch_P:=Ch_P+1;   'C', 'c' : Ch_C:=Ch_C+1;
      ' ' : Ch_:=Ch_+1;   End;
If VUKBA <> ' ' Then Goto M1;   Writeln(' Хәриплер саны ');
Write( ' A=' , Ch_A :4, ' E=' , Ch_E :4, ' P=' , Ch_P :4,
      ' C=' , Ch_C :4, '   =' , Ch_ :4)   END.

```

### § 3.4. Такрарланыўшы есаплаў процессин шөлкемлестириў операторлары

Есаплаў процессинде, өзгериўшилердиң хәрқыйлы мәнислеринде бир аңлатпаны бирнеше мәртебе индекси бойынша такрарлап есаплаўға тура келеди. АТ лерде бул программа үлеси **цикл**- деп аталады. Цикллы есаплаў процесси, программа көлемин киширейтеди хәм компьютерде есаплаў уақты азайады. ТП тилинде циклли есаплаў процесси үш түрдеги операторлар жәрдеминде шөлкемлестириледі:

1. Цикл параметри бойынша, такрарлаў операторлары;
2. Алдыңғы шәрт бойынша, такрарлаў операторы;
3. Ақырғы шәрт бойынша, такрарлаў операторы .

#### § 3.4.1. Цикл параметри бойынша такрарлаў операторлары

Жазылыў үлгиси : A. For < цикл параметри I >:= N1 To N2 Do  
 Begin < цикл денеси > End;  
 B. For < цикл параметри I >:= N1 Downto N2 Do  
 Begin < цикл денеси > End;

бул жерде For ( ушын \ для ); To ( шекем \ до ); Do ( орынлаў \ выполнение ); Downto ( төменге \ вниз ) хызметши сөзлер. N1 хәм N2 - цикл параметриниң дәслепки хәм ақырғы мәнислерин анықлаўшы турақлы шамалар ямаса мәнислери пүтин санға тең аңлатпалар. Цикл параметрлери хақыйқый санлардан басқа, кәлеген скаляр түрдеги мағлыўматларды өзлестиреди. Цикл параметрлериниң мәнислери пүтин, санақлы хәм аралық түрлерде бериледи.

Ўазыйпасы : ”For... To ... Do” операторында цикл параметри мәніси бирге артып барады. Ал, ”For... Downto ... Do” операторында бирге кемип барады.



### § 3.4.2. Алдыңғы шәрт бойынша такрарлау операторы

Жазылыу үлгиси: **While < шәрт > Do**

**Begin < цикл денеси >; End;**

бул жерде **While** ( хәзирше \ пока ), **Do** - хызметши сөзлер; Шәрт - бул аңлатпасы; Цикл денеси, әпиуайы ямаса курамлы операторлар избе-излиги менен дүзиледи.

**Ұазыйпасы:** Цикл денеси ишиндеги хәр бир оператор орынланыуы алдынан аңлатпа - шәрт мәниси тексериледи. Егерде шәрт мәниси **True** болса, онда цикл денеси орынланады. Шәрт мәниси **False** болғанға шекем цикллы есаплау процесси қайталанады хәм соңынан цикл денесинен кейинги 1- оператор орынланады.

Мысаллар: 1. **I:=30; While(I < 10 ) Do Begin I:=I+1; Writeln(I) End;**

Мысалда, **I** өзгеріушиниң дәслепки мәниси **10** дан үлкен, сонлықтан дәслепки мәниси **False**, онда цикл денеси орынланбайды.

2. **While** операторында шәртли аңлатпаның мәнисин өзгертиу операциясы (циклдан шығыу шәртин анықлаушы) шөлкемлестириледі. Кери жағдайда цикл денесинде шексиз есаплау дауам етиледі: **< параметр > := < шегаралық мәнис > + h;**

**While I<100 do Begin< операторлар >**

**I := I+5.3; {циклдан шығыуды шөлкемлестириу }< операторлар > . . . ;**

3. Циклли есаплау процесси шексиз орынланатуғын етип шөлкемлестирилиуде, шәрт **True** турақлысы менен алмастырылады ямаса кәлеген жағдайда мәниси **True** болатуғын аңлатпа арқалы жазылады.

**While True Do Write('шексиз цикл');**

**While 9=Sqr(3) Do Write('шексиз цикл');**

7-программа. **(n!)!** ды есаплау программасы :

**Program Prog7\_WHILE;** { (n!)! ды есаплау }

**Var X,N :Integer; F,Fact : Longint;**

**BEGIN X:=1; Read(N); F:=1; Fact:=1;**

**While X<=N Do Begin F:=F\*X; X:=X+1; End;**

**Writeln(' ':5,N:10,'! ',F : 15);**

**X:=1; While X<=F Do Begin Fact:=Fact\*X; X:=X+1; End;**

**Writeln(' ': 3, N : 5, ' !! ', Fact : 25) END.**

### § 3.4.3. Ақырғы шәрт бойынша такрарлау операторы

Жазылыу үлгиси: **Repeat < 1- оператор >; -----**

**< N- оператор > Until < шарт >;**

бул жерде **Repeat** ( такрарлау \ повторить ); **Until** ( соған шекем \ до сих (тех) пор ) хызметши сөзлер; Шәрт-бул аңлатпасы. Ол логикалық хәм салыстыруу операциялары жәрдемінде дүзиледи. **Repeat** хәм **Until** сөзлери арасында цикл денесине тийисли операторлар жазылады.

**Ұазыйпасы:** Дәслеп цикл денеси бир мәртебе орынланып , соңынан қойылған шәрт тексериледи. Егерде, шәрт мәниси **False** болса, онда циклли есаплау процесси такрарла-нылады. Ал, шәрт мәниси **True** болса, онда циклдан шығылады.

**Repeat** операторы **While** операторынан үш түрдеги өзгешеликке ийе:

а) Цикл денеси ең кемінде бир рет орынланады;

б) Цикл денеси, қойылған шәрт мәниси **True** болғанша қайталанылады ;

в) Цикл денеси ишинде операторлық **Begin ... End;** қаусырмаларсыз, кәлеген сандағы операторларды жазыуға болады.

**Ескертиу:** 1. **While, Repeat** операторлары жәрдемінде ишлей бирлестирилген циклли есаплау процессин шөлкемлестириуға болады;

2. Цикл денеси ишиндеги операторлардың ең кемінде биреуи қойылған шәрттиң мәнисин өзгертиуға тәсирин тийгизиуи тийисли, кери жағдайда цикл шексиз қайталанады ;

3. **Repeat, While** операторларында цикл денеси ишинен қойылған шәрт орынлан-бастан бурын шығып кетиу ушын **If ... Then GoTo < тангба >;** операторларын пайдаланыуға болады.

Мысаллар: 1. Шексиз циклдиң такрарланыуы:

**A := 6; D:=0; Repeat < операторлар > A:=A+D Until A>25;**

2. Шекли циклдиң такрарланыуы:

**A := 6; D:=7; Repeat < операторлар > D := D+0.5; A:=A+D Until A>25;**

8-программа. 1 ден **N** ге шекемги санлардың квадратларының косындысын **For ... To ... ; For ... Downto If ... ; While ... ; Repeat** операторлары жәрдемінде есаплау программасы : {  $S = 1^2 + 2^2 + 3^2 + 4^2 + \dots + N^2 = (N*(N+1) * (2*N+1)) / 6;$  }

