

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

Орындаған: 10а сынып Сәдірова Несібелі

10а Шакен Анель

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

2

1.Кіріспе

2.Тармақталу

3.Күрделі
шарттар

4.Циклдер

5.Шартты циклдер

6.Таңдау операторы

7.Графика

8.Функциялардың
графигі

9.Процедуралар

10.Рекурсия

11.Анимация

12.Кездейсоқ сандар

13.Функциялар

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

3

Тақырып 1. Кіріспе

Алгоритм – атқарушы орындайтын амалдардың тиянақты жоспары.

Алгоритм қасиеттері

- **дискреттілік:** жекелеген қадамдардан(командалардан) тұрады
- **айқындылық:** атқарушыға түсінікті болатын командалардан тұруы керек
- **белгілілік:** бастапқы деректері бірдей болған жағдайда нәтижесі де бірдей болады
- **нәтижелік:** әрекеттердің шектеулі санынан кейін белгілі бір уақытта қорытынды нәтиже алуымыз керек
- **жалпылық:** алгоритм әр түрлі алғашқы мәліметтер үшін әр түрлі нәтижелер беруі тиіс
- **дұрыстылық:** алғашқы мәліметтер әр түрлі болғандығына қарамастан барлық жағдайда дұрыс нәтиже береді

Программа дегеніміз -

- қандай да бір программалау тілінде жазылған алгоритм
- компьютерге қажетті командалардың жиынтығы

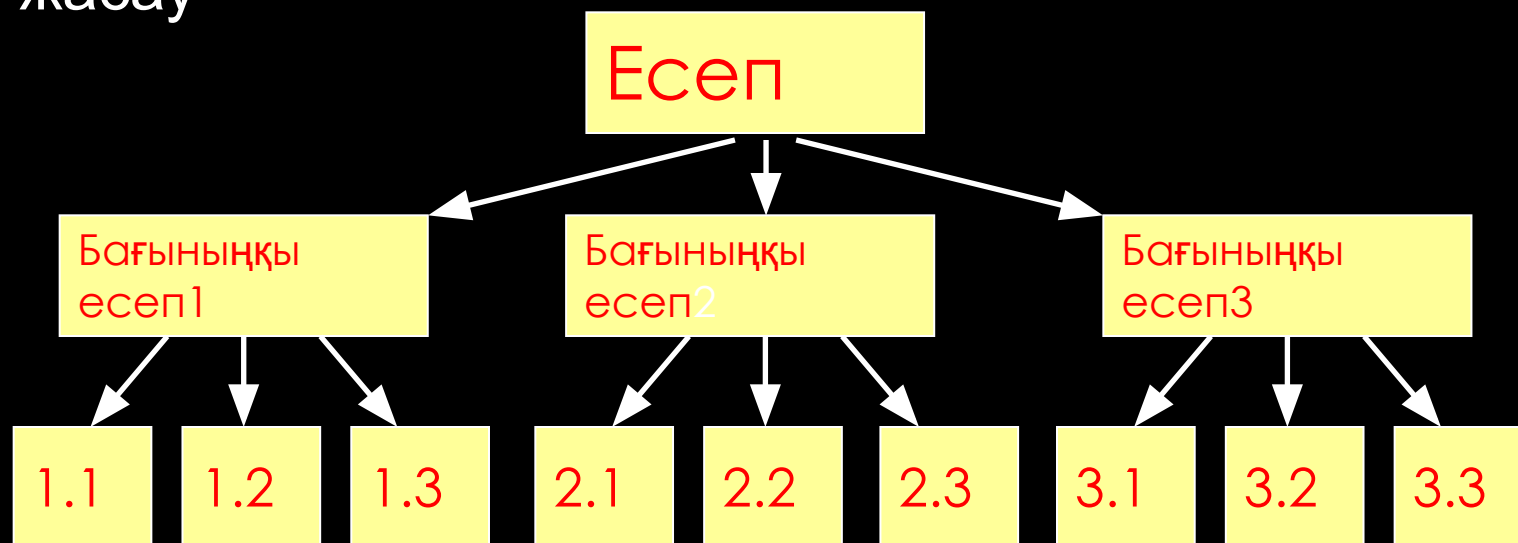
Команда – компьютер орындауға міндетті әрекеттің өрнектелуі.

- алғашқы мәліметтерді қайдан алу керек?
- олармен қандай амалдар орындау керек?