```

Program Prog8_SumN;
Label Met1, Met2, Met3, Met4, Met5, Met, 11,22;
  Var I, Sum1, Sum2, Yc, N, K : Integer;
BEGIN Writeln(' косынды есаплау шэртлери ');
  Writeln(' Yc = 1 болса, For ... операторменен ');
  Writeln(' Yc = 2 болса, While ... операторменен ');
  Writeln(' Yc = 3 болса, Repeatr ... операторменен ');
Writeln(' Yc = 4 болса, If ... операторменен ');
Writeln(' Yc = 5 болса, For ... Downto .. операторменен ');
Readln(N,Yc); Writeln(' N = ', N:4, ' Yc=', Yc:4)
  If Yc=1 Then Goto Met1;  If Yc=2 Then Goto Met2;
  If Yc=3 Then Goto Met3; If Yc=4 Then Goto Met4;
  If Yc=5 Then Goto Met5;
Met1: Sum1:=0; For I:=1 To N Do Sum1:=Sum1+Sqr(I);
  Writeln(' Summa = ', Sum1: 6); Goto Met;
Met5: Sum1:=0; For I:=N Downto 1 Do Sum1:=Sum1+Sqr(I);
  Writeln(' Summa = ', Sum1: 6); Goto Met;
Met2: Sum1:=0;K:=1; While K<=N Do Begin Sum1:=Sum1+Sqr(K);
  K:=K+1; End;  Writeln(' Summa = ', Sum1: 6); Goto Met;
Met3: Sum1:=0;K:=1; Repeat Sum1:=Sum1+Sqr(K); K:=K+1
  Until K>N;  Writeln(' Summa = ', Sum1: 6); Goto Met;
Met4: Sum1:=0; K:=1;
  22: Sum1:=Sum1+Sqr(K); K:=K+1;
  If K<=N Then Goto 22;  Writeln(' Summa = ', Sum1: 6); Goto Met;
Met: Writeln('N*(N+1)*(2*N+1)/6=', N*(N+1)*(2*N+1)/6:6);  END.

```

### Алған билимди тексеріуге арналған сораўлар

1. Программаластырыў тилинде операторлар не мақсетте пайдаланылады ?
2. Қандай операторлар әпиўайы ямаса қурамалы деп айтылады ?
3. Меншиклестириў операторы қалай жазылады хәм ўазыйпасы неден ибарат ?
4. Меншиклестириў операторының оң тәрәпиндеги арифметикалық аңлатпа қурамына кирген өзгериўшилердиң мағлыўмат түрлери қандай шамалар болады хәм алынған нәтийже шеп тәрәпиндеги идентификатор түрине сәйкес қалай түрлендириледі ?
5. Логикалық меншиклестириў операторының арифметикалық меншиклестириў операторы-нан өзгешелиги неде ?
6. Шэртсиз өтиў операторы не мақсет ушын пайдаланылады ?
7. Мағлыўматларды компьютер ядына киритиўде ямаса ядтан шығарыўда қандай процедуралар пайдаланылады хәм олар қалай жазылады ?
8. Компьютер ядынан мағлыўматларды үлги (формат) бойынша шығарыўдың қандай формалары бар хәм шығарыў үлгилери не мақсетте пайдаланылады ?
10. Шэрт бойынша басқарыў операторы қалай орынланылады хәм шөлкемлестириледі ?
11. Вариант бойынша сайлаў операторы қалай орынланылады хәм шөлкемлестириледі ?
12. Цикллы есаплаў процесси программада не мақсет ушын пайдаланылады хәм неше түрли операторлар жәрдемінде орынланылады ?
13. Такрарлаў (цикл) операторларының өзгешеликлери неде ?
14. **While** хәм **Repeat** операторларын пайдаланыўда айырмашылықлары нелерде?

### Өзлестирилген билимди беккемлеўге арналған шынығыўлар!!

3.1. Меншиклестириў операторы жәрдемінде төмендеги аңлатпаларды жазың хәм олардың мәнисин есаплаў программасын дүзиң :

a)  $Y = \sin^2 a^2 + \sqrt[3]{\cos(a - 3.14)} + (1 / (a^2 - b^2))x,$

b) 
$$f = \frac{Y}{1 + \frac{ZY^2}{1 + \frac{3Y^3}{1 + \frac{4Y^4}{1 + 5Y^5}}}}$$

c)  $P(x) = (((a_5x + a_4)x + a_4)x + a_3)x + a_2)x + a_1)x + a_0$

3.2. Логикалық аңлатпаны меншиклестириў операторы менен жазың:

a)  $a \neq b \vee c = d \wedge a > d$

b)  $A \wedge B + C - 2A B = 2A C \vee A - C < 5.6$

c)  $a/b + c - 2b = 2c \vee a \geq c$

d)  $x^3 - a > 0.2 \quad e^x < x + 2^{\sqrt{x-1}} \leq 0.2$

3.3. Логикалық меншиклестируй операторы орынланғанда мәніслери неге тең болады (X:=True; A:=3.6; B:=-4.7;) ?

- a) X:=A>B And ( Not X);                      b) X:=( X Or (A+B) <=6.2 ) Not ( Not X);  
 c) X:= Not X And (Not False);              d) X:= X And (( Not X ) Or (Not False ));

3.4.Шәртсиз өтуй операторының қайсылары кәте?

- a) Goto 001;                                      b) Goto A0;                                      c) GotoA01;  
 d) Goto M1A;                                   e) Goto M&A;                                   f) Goto IF;

3.5. Сызықлы есаплай процессине программа дүзин

a)  $R = e^{|x^2 - y^2|} + 2ctg(x^2 - xy); \quad x = 2,25; \quad y = 1,83;$

b)  $F = \sqrt[5]{x + \sqrt[5]{|y|} + \log_2(\sin^2(xy))}; \quad x = 1,23; \quad y = 0,34;$

c)  $F = \frac{10 + \sqrt{x}}{|y^2 - x|} + e^{x^2 - 1} + |\cos y|; \quad x = 2,32; \quad y = -3,4;$

3.6. Тармақланыўшы есаплай процессине программа дүзин

a) 
$$P(x) = \begin{cases} ctg(x+a) / tg(x+b), & x < b, & a = \sqrt{2\pi} \\ x^2 e^{-ax}, & x = b, & b = \ln 3, \\ \sqrt{b^2 + \sin^2 ax}, & x > b, & x = tg \frac{\sqrt{5}}{22}, \end{cases}$$

b) 
$$P(x) = \begin{cases} a \cos^2 bx, & x < a, & a = tg 47^0 \\ \arccos \frac{1}{1+x^2}, & x = a, & b = \cos 47 / \log_3 7, \\ 2^{-bx}, & x > a, & x = \log_{6,7} \cos 47, \end{cases}$$

3.7. Такрарланыўшы есаплай процессине программа дүзин

a)  $S = \sum_{i=1}^{10} \prod_{m=i}^{14} tg \frac{i^m + m^i + 6^i + 4}{m^i + m^6 + im + 3};$       b)  $S = \sum_{i=3}^{12} \prod_{m=3}^{12} (-1)^m \frac{(i^3 + m)^{\frac{3}{4}}}{\sqrt{\ln(i+m)} + i^{\frac{1}{4}}};$

3.8. Ишки цикллы есаплай процессине программа дүзин

a)  $S = \sum_{i=2}^8 \prod_{m=1}^{10} \frac{i^m + im}{m^i + m^3 + im + 5};$       b)  $S = \prod_{i=1}^{10} \prod_{m=i}^{14} tg \frac{i^m + m^i + 6^i + 4}{m^i + m^6 + im + 3};$

c)  $S = \prod_{i=1}^{14} \sum_{m=1}^{10} tg \frac{i^m + m^i + 6^i + 4}{m^i + m^6 + im + 3}$       d)  $S = \sum_{i=3}^{12} \sum_{m=3}^{12} (-1)^m \frac{(i^3 + m)^{\frac{3}{4}}}{\sqrt{\ln(i+m)} + i^{\frac{1}{4}}};$

3.9. Бир ўақытта өзгерийши шамалардың мәніслери бойынша есаплай процессин программаластырың :  $Y = (AX^3 + BX^2 - C) / (B \ln|X| - A)$ ; бул жерде  $-2.5 \leq X \leq 1.5$  аралықта  $Xh = 0.25$  адым менен өзгергенде,  $A = 0$ ,  $B = 1.0$ ,  $C = 2.0$  өзгерийшилер, сәйкес  $Ah = 0.5$ ;  $Bh = -0.5$ ;  $Ch = 1.5$  адымлар менен өзгереді.

3.10. Case операторы жәрдеминде берилген функцияларды есаплай блогин дүзин :

1.  $F_1 = (X_1 - X_2^2) + (X_1 - X_2)^2$ ;      2.  $F_2 = (X_2 + X_1^2) + (X_1 + X_2)$ ;  
 3.  $F_3 = 100 * (X_1 - X_2^2) + (X_1 - X_2)^2$ ;      4.  $F_4 = (X_1 + 2X_2^2 - 7) + (2X_1 - X_2 - 3)^2$ ;

3.11. Шәрт бойынша өтуй операторын Case операторы менен қалай жазылады

(B1 = False ; B2 = True)

If B1 Then Begin If B2 Then Y:=B1 And B2 Else Y:=B1 Or B2 End  
 Else Y:= B1 Xor ( Not B2 );

3.12. Ишки цикллы есаплай процесси If... операторы менен қалай жазылады

S:=0; For I:= 10 To 105 Do  
 For T:= 1 To 15 Do S:= S + I \* T;

3.13.  $3X^2 - 11X - 30 = 0$  квадрат теңлемени, берілген [ 4.5; 6.5 ] аралықта  $Eps=10^{-4}$  дәллікте итерация усылы менен шешиўди программаластырың :

Есаплаў формуласы :  $X_n = -c / (a X_{n-1} + b)$  ;

3.14. Квадрат корен  $Y = \sqrt{a}$  хәм куб корен  $Y = \sqrt[3]{a}$  мәнислерин  $Eps = 10^{-6}$  дәллікте итерация усылы менен есаплаўды программаластырың ( $|X_n - X_{n-1}| < Eps$ ) :

а)  $\sqrt{a}$  ды есаплаў формуласы :  $X_n = 0.5 * (a / X_{n-1} + X_{n-1})$  ;