- Машинаға бейімделген (төменгі деңгейлі) - әрбір команда процессордың бір командасына сәйкес келеді (ассемблер)
- Жоғарғы деңгейлі тілдер – кәдімгі табиғи тілге (ағылшын тіліне) ұқсайды, адамның түсінуіне оңай, бір ғана компьютерге тәуелді болмайды.
 - *Жаңадан үйренушілерге*: Бейсик, ЛОГО, Паскаль
 - *Кәсіби*: Си, Фортран, Паскаль
 - *Жасанды интеллект тапсырмалары үшін*: Пролог, ЛИСП
 - *Интернет үшін*: JavaScript, Java, Perl, PHP, ASP

1970 – Никлаус Вирт (Швейцария)

- студенттерді оқытуға арналған тіл
- «жоғарыдан төменге қарай» программаларын жасау



- берілгендердің әр түрлі құрылымдары(массивтер, құрылымдар, жиымдар)

Программа қандай бөліктерден тұрады?₈

```
program <программаның аты>;  
const ...; {тұрақтылар}  
var ...; {айнымалылар}  
{ процедуралар мен функциялар  
}  
begin  
    ... {негізгі программа}  
end.
```

фигуралық жақшаның ішінде
орналасқан түсініктемелер
программамен өңделмейді

Программа қандай бөліктерден тұрады?

Тұрақты – өз атауы болатын, өзгермейтін шама, .

Айнымалы – өз атауы (жад ұяшығы) болатын, өзгертін шама.

Процедура – кейбір әрекеттерді сипаттайтын қосалқы алгоритм (мысалға шеңберді салу).

Функция – есептеулерді орындайтын қосалқы алгоритм (квадрат түбірді табу, \sin).

Программалар, тұрақтылар, айнымалылар атауы

10

Атауларды сипаттағанда

- бас әріптер мен кіші әріптердің арасында айырмашылық болмайды
- сандарды
Атаулар санмен басталмауы керек
- астын сызу белгісін _

қолдануға болады

Атауларды сипаттағанда

- орыс, қазақ әріптерін
- бос орындарды
- жақшаларды, +, =, !, ? белгілерін және т.б.

қолдануға болмайды

Қай атаулар дұрыс жазылған?

AXby R&B 4Wheel Вася “PesBarbos” TU154
[QuQu] _ABBA A+B

const

i2 = 45; { бүтін сан }

pi = 3.14; { нақты сан }

бүтін және бөлшек бөліктері нүктемен
ажыратылады

qq = 'Вася'; { символдар қатары }

орыс, қазақ әріптерін қолдануға болады!

L = True; { логикалық шама }

екі мән қабылдай алады:

- True (ақиқат, «иә»)
- False (жалған, «жоқ»)

Айнымалы – аты, типі және мәні болатын шама.

Айнымалының мәнін программаның жұмысы кезінде өзгертіп отыруға болады.

Айнымалылардың типі:

- integer { бүтін }
- real { нақты }
- char { бір символ }
- string { символдық тіркес }
- boolean { логикалық }

Айнымалыларды сипаттау (жад бөлу):

```
var a, b: integer;  
    Q: real;  
    s1, s2: string;
```

Айнымалының мәнін қалай өзгертуге болады?

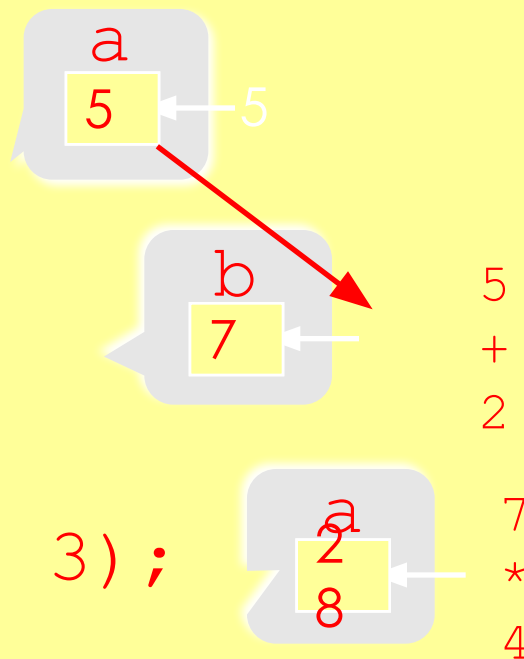
13

Оператор – жоғары деңгейлі программалау тілінің командасы.

Меншіктеу операторы айнымалының мәнін өзгерту үшін қолданылады.

Мысал:

```
program qq;  
var a, b: integer;  
begin  
  a := 5;  
  b := a + 2;  
  a := (a + 2) * (b - 3);  
end.
```



Жалпы құрылымы:

```
<айнымалы атауы> := <өрнек>;
```

Арифметикалық өрнектің құрамында

- тұрақтылар
- айнымалы атаулары
- арифметикалық амалдар белгілері:

+ - * / div mod

көбейту

бөлу

бүтінді
бүтінге бөлу

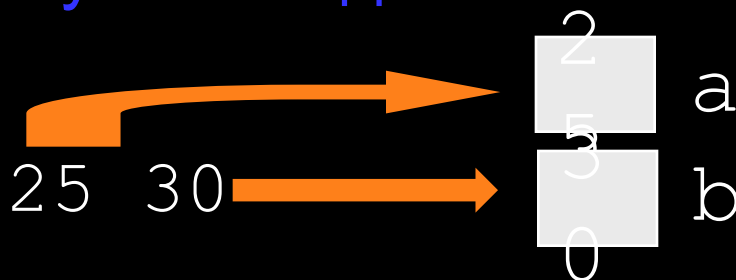
бөлуден
қалған
қалдық

- функцияларды шақыру
- жақшалар ()

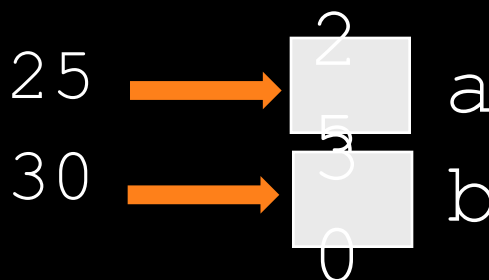
```
read ( a );      { a айнымалысының мәнін  
                  енгізу}  
read ( a, b );  { a және b  
                  айнымалыларының мәндерін  
                  енгізу}
```

Екі санды қалай енгізуге болады?

бос орын арқылы:



Enter арқылы:




```
write ( a ); { a айнымалысының мәнін  
шығару }
```

```
writeln ( a ); { a айнымалысының мәнін  
шығарып, келесі жолға  
көшу }
```

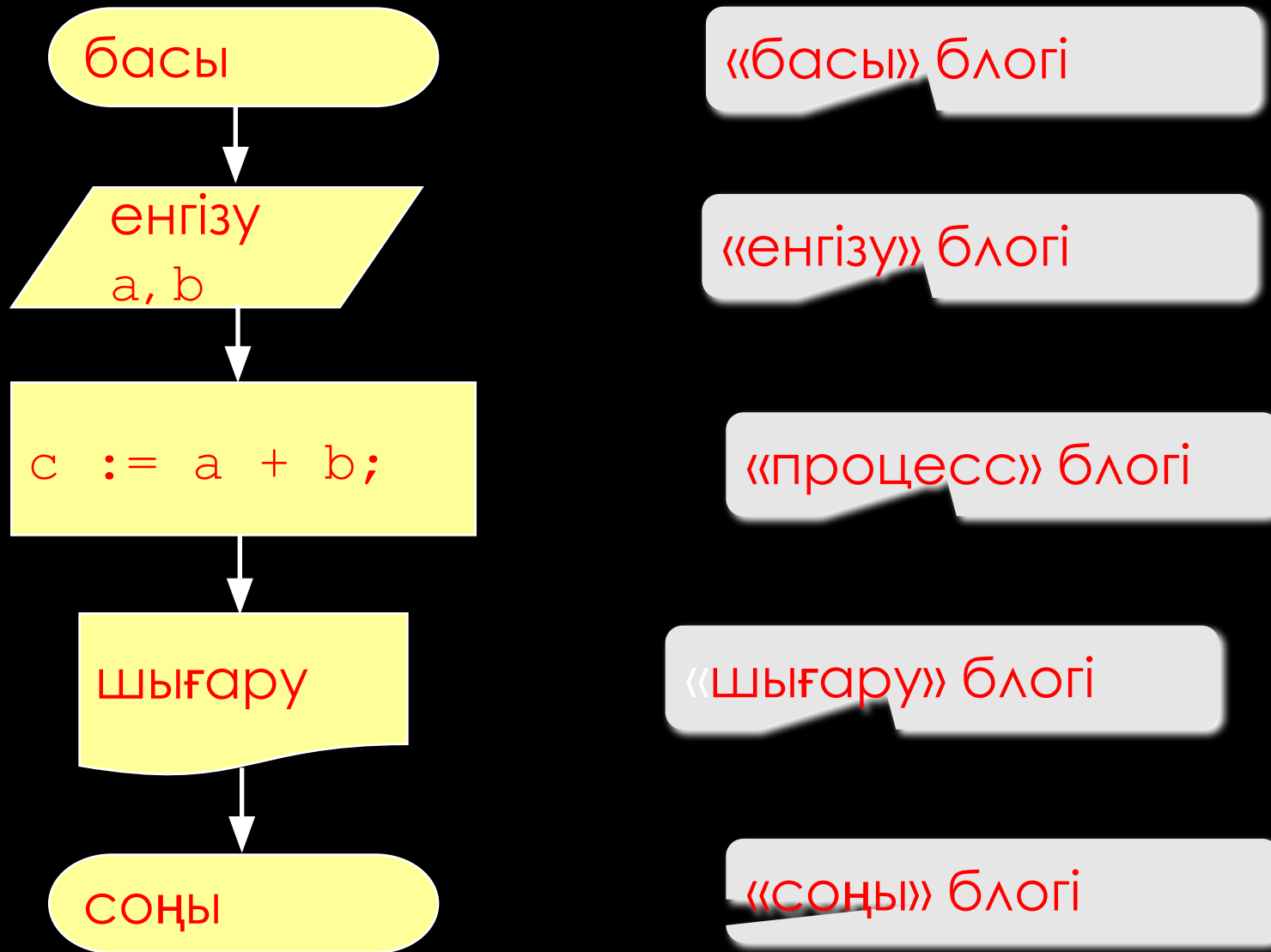
```
writeln ( 'Салем!' ); { мәтінді шығару }
```

```
writeln ( 'Жауап: ', c ); { мәтінді  
және c айнымалысының мәнін шығару }
```

```
writeln ( a, '+', b, '=', c );
```