в)  $\sqrt[3]{a}$  ды есаплаў формуласы :  $X_n = (1/3) * (2 * X_{n-1} + a / X_{n-1}^2)$  .

3.15. Шекли санлы қосындының теңлигин дәлиллейў программасын дүзиң :

а)  $S_1 = 1^3 + 2^3 + 3^3 + \dots + N^3 = (N * (N + 1) / 2)^2$  ;

б)  $S_2 = 1^3 + 3^3 + 5^3 + \dots + (2N-1)^3 = N^2 * (2N^2 - 1)$  ;

с)  $S_3 = 1 * 2^n + 2 * 2^{n-1} + \dots + N * 2 + (N+1) = 2^{n+2} - (N+3)$  ;

3.16. Алты қаналы “  $a_1 a_2 a_3 a_4 a_5 a_6$  ” цифрлары менен анықланған санлар көплигинен “Бақытлы билетлер” дүзетуғын билет номерлерин хәм олардың санын анықлаў прог-раммасын дүзиң. “Бақытлы билетлер”- бул, төмендеги теңликлер орынланылатуғын номерлер :

а)  $a_1 + a_2 + a_3 = a_4 + a_5 + a_6$  ;      б)  $a_1 + a_3 + a_5 = a_2 + a_4 + a_6$  .

3.17. Берілген санлы қатардың қосындысын есаплаў программасын дүзиң хәм алынған нәтийжелерди салыстырың:

$$S = 1 - 1/2 + 1/3 - 1/4 + \dots + 1/999 + 1/1000 = 0.69309718305934529691723;$$

Есаплаў жұмысын төмендеги усыллар менен орынлаң :

а) Қосынды, шептен оң тәрепке қарай орынлансын;

б) Қосынды, оң тәрептен шепке қарай орынлансын;

с) Қосынды, шептен оң тәрепке қарай оң ағзаларын өз алдына, терис ағзалары өз алдына есаплансын;

д) Қосынды, оң тәрептен шепке қарай оң ағзаларын өз алдына, терис ағзалары өз алдына есаплансын.

3.18. Санлы қатардың 50 ағзасына шекемги қосындыны есаплаўды программаластырың хәм теңликти дурыслығын тексериң:

а)  $3/4 = 1/(1*3) + 1/(2*4) + 1/(3*5) + 1/(4*6) + \dots + 1/((n-1)*(n+1))$  ;

б)  $Pi/4 = 3.1415926 \dots / 4 = 1 - 1/3 + 1/5 - 1/7 + \dots + 1/(2n-1)$  ;

с)  $1/e = 1 - 1/1! + 1/2! - 1/3! + 1/4! - \dots + (-1)^n / n!$  ;

3.19. Функциялардың шексиз қатарға тарқалыўы бойынша берілген  $Eps=10^{-4}$  дәллікте мәнислерин есаплаўды программаластырың.

а)  $e^X = 1 + X/2! + X^2/2! + X^3/3! + X^4/4! + \dots + X^n/n! + \dots$  ;

б)  $3^X = 1 + (Ln 3 / 1!) X + (Ln^2 3 / 2!) X^2 + \dots + (Ln^n 3 / n!) X^n + \dots$  .

бул жерде  $0.2 \leq X \leq 0.8$  аралықта өзгереді .

3.20. Мейли  $A = 5679$  пүтин саны берілген болсын,  $3^k > A$  шәртин қанағатландырыўшы ең киши  $K$  санын анықлаң.

#### IV. ПРОЦЕДУРА, ФУНКЦИЯ ОПЕРАТОРЛАРЫ

Программа ишинде хәрқыйлы орынларда бирдей есаплаў жұмысларын бирнеше мәртебе қайта орындауға тура келеди. Әмелий программаластырыў шеберлигинде, бундай жағдайларда программа көлемин киширейтиў максетинде процедура хәм функция операторлары пайдаланылады. Олардың хәрбири өз алдына алгоритм-лестириледи. Бул программалық бирликлер жәрдемши программа - деп айтылады.

Жәрдемши программа - программада өз алдына ат қойылған логикалық жуўмақ-ланыўға ийе болған операторлар топарынан дүзилген үлес программа. Олар аты бойынша программада мәселени шешиў алгоритмине сәйкес бир неше мәртебе шақырып алынып орынланады. Жәрдемши программаларды өз алдына дүзиў мүмкиншилигин есапға алсақ, онда оларды басқа мәселелерди шешиўде пайдаланыўымыз мүмкин. Нәтийжеде, мәселелерди компьютерлерде шешиў ўақты тезлеседи.

Функция хәм процедура атлары - программа ишинде бирден-бир ат пенен бериледи. Олар программада толықтырыўшы (локаль) блок көринисинде жазылады.

##### § 4.1. Функция операторын шөлкемлестириў

Функция операторы - FUNCTION хызметши сөзи менен басланып, функция аты хәм жай қаўсырма ишинде формаль (жалған) параметрлер дизими хәм олардың түри, алынатуғын функция мәнисиниң түри хәм орынланатуғын операторлар (функция денеси) избе-излиги менен шөлкемлестирилинеди.

Жазылыў үлгиси:

Function <аты>[(<формаль параметр(лер), ... >:<түри>)] : < мәниси түри >;

**Label** <таңба(лар)>;  
**Const** <турақлы шама(лар)>;  
**Type** <мағлыұматлар түрлерин баянлаў>;  
**Var** <өзгеріушілерди дағазалаў>;  
**Begin** { функциялар денеси } **End**;

Функция операторында функция аты хәм хақыйқый параметрлер (аргументлер) дизими, аңлатпада ямаса өз алдына меншиклестириу операторы аркалы көрсетилинип жазылыу аркалы орынланады.

**Жазылыу үлгиси : <функция аты > (хақыйқый параметрлер);**

Мысал : **Write ( FAKT ( N ) :12 ) ;** { параметр өзгермели мәнисли }  
**Y:= Funk(3,17,-5) : Real;** { параметрлер, анык берилген }  
**A:=Func\_NoPar;** {хақыйқый параметрлер берилмеген, бул жерде шақырылынатуғын функция операторы параметрлерсиз жазылған болады}

Функция денеси бөлиминде <функция аты> := <аңлатпа>; бирден-бир оператор, функция аты ақырғы алынған нәтижени өзлестиреди. Функция операторында нәтиже, “функция аты“ өзлестиретуғын шамалар кәлеген скаляр түрлерде (пүгин хәм хақыйқый санлар, символлы, логикалық, шегераланған аралықта, санақлы), қатарлық - **STRING**, хәм көрсеткишли - **POINTER** түрлердеги мағлыұматлар болыулары мүмкин. Нәтиже шақырылған блокға қайтарылады.

#### § 4.2. Процедура операторын шөлкемлестириу

Процедура операторы - **PROCEDURE** менен басланып, процедура аты хәм жай қауыс ишинде формаль параметрлер дизими, баянлау бөлими хәм процедура денеси избе-излигинде жазылыу менен анықланылады.

**Жазылыу үлгиси :**

**Procedure** < процедура аты > [ < формаль параметр(лер), ....> ];

**Label** <таңба(лар)>;

**Const** <турақлы шама(лар)>;

**Type** <мағлыұматлар түрлерин баянлаў>;

**Var** <өзгеріушілерди дағазалаў >;

**Begin** { процедура денеси } **End**;

Процедураны орынлау операторы процедура аты 81м жай 3а7сырма ишинде бири - екіншисинен бтир менен айрылып жазыл2ан 8а3ый3ый параметрлер дизими менен (егерде, олар бар болса) шазырылынады.

**Жазылыу үлгиси: < процедура аты > [( хақыйқый параметрлер дизими )];**

Мысал: **Max\_Min( -4.8, 78.67, 67.908, -89.789, MaxABCD, MinABCD);**

Егерде процедура денесиндеги барлыз объектлер (параметрлер) программаны4 баянла7 блогинде да2азалан2ан болса, онда Я2ный, процедура операторын формаль параметрлерсиз жазы7а болады : < процедура аты>;

Мысал: **SumMat;**

**Metod\_Gauss;**

*Ескертиу:* Функция хәм процедура операторын орынлауда хақыйқый параметрлер менен олардың формаль параметрлери арасында, шептен оң тәрепке қарай, сәйкес избе-изликте мағлыұматлардың түрлериниң сәйкеслиги хәм бирдейлиги есабға алынады. Формаль параметрлер саны хақыйқый параметрлер санына тең болады.

**9 -программа . Cmn = n! / m! (n-m)!** формуласын **Function , Procedure** операторлары менен есаплау програмасы :

**Program Prog9\_FUNCTION\_PROCEDURE;**

**Var N,M, PP1,F1,F2 : Longind;**

**Cnm : Dobly;**

**FUNCTION Ffakt(K:Integer) : Longind;**

{ Функция операторын орынлау блогы }

**Var P : Longind; I : Integer;**

**Begin P:=1; For I:=2 To K Do P:=P\*I; FFakt:=P End;**

**PROCEDURE PFakt( K : Integer; Var PP : Longind);**

**Var P : Longind; I :Integer;**

{ Процедура операторын орынлау блогы }

**P:=1; For I:=3 To K Do P:=P\*I; PP:=P End;**

**BEGIN Write('N > ,M= '); Readln(N,M);**

{ Функция операторын шақыруу }

**Cnm:=Ffakt(N) / (FFakt(M) \* FFakt(N-M));**

**Writeln( ' Функция Cnm=', Cnm:20);**

{ Процедура операторын орынлау }