СЫЗЫҚТЫҚ АЛГОРИТМНЫҢ БЛОК-СХЕМАСЫ

17



ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

18

Тақырып 2. Тармақталу

```
if <шарт> then begin
    {егер шарт орындалса, не істеу керек}
end
else begin
    {егер шарт орындалмаса, не істеу керек}
end;
```

Ерекшеліктері:

- *else*-тің алдында нүктелі үтір **болмауы** керек
- екінші бөліктің (*else ...*) болмауы да мүмкін (толымсыз түрі)
- егер блокта бір ғана оператор болса, *begin* мен *end* сөздерін жазбауға да болады

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

20

Тақырып 3. Күрделі шарттар

Күрделі шарт – бір-бірімен логикалық операциялардың көмегімен байланысатын бірнеше қарапайым шарттардан (қатынастардан) тұрады:

- `not` – ЕМЕС (терістеу, инверсия)
- `and` – ЖӘНЕ (логикалық көбейту, конъюнкция, шарттардың бір мезгілде орындалуы)
- `or` – НЕМЕСЕ (логикалық қосу, дизъюнкция, ең болмағанда бір шарттың орындалуы)
- `xor` – арифметикалық НЕМЕСЕ (екі шарттың екеуін емес, біреуін ғана орындау)

Қарапайым шарттар (қатынастар)

<

<=

>

>=

тең

=

тең емес

<>

Орындау тәртібі

- жақша ішіндегі өрнектер
- not
- and
- or, xor
- <, <=, >, >=, =, <>

Ерекшелігі – қарапайым шарттардың әрқайсысын жақшамен жабу.

Мысал:

```
      4      1      6      2      5  
if not (a > b) or (c <> d) and (b <> a)  
then begin  
    ...  
end
```


$a := 2; b := 3; c := 4;$ болған жағдайда ақиқат немесе жалған:

$\text{not } (a > b)$

True

$(a < b) \text{ and } (b < c)$

True

$\text{not } (a \geq b) \text{ or } (c = d)$

True

$(a < c) \text{ or } (b < c) \text{ and } (b < a)$

True

$(a < b) \text{ xor not } (b > c)$

FALSE

x-тің қандай мәндерінде шарт ақиқат болады:

$(x < 6) \text{ and } (x < 10)$

$(x < 6) \text{ and } (x > 10)$

$(x > 6) \text{ and } (x < 10)$

$(x > 6) \text{ and } (x > 10)$

$(x < 6) \text{ or } (x < 10)$

$(x < 6) \text{ or } (x > 10)$

$(x > 6) \text{ or } (x < 10)$

$(x > 6) \text{ or } (x > 10)$

$(-\infty, 6)$	$x < 6$
\emptyset	
$(6, 10)$	
$(10, \infty)$	$x > 10$
$(-\infty, 10)$	$x < 10$
$(-\infty, 6) \cup (10, \infty)$	
$(-\infty, \infty)$	
$(6, \infty)$	$x > 6$

"4": Айдың нөмірін енгізу арқылы, жыл мезгілінің атауын шығару.