```

Pfakt(N,PP1); F1:=PP1;
Pfakt(M,PP1); F2:=PP1; PFakt(N-M,PP1);
Cnm:=F1 / (F2*PP1);

```

```

Writeln(' Процедура Cnm=', Cnm: 20) END.

```

**10-программа.** Берілген аңлатпа мәнісін параметрлері берілмеген функция хәм процедура операторлары менен есаплай программасы :

$$F = \text{th}(a * n + b) / \text{th}^2(a^2 + b^2) - \text{th}^2(a + m * b) / \text{th}(a^2 - b^2);$$

бул жерде  $\text{th } x = (e^{2x} - 1) / (e^{2x} + 1)$  формуласыменен анықланылады :

```

Program Prog10_NOPARAMETR;

```

```

Var F, F1, F2, A,B,N,M,R1,R2,R3,X,Y,TH_X : Real;

```

```

X1, X2, X3, X4 : Real;

```

```

Function F_TH ;

```

```

Begin Y:=Exp(2*X); F_TH:=(Y-1)/(Y+1) End;

```

```

Procedure P_TH;

```

```

Begin Y:=Exp(2*X); TH_X:=(Y-1)/(Y+1) End;

```

```

BEGIN Read( A,B,M,N);

```

```

X1:=A*N+B; X2:=A+M*B; X3:=A*A+B*B; X4:=Sqr(A)-Sqr(B);

```

```

F:= F_TH(X1) / F_TH(X3) - F_TH(X2) / F_TH(X4);

```

```

Writeln(' аңлатпа мәніси = ', F : 20);

```

```

X:=A*N+B; P_TH; R1:=TH_X; X:=A+M*B; P_TH; R2:=TH_X;

```

```

X:=A*A+B*B; P_TH; R3:=TH_X; X:=Sqr(A)-Sqr(B); P_TH;

```

```

F:=R1/R3 - R2/TH_X;

```

```

Writeln(' F аңлатпа мәніси = ', F : 20) END.

```

### § 4.3. Параметрлер механизми

**Function** хәм **Procedure** операторларында параметрлер берилиўи бойынша кәлеген скаляр түрлерде (4-, 5- кестелер) хәм курамалы мағлыўматлар түрлері менен (5-бөлим) берілген болыўлары мүмкин. Параметрлер үш усылда берилип анықланылад: мәнісли параметрлер, өзгермели мәнісли параметрлер хәм түри көрсетилинбеген өзгермели параметрлер.

1. Формаль параметрлерди анықлаўда **Var** сөзи жазылмай, параметрлерден кейин түри көрсетилинген болса, онда олар **мәнісли параметрлер** - деп айтылады.

Мысалы, **Function FMax(A, B, C, D : Integer) : Integer;** жазылыўында **A, B, C** хәм **D** мәнісли параметрлер. Формаль параметрлердиң мәніслери өзгергени менен, оларға сәйкес ҳақыйқый параметрлердиң мәніслери өзгермейди. Мәнісли параметрлер ушын, файллы мағлыўматлар түри пайдаланылмайды. Параметр ретинде қатарлық түрдеги өзгериўшилер пайдаланылса, онда формаль хәм ҳақыйқый параметрлердиң узынлығы **255** байттан үлкен болмайды.

2. Формаль параметрлер алдында дәслеп хызметши сөз **Var** жазылып, одан кейин олардың түри көрсетилинген болса, онда олар **өзгермели мәнісли параметрлер** - деп айтылады. Мысалы, **Procedure PMaxMin(A, B, C, D : Real; Var MaxABCD, MinABCD : Real);** , бул жерде **MaxABCD, MinABCD** өзгермели мәнісли параметрлер. Өзгермели мәнісли параметрлер пайдаланылады сол ўақытта, егерде процедураны орынлаўда параметрлердиң мәніслери шақырылған блокға қайтарылатуғын болса. Процедура хәм функция операторлары шақырылып орынланғанда, формаль параметрлер ҳақыйқый параметрлердиң мәніслерин өзлестиреди. Процедура операторында өзгермели мәнісли формаль параметрлердиң мәніслерин сәйкес ҳақыйқый параметрлер меншиклестиреди.

Ҳақыйқый параметрлердиң түрлері, саны хәм жайласыў тәртиби формаль параметрлер менен сәйкес келиўи тийисли (түри көрсетилинбеген өзгермели параметрлерди пайдаланыўда буны сақлаў шәрт емес). Егерде, формаль параметр қатарлық түрде дағазаланған болса, онда оған узынлығы **255** символға тең болған атрибут меншиклестирилинеди. Бул жағдайларда ҳақыйқый параметрлерде қатарлық түрде болып, атрибут узынлығы **255** тен үлкен болмайды. Өзгермели мәнісли параметрлер ретинде кәлеген мағлыўматлар түри пайдаланылады. Олай болса, файллы мағлыўматларда пайдаланылады.

3. Параметрлер алдында **Var** сөзи жазылып, ал параметрлер түри көрсетилмесе, олар **түри көрсетилмеген өзгермели параметрлер** - деп аталады.

Мысалы, **Procedure FReal (Var A, B, C, D : Real; Var X,Y,Z );** бул жерде **X,Y,Z** - түри көрсетилмеген өзгермели параметрлер. Параметрлердиң ҳақыйқый мәніслери яд областына көрсеткиш арқалы өткерилип жазылады. Олар менен жұмыс ислеўде программа дузиўши түри көрсетилмеген параметрлердиң мәніслериниң сәйкеслигине айрықша дыққат аўдарыў керек болады. Себеби, бул жағдайда ТП система орталығында мағлыўмат түрлериниң сәйкес келмеў жағдайлары ушырасса, онда компиляторда бул қәтелик есапқа алынбайды.

Мысалы, **FReal(Var C, D): Real;** функция операторында параметр көрсеткіштері **C** хәм **D** қәлеген мағлыұмат түрлерин өзлестиреди. Бул жағдайда, пүтин санлардың орнына символлы мағлыұматлар жазылыұы мүмкин болады.

#### § 4.4. Идентификаторлардың хәрекетте болыұ областы

Процедура хәм функция операторлары курамында қатнасуғын идентификаторлардың (объектлердің, параметрлердің) хәрекетте болыұ областын дурыс шөлкемлестиріұ қағыйдалары:

1. Хәр бир идентификатор дағазаланылады;
2. Идентификаторлардың хәрекетте болыұ областы, ол дағазаланған блок ишинде болады;
3. Жәрдемши программа ишинде идентификаторлар бирден-бир атамаға ийе болып, олар екінши объект аты ушын пайдаланылмайды;
4. Бирдей атамалы идентификаторларды, екінши бир процедура ямаса функция операторларында өз алдына қайтадан дағазалап пайдаланыұға болады;
5. Егерде барлық параметрлер (объектлер) бас программаның басында, улыұма блокда баянланылған болса, онда **Function, Procedure** операторларында параметрлер жазылыұы шәртли емес;
6. Егерде жәрдемши программада пайдаланыұшы идентификаторлар стандарт процедуралар хәм функциялар атлары менен (**Cos, Ln, Exp** х. т. б. ) сәйкес келсе (11 - кесте), онда жәрдемши программада хәрекетте болыұы шегарасында усы атамадағы стандарт функциялар хәм процедуралар орынланбайды. Олар ўақытша системадан шығарылынады хәм жәрдемши программада идентификатор ретинде пайдаланылады.

**11-программа.** Параметрлердің хәрекетте болыұ механизмин **Function, Procedure** операторларында пайдаланыұды мысалменен түсіндиремиз :

**1- блок.** Ушмүйешлик майданы Герон формуласы менен **Function F\_Sabc( A, B, C, Real ) : Real;** операторы жердемінде есапланылады ;

**2-блок.**  $ax^2 + bx + c = 0$  квадрат теңлемениң шешимлери **Procedure P\_KBT( A1, B1, C1, Var X\_1, X\_2 : Real);** процедурасы менен анықланылады ;

**3-блок.** Берилген еки санның үлкени хәм кишиси **Procedure P\_Max\_Min( AA, BB : Real; Var Max\_AB, Min\_AB);** операторы менен табылады ;

```
Program Prog11_DEMO_PARAMETR;
```

```
Var A, B, C, X1,X2,Sabc : Real;
```

```
Log : Boolean;
```

```
Us : Integer;
```

```
Function F_Sabc( A, B, C : Real) : Real;
```

```
Lfbel 1;
```

```
{ 1-Blok.} Var P : Real;
```

```
Begin Log := (A + B > C) And (A + C > B) And (B + C > A);
```

```
{ уш мүйешлик жасау шәртин тексеру }
```

```
If Log = True Then P := 0.5 * (A+B+C) Else Goto 1;
```

```
F_Sabc := Sqrt( P*(P-A)*(P-B)*(P-C));
```

```
1: End; { End F_Sabc }
```

```
Procedure P_KBT( A1, B1, C1: Real; Var X_1, X_2 :Real) ;
```

```
{ 2-Blok } Var D : Real;
```

```
Begin D := B1*B1 - 4*A1*C1;
```

```
If D < 0 Then Begin Writeln( ' шешимге ийе емес '); Exit End;
```

```
If D = 0 Then Begin Writeln( 'X1= X2 =', -B1/ (2*A1)); Halt( 0 ) End;
```

```
X_1:= (-B1 - Sqrt(D))/( 2*A1); X_2:= (-B1 + Sqrt(D))/( 2*A1);
```

```
End; { End P_KBT }
```

```
Procedure P_MaxMin(AA, BB : Real; Var Max_AB, Min_AB);
```

```
Begin If AA > BB Then Begin Max_AB:=AA; Min_AB:= BB End
```

```
Else Begin Max_AB:=BB; Min_AB:=AA End;
```

```
End; { End P_Max_Min }
```

```
BEGIN Writeln( ' Маселе шарты '); Read(Us);
```

```
Case Us Of
```

```
1 : Begin Read( A, B, C);
```

```
Sabc:= F_Sabc( A, B, C);
```

```
If Log=False Begin Then Writeln( ' ушмүйешлик жасалмайды ');
```

```
Exit; End;
```

```
Writeln( 'Sabc = ', Sabc :12:7); End;
```

```
2 : Begin Read( A, B, C);
```

```
P_KBT( A, B, C, X1, X2);
```

```

Writeln( 'X1 = ', X1:12:7, ' X2 = ', X2:12:7) End;
3 : Begin Read( A, B); P_Maxmin( A, B, X1, X2) ;
Writeln( 'Max_AB = ', X1 :12:7, ' Min_AB = ', X2 :12:7); End; END.

```

#### § 4.5. Функция хэм процедура операторларын пайдаланыў өзгешеликлери

1. **Function** операторын аңлатпада операнд ретинде пайдаланыўға болады. Алынған бирден-бир нәтийже, шақырылған орынға қайтарылады. Функция аты, аңлатпада **функция көрсеткиши** - деп айтылады.

2. **Procedure** операторын аңлатпада операнд ретинде пайдаланыўға болмайды. Программада процедураны орынлаў оны аты хэм параметрлери (ямаса, параметр-лерсиз) бойынша шақырып алынып орынланылады.

3. Функция операторында өткерилиўши параметрлер бирнеше болып, ал алынатуғын нәтийже биреў болады. Процедура операторында өткерилиўши параметрлер бирнеше хэм нәтийжелерде бирнеше параметрлерге жазылады хэм қайтарылынады.

Мәселен, үшмүйешликтиң берилген үш тәрәпи **a, b, c** бойынша майданын Герон формуласы менен есаплаўды функция операторы жәрдемінде программаластырыў мақсетге муўапық болады. Себеби, алынатуғын нәтийже биреў, үшмүйешлик майданы. Мейли, экран бетинде үшмүйешликтиң төбелериниң координатлары бойынша, оны сызыў программасын дүзиўде (графикалық режимде) ҳеш бир есаплаў жумысы орынланбайды. Сонлықтан, үшмүйешликти сызыў функция операторына салыстырғанда процедура операторында аңсат орынланылады.

**12- программа.** Төменде берилген үш қосындыны, бир **PSUM** атамалы процедура арқалы есаплаўды орынлаў программасы :

$$S_q = \sum_{i=1}^{100} i^2 ; S_2 = \sum_{x=0}^{45} \text{Cos}2x ; S_3 = \sum_{n=5}^{15} a_n .$$

Қосындылар улыўма  $S = \sum_{k=m}^n f_k$  түринде жазылады. Бул жерде **f** - аңлатпа, **k** - индекс.

**Program Prog12\_SUMMA;**

```

Type Mas=Array [1 .. 25] Of Real;

```

```

Var N,I,X : Integer;

```

```

Summa : Real; A : Mas;

```

```

Procedure PSum(F,K : Real; M,N : Integer; Var Summa : Real);

```

```

Var L : Integer;

```

```

Begin Summa:=0; For L:=M To N DO Summa:=Summa+F End;

```

```

BEGIN { биринши сумманы есаплау блогы }

```

```

PSum(Sqr(10),I,1,20, Summa); Writeln(' S1 = ', Summa :15);

```

```

{ екинши сумманы есаплау блогы }

```

```

PSum(Cos(2*0.32),X,0,15, Summa); Writeln(' S2 = ', Summa :15);

```

```

{ ушинши сумманы есаплау блогы }

```

```

For N:=5 To 25 DO Begin A[I]:=I+1;

```

```

PSum(A[N],N,5,25, Summa); End; Writeln(' S3 = ', Summa :15); END.

```

Сәйкес қосындыларды есаплаўды процедура операторлары төмендегише :

1. **Psum ( Sqr(I), I, 1, 100, S1);**

2. **Psum ( Cos(2\*X)), X, 0, 45, S2);**

3. **Psum ( A[n], N, 5, 15, S3);**

Бул процедура операторларында **F, K** формал параметрлер бирдей сәйкесликте болыўлары тийис.

#### § 4.6. Рекурсив жәрдемши программа

Мәселелерди шешиў алгоритмин оптимизациялаўда жәрдемши программалық үлестин операторлар бөлиминде, сол жәрдемши программаның өзин-өзи қайтадан шақырып алып орынлаўды иске асырыўға болады. Есаплаў процессин усылай шөлкемлестириў **рекурсив жәрдемши программа** менен орынланады. Оны сол ўақытта қолланыў пайдалы, егерде тийкарғы мәселени бирнеше киши ишки мәселелерге бөлип алып хэм олар дәслепки қойылған мәседедей алгоритмлик дүзилiske ийе болса. Мысалы, төмендеги дүзилген программада **PrintA** процедурасы өзин-өзи шақырып, экранда шексиз мәртебе ” **XX1 A'SIR JASLARI** ” сөзи пайда болады.

Мысал: **Program Prim RECURISB;**

```

Procedure PruntA;

```

```

Begin Writeln (' XX1 A'SIR JASLARI'); PrintA;

```

```

{ PrintA - процедурасы өзин-өзи шақырады } End;

```

```

BEGIN Readln; PrintA END.

```

Рекурсив жәрдемши программаны пайдаланыўда, одан кәлеген ўақытта шығып кетиў мүмкиншилиги есабға алынады. Оның ушын қосымша өзгеріўшиниң мәніси алдын-ала берилген шамадан киши ямаса үлкен болыўы, ямаса мәніси киширейип барып, нольге тең болған шәртлер қаралады.

Мысал : **Var K : Integer;**  
**Procedure BIXOD( N : Integer);**  
**Begin Write( ' ABC ', ' '); If N>=1 Then BIXOD(N-1) End;**  
**BEGIN Read( K ); BIXOD( K ); END.**

Мысалда, **K=1** болса экранда **ABC**, сәйкес **K=2** болса, **ABC ABC** хәм **K=3** болса, **ABC ABC ABC** символлары экран бетинде пайда болады.

**13- программа.** Рекуррент формула  $\{ f_{i+1}=f_i + f_{i-1}; f_0 = 0; f_1 = 1 \}$  Фибоначчи санлары избе-излигин ( **0, 1, 1, 2, 3, 5, 8, 13, 21, 34, . . .** ) дүзиў программасы :

**Program Prog13\_FIBOCHI;**  
**Var K : Integer;**  
**Function Fib( N : Integer ) : Integer;**  
**Begin If N=0 Then Fib :=0 Else If N=1 Then**  
**Fib := 1; Fib := Fib( N-2)+ Fib( N-1) end;**  
**BEGIN For K:= 2 To 112 Do Writeln( Fib(K)**  
**END.**

#### § 4.7. Сыртқы жәрдемши программаны дағазалаў

Жәрдемши программаны пайдаланыўда, биринши жәрдемши программа орынланыўында, еле анықланбаған (биринши жәрдемши программадан кейин дүзилген) екинши жәрдемши программаны шақырып алып есаплаў процессин орынлаўды алгоритмлестириў керек болады. Бул усыл сырғы жәрдемши программа менен орынланылады. Сыртғы жәрдемши программалар, алды бурын арнаўлы директива **FORWARD** (алдынан \ вперед ) хызметши сөзи менен дағазаланылады.

Мысал: **Procedura CORRECT; Forward;**  
**Procedure NOLOG ( Var Zap1: Real); Forward;**  
**Function MAXMIN (Var A, B, C, D : Integer): Integer ; Forward;**

**14-программа.** Еки тәрепи **a, b** хәм олар арасындағы **C** мүйеши бойынша үшмүйешлик майданын **S= 0.5 \* a \* b \* Sin(C)** ; формуласы менен есаплаў программасы :

**Program Prog15\_FORWARD;**  
**Var A,B : Real;**  
**Uc : Real;**  
**Procedure PUgolR( Z: Real; Var Ugz : Real ); FORWARD;**  
**Procedure Psabc( X,Y,Z : Real; Var Sabc : Real);**  
**Begin PUgolR(Z, W);**  
**Sabc:= 0.5 \* X\*Y\*Sin(W) End;**  
**Procedure PUgolR( Z: Real; Var Ugz : Real );**  
**Begin Ugz := (Pi\*Z) / 180 End;**  
**BEGIN Writeln( ' ушмүйешликтин элементлери A, B, мүйеш C = ? ');**  
**Readln( A, B, Uc ); PSabc(A, B, Uc, Sabc );**  
**Writeln( ' Sabc =', Sabc : 10 :6) END.**

Программада градусды радианға өткеріў процедурасы **PUgolR**, майданды есаплаўшы **PSabs** дан кейин шөлкемлестирилген. Сонлықтан, **PUgolR** процедурасы **Forward** сөзименен алдын дағазаланған.

#### § 4. 8. SYSTEM модулиниң қосымша процедуралары хәм функциялары

ТП системасы орталығында , хәрбир программа орынланыўы ўақтында **SYSTEM** модули курамына киритилинген операторлар, процедура хәм функциялардан тиккелей пайдаланылады. Мысалы, стандарт математикалық функциялар (11-кесте, §2.5.). Сонлықтан, **'USES SYSTEM;'** хызметши сөзин программада жазыў шәртли емес.

##### § 4.8.1. Программа орынланыўын басқарыў процедуралары

Бул топар процедуралары программа орынланыўын даўамлаўды ямаса тамамланыўын әмелге асырады.

**1. Delay( T: Longint);** - процедурасы программа орынланыўын **T** милли-секунд токтатып турады : **Delay (1000 + 2000); { 3000 = 3 сек }**

**2. Exit;** Процедурасы орынланып атырған блоктан орталыққа шығыўды әмелге асырады. Егерде, ол процедура ямаса функция операторы ишинде болса, ТП орталығына өтиледі. Егерде, **Exit** бас программада

пайдаланылған болса, программа жұмыс іслеуін тоқтатып **NC** айнасына өтиледі : **D:= B\*B - 4\*A\*C; If D < 0 Then Exit ;**

**3. Halt [(Code : Byte)];** - процедурасы программа орынланыуын тоқтатады. Басқарылуы ТП орталығына өтиледі, егерде “ ... .pas ” - файлында пайдаланылса. **NC** айнасына (ямаса, **MSDOS**) өтиледі, егерде “ ... .exe “- файлы орынланса. **Code** - программаның тамамланыуы коды : **If A<0 Then Halt (15) Else Y=Ln(A);**

**4. RunError[(Code : Word)];** - процедурасы, программа орынлауын тоқтатып, орынланыуы уақытында кеткен қәтеликти анықлайды (генерациялайды). **Code** параметринің қәтелик номери хәққында мәлимлеме экран бетине шығарылады :

**If B=0 Then RunError(0) Else Y :=A/B;**

#### § 4.8.2. Мағлыұматлар түрін түрлендириуши функциялар

Бул функциялар берилген мағлыұматлар түрін бир скаляр түрден екінши түрге түрлендиреди.

**1. Chr(N: Byte) : Char;** - функциясы **ASCII** коды бойынша пүтин санға тең болған номерлер менен сәйкес келетуғын белгилерди (символларды) анықлайды (**N**-символлардың сәйкес қатар номерлері):

**For N:=0 To 52 Do**

**BEGIN I:=N ; Writeln( I, ' = ', Chr(I), I+1, ' = ', Chr(I+1),  
I+2, ' = ',Chr(I+2),I+3, ' = ',Chr(I+3), I+4, ' = ',Chr(I+4) END;**

программалық үлес бес бағана бойынша, **ASCII** коды қатар номерлерин хәм сәйкес символларды экран бетине шығарады. Егерде **N > 255** болса, функция орынланбайды.

**2. Ord(Ch: Char) : Byte;** - функциясы санақлы түрде берилген символлы мағлыұматлар-дың **ASCII** кодына сәйкес, қатар номерлерин анықлайды.Бул жерде **Ch**-символлы мағлыұмат : **For Ch := 'A' To 'Z' Do Writeln(Ch :2, ' = ', Ord(Ch));**

программалық үлес **A .. Z** шекемги латын хәриплериниң **ASCII** кодына сәйкес код номерлер қатарын анықлайды.

**3. Round(X: Doubly) : Longint;** - функциясы берилген хәқыйқый түрдеги санды, оған жақын пүтин санға дөңгелеклейди :

**Writeln( натийже =', Round(2.8) ); натийже=3**

**Writeln( натийже =', Round(-3.7) ); натийже=4**

**Writeln( натийже =', Round(6.5) ); натийже=7**

**4.Trunc(X:Real):Longint;** - функциясының мәниси берилген хәқыйқый санның пүтин үлесине тең болады. Егерде **X>=0** болса, онда **X** дан киши ямаса оған тең, ал **X<0** болса, онда **X** дан үлкен ямаса оған тең пүтин санлар алынады :

**Writeln( натийже =', Trunc(6.783) ); натийже=6**

**Writeln( натийже =', Trunc(0.583)); натийже=0**

**Writeln( натийже =', Trunc(-3.83) ); натийже=-3**

#### § 4.8.3. Санақлы мағлыұматларды түрлендириуши функция хәм процедуралар

Бул топардағы процедура хәм функциялар, хәқыйқый санлы мағлыұматлар түринен басқа, қәлеген скаляр түрдеги мағлыұматларды түрлендиреди.

**1. Dec(Var X [; N : Longint]);** - процедурасы берилген **X** пүтин санды **N**-бирликке азайтады. Егерде, **N**-көрсетилинбесе **X** саны **1** бирликке кемейеди:

**Writeln( натийже=',Dec(26,9)); натийже=17**

**Writeln( натийже=',Dec(26)); натийже=25**

**Writeln( натийже =', Dec(-26 , 5) ); натийже=-31**

**2.Inc(Var X[;N : Longint]);** Процедурасы берилген **X** шамасын **N** бирликке арттырады. Егерде, **N** көрсетилинбесе, **X** саны **1** ге артады.

**Writeln( натийже=',Inc(26,9)); натийже=35**

**Writeln( натийже=',Inc(26)); натийже=27**

**Writeln( натийже=',Inc(-26 , 5) ); натийже=-21**

**3. Odd(X: Longint): Boolean;** Функциясының мәниси, егерде **X** тақ сан болса, онда мәниси **True**, ал **X** жұп сан болса **False** болады:

**Writeln( натийже=',Odd(2\*36-1)); натийже=True**

**Writeln( натийже=',Inc(2\*31)); натийже=False**

**4. Pred(X);** Функциясы, берилген **X** мағлыұмат элементинен алдында жайласқан элементи анықлайды:

**Writeln( натийже=',Pred(56)); натийже=55**

**Writeln( натийже=',Pred(Chr(70)); натийже=E**

**Writeln( натийже=',Pred( Ord('F'))); натийже=69**

**5. Succ(X);** Функциясы, берилген **X** мағлыұмат элементинен кейингисин анықлайды:

**Writeln( натийже=',Succ(56)); натийже=57**



**X :=Lo(\$771B);**

**натийже= \$001B**

**6.Swap(X);** Функциясы, аргументтің киши хәм үлкен байтлар орнын алмастырады. Бул жерде, **X : Integer** ямаса **Word** түріндегі аңлатпа.

**X:=Swap(\$8968);**

**натийже =\$6889**

**7.KeyPressed;** Функциясының мәнісі **True** болады, егерде клавиатурада қалған бір түйме басылса, кері жағдай да **False** болады.

**Var I : Integer;**

**BEGIN I:=0; Repeat I:=I+1 Until KeyPressed;**

**Writeln (Түйме басылғанша цикл, I:10, `мартебе кайталанды`) END.**

#### § 4.8.7. Тосынарлы санларды анықлау функциялары.

Бул функциялар итималықлар теориясында мәселелерді шешуге көбірек пайдаланылады.

1. **Random;** - функциясы **0 .. 0,99** аралығындағы тосынарлы санларды анықлайды. Нәтиже түрі хақықый сан болады.

**For I := 1 To 3 Do Writeln ( ' Натийже=', Random(I) : 8 : 5 );**

**Натийже = 0.06919**

**Натийже = 0.78539**

**Натийже = 0.17919**

2. **Random(N :Integer);** - функциясы **0..N** аралығында тосынарлы пүтин санларды генерациялайды.

**For N := 1 To 3 Do**

**Writeln(Random(10) , e);** нәтиже =7 2 9

3. **Randomize;** Функциясы, инициализациялау уақытында тосынарлы санлар генераторын жаңа база менен алмастырып турады (генерациялайды). Яғнай, тосынарлы санлар ишинен «жүдә тосынарлы санларды» анықлайды.

Мысал: **134** ге шекемгі цифрлар ишинен тосынарлы **15** цифрды сайлап алыу программа үлеси :

**Program Prim\_Randomize;**

**Var I : Integer;**

**BEGIN For I:=1 To 15 Do**

**Begin Write (Random(134): 4); RANDOMIZE End END.**

#### Алған билимди тексеруу ушын сораулар

1. Жәрдемши программалар қандай мақсет ушын пайдаланылады ?
2. Бас программадан, жәрдемши программаның өзгешеликтері нелерде ?
3. Функция ямаса процедура операторының қайсысын, қандай есаплау процессінде пайдаланыу қолайлы болады ?
4. Функция хәм процедура операторын орынлау қалай әмелге асырылады?
5. Процедура хәм функция операторларын орынлау айрымашылықтары неде?
6. Жәрдемши программада формаль параметрлердің өзлестиретуғын мәніслери қалай дағазаланады ?
7. Жәрдемши программада формаль параметрлердің дәслепки мәніслерин анық шамалар түрінде көрсетип бериуге болама ?
8. Жәрдемши программада хәм бас программада бир атамалы объектлерди пайдаланыуға болама?
9. Жәрдемши программада формаль параметрлер дизими көрсетилинбесе, онда жәрдемши программа менен бас программа арасында мағлыұматлар қалай алмасланылады?
10. Жәрдемши программаның формаль параметрлери менен орынланыушы хақықый параметрлер арасында, мағлыұматлардың берилиуінде қандай сәйкесликлер орнатылған болыулары керек ?
11. Параметрлердің берилиуінің локаль хәм глобаль түрлерин қалай түсіндиресиз?
12. Рекурсив жәрдемши программа қандай есаплау жұмысларында пайдаланылады ?
13. Жәрдемши программадан сыртқы, рекурсив (жанапай) жәрдемши программаларды дағазалауда қандай хызметши сөз пайдаланылады хәм оның орынланыуын түсіндириң.
14. **SYSTEM** модулинің қандай қосымша процедуралары хәм функциялары бар хәм олар не мақсетте пайдаланылады

#### Өзлестирілген билимди беккемлеуге арналған шынығыулар !

4.1. Жәрдемши программа функция операторын пайдаланып, Берилген аңлатпа мәнісин есаплау ушын программа дүзиң. Өзгеруушилер мәніслерин өзлериң сайлап алың.

а) 
$$x = \frac{\sqrt{SZ^3 + qX^2 + tX + 1}}{1 + e^{qZ^2 + tZ - 1}};$$

$$b) Y = \frac{a_1x^2 + b_1x_1 + c_1}{a_2x^2 + c_2} + \frac{a_3x^2 - b_4x}{a_4x^2 + b_4c - c_1};$$

$$c) Z = \frac{v_1 + v_2 + v_3}{3}, \quad (\text{бул жерде } v_1, v_2, v_3 \text{ сәйкес радиустары } r_1, r_2, r_3 \text{ болған шардың көлемлери})$$

4.2. Берилген аңлатпалардың мәнісін есаплау үшін функция операторын пайдаланып жәрдемши программа дүзиң

$$Y = x^3 - x^2 + x - 0.4 \quad \text{хәм} \quad Z = Y / (1 + \frac{Y}{1+Y});$$

Дүзилген жәрдемши программа  $f(x)$  функция мәнісін есаплау үшін пайдаланылады:

$$f(x) = \begin{cases} Y + 1, \\ Z^2 - 1, \end{cases}$$

Бул жерде  $X$  дың мәніслери  $X_1, X_2, \dots, X_{10}$  аралықта өзгереді ( $X_1 = -0.6, X_i = X_{i-1} + 0.3; i = 2, 3, \dots, 10$ ). Берилген аңлатпалардың мәнісін есаплау үшін функция операторын пайдаланып жәрдемши программа дүзиң хәм Дүзилген жәрдемши программа  $f(x)$  функция мәнісін есаплау үшін пайдаланылады:

**Program Prim4\_4;**

```

Type Mas=Array [1 .. 6] Of Real;
Const N : Array [1 .. 6]
Of Real = (-2.5, 3.5, -3.5, 6.0, -7.5, 8.0 );
Function SF(C1, C2 : Integer; N : Real);
Var X :real;
Begin X:=Exp(0);
For I:=C1 To C2 Do X := X+N[I];
SF:=X End;
BEGIN Y:=SF(N, 3, 6)
Writeln( ' Y= ', Y:14) End;

```

**Program Prim4\_5;**

```

Var X, Y, Z: Integer;
Function B1( T : Integer): Integer; Forward;
Function B( Y : Integer): Integer;
Begin Y:= Y+2*Sqr(Y); Y:=B1(Y);
B:= (Y-Sqr(Y)) End;
Function B1( T : Integer): Integer;
Var R : Integer;
Begin R:=T + T*T*T -3;
B1:= (R-Sqr(R)) End;
BEGIN X:=0; Z:=B(X);
Writeln( 'X= ', Z:14) End.

```

**Program Prim4\_6;**

```

Function F(X,Y: Integer) : Real;
Begin If X>=Y Then F :=(X+Y)/2
Else F:=F((X+2),Y-1) + F((X+1),Y-2) End;
BEGIN Write( 'F(1,10) =', F(1,10):15:1) End.

```

4.7. Процедура операторын пайдаланып аңлатпалардың мәніслерін есаплау үшін программа дүзиң. Өзгериушилердің мәніслерін өзлериң сайлап алың:

$$X = (Sh^2 a + Sh(a - b)) / Shb + Sh(a^2 - b^2); \quad \text{бул жерде } Shx = (e^{2x} - 1) / (2e^x)$$

4.8. Процедура операторын параметрлерсиз шөлкемлестирип, тууры мүйешлик координатлар  $X$  хәм  $Y$  ( $X > 0$ ) бойынша поляр координаталарын  $r = \sqrt{x^2 + y^2}$  хәм  $f = Arctg(y/x)$  есаплау программасын дүзиң.

4.9. Процедура операторы жәрдемінде аңлатпа мәнісін есаплау үшін программа дүзиң.  $U = e^{x_1 + x_2} - e^{y_1 - y_2}$  бул жерде  $x_1, x_2$  лер  $ax^2 + bx - 1.5 = 0$  теңлемесиниң шешимлери:  $y_1, y_2$  лер  $2y^2 - y + c = 0$  теңлемесиниң шешимлери. Процедура аркалы коренлери табылады. Егерде жормал коренлер болса, онда олар нолге тең болсын.

4.10. Симпсон формуласын пайдаланып анық интегралды есаплау программасын **Function** хәм **Procedure** операторлары жәрдемінде дүзиң:

$$\int_a^b f(x)dx = \frac{h}{3}[(f_0 + 4(f_1 + f_3 + \dots + f_{n-1})) + 2(f_2 + f_4 + \dots + f_{n-2}) + f_n], \text{ бул жерде}$$

$f_i = f(a + ih), h = (b - a) / n; n = 2^k; k=10$ . Дүзилген программаны  $\int_0^{\frac{\pi}{2}} \sin(x)dx$  менен тексерип көриң.

```

Program Prim4_11;
Uses CRT;
Var A, B, C : Integer;
Procedure P_abc( X, Y : Integer; Var Z : Integer);
  Begin Z:=X+Y+Z;
  Writeln( X, ' ', Y, ' ', Z, ' '); End;
BEGIN A:=5; B:=8; C:=1; Clrscr;
  P_abc(A, B, C);      Writeln(' C= ', C :5);
  P_abc(7,A+B,C);     Writeln(' C= ', C :5);
  P_abc(A*B,(A Div 2), C);  Writeln(' C= ', C :5);
  P_abc(A Shl 2, B Shr 3, C); Writeln(' C= ', C :5); END.

```

```

Program Prim4_12;
Uses CRT;
Var X, Z : Integer;
Procedure P_C1( Var X : Integer);Forward;
Procedure P_T1( A : Integer; Var V : Integer);
  Begin V:=Sqr(A); P_C1(V) End;
Procedure P_C1( Var X : Integer);
  Begin X:=Sqr(4*X) End;
BEGIN X:=2; P_T1(X,Z ); Clrscr;
  Writeln(' Z= ', Z :5);  END.

```

```

Program Prim12_DEMO_PARAMETR;
Type Mas=Array [1..10] Of Real;
  Var  M : Mas;
Function F( K : Mas) : Integer;
  Var J : Integer;
  Begin F:=0; For J:=1 To 5 Do
    F:=F + K[J] End;
  BEGIN M[1] := 1; For I:=2 To 5 Do
    M[I]:=1+M[K-1]; N:=F(M[I]); Writeln(' N= ', N:4) End;

```

#### Әдебиятлар

1. Абрамови В. Г. и др. Введение в язык паскаль. М. Наука 1988, 230с.
2. Азларов Т.Р., Аширов Ш.Ш. Информатика олимпиада масалалари. Т.: 1993, 72б
3. Бородич Ю.С. и др. Паскаль для персональных компьютеров. Справ. пособие. Минск, Выш. шк. БФ ГИТМП “НИКА” : 1991, 365с
4. Вирт Н. Систематическое программирование. М.Мир, 1977, 183с. Гритас Г. Программалаш асослари Т., Укитувчи 1990, 112б
5. Зуев Е.А. Язык программирования Turbo - Pascal 6.0, 7.0 М: Веста, Радио и связь, 1993 384с.
6. Информатика, Маръузлар матни. Т. Абу Райхан Беруний номидаги. ТДТУ 2000, 135б.
7. Кабулов В.К. Человек - ЭВМ. Т. ФАН., 1988, 144с.
8. Перминов О.Н. Язык программирование Паскаль . Справочник. М. Радио связь 1989, 128 с
9. Семашко О.Н. Язык программирование Паскаль. Справочник. М. Радио и связь 1989, 128с.
10. Шагин В.Ф. и др. Прогаммирование на языке паскаль М.Высшая школа, 1991, 142 с
11. Чернов Б. И. Программирование на алгоритмических языках. Бейсик, Фортран, Паскаль. М. Просвещение, 1991, 192с.
12. Файсман А. В. Профессиональные программирование на Турбо - Паскале. Т. 1992, 270с