Мысал:

Ай нөмірін енгіз:

4

көктем

Циклдік операторлар структуралық операторлар тобына жатады, өйткені құрамына бірнеше операторлар кіреді. Программда алдымен циклдің тақырыбы жазылады, онан кейін қайталанатын операторлар бөлімі – цикл денесі жазылады. Цикл денесі бір немесе бірнеше операторлардан тұруы мүмкін. Барлық циклдық операторлардың келесі ерекшелігі бар:

- а) Қайталанатын операторлар (циклдің денесі) бір – ақ рет жазылады;
- ә) Циклге тек қана басынан (тақырыбы) ғана кіруге болады;
- б) Циклің айнымалылары алдын – ала циклге дейін анықталуы тиіс;
- в) Цикл денесінде көшу операторын (`Go to`) қолданып программаның басқа бөліктеріне шығуға болады.
- д) Цикл ұйымдастырғанда, міндетті түрде циклден шығу шарты орындалатынын қарастыру керек, әйтпесе программа орындалуы циклге келгенде тоқтап тұрып қалады (программа «мәңгі циклденіп» қалады).

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

26

Тақырып 4. Циклдер

- Циклдік процесс ұйымдастыру үшін келесі операторлар қолданылады:
- 1.FOR – параметрлі циклдік оператор;
- 2.WHILE – алдын – ала циклдің орындал шартын тексеретін оператор;
- 3.REPEAT – соңында циклдің тоқтау шартын тексеретін оператор.

Pascal

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

28

Тақырып 5. Шартты циклдер

```
while <шарт> do begin
    {цикл денесі}
end;
```

Ерекшеліктері:

- күрделі шарттарды қолдануға болады:

```
while (a<b) and (b<c) do begin
    {цикл денесі}
end;
```

- егер цикл денесінде бір ғана оператор болса, `begin` мен `end` сөздерін жазбай кетуге болады:

```
while a < b do
    a := a + 1;
```


Ерекшеліктері:

- циклге кірген сайын шарт есептеліп отырады
- егер шарт циклге кіруде жалған болса, цикл бірде-бір рет орындалмайды

```
a := 4; b := 6;  
while a > b do  
    a := a - b;
```

- егер шарт түбінде жалған болмайтын болса, программаның жұмысы дағдарысқа ұшырайды (тоқтаусыз орындала береді)

```
a := 4; b := 6;  
while a < b do  
    d := a + b;
```

For-ны while-ға ауыстыру және керісінше

```
for i:=1 to 10 do begin
  {цикл денесі}
end;
```

```
for i:=a downto b do
  begin
  {цикл денесі}
end;
```

```
i := 1;
while i <= 10 do begin
  {цикл денесі}
  i := i + 1;
end;
```

```
i := a;
while i >= b do begin
  {цикл денесі}
  i := i - 1;
end;
```

for-ны while –ауыстыруға қашанда болады.

while-ды for-ға , циклдегі қадамдар саны алдын ала белгілі болған жағдайда ғана ауыстыруға болады.

Pascal

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

32

Тақырып 6. Таңдау операторы

Ерекшеліктері:

- case-тен кейін айнымалы атауы немесе бүтін типті (integer) арифметикалық өрнек

```
case i+3 of
  1: begin a := b; end;
  2: begin a := c; end;
end;
```

немесе символдық типті (char) арифметикалық өрнек

```
var c: char;
...
case c of
  'a': writeln('Бөкен');
  'б': writeln('Борсық');
  else writeln('Білмеймін');
end;
```

Ерекшеліктері:

- егер бір ғана операторды орындау керек болса, `begin` мен `end` сөздерін жазбай кетуге болады

```
case i+3 of
  1: a := b;
  2: a := c;
end;
```

- бірдей екі мәнді қоюға болмайды

```
case i+3 of
  1: a := b;
  1: a := c;
end;
```

Ерекшеліктері:

- бірдей әрекеттер орындалатын кезде қолданылатын мәндерді топтастыру қажет

тізбелеп
санап шығу

диапазон

аралас

```
case i of
  1:           a := b;
  2, 4, 6:     a := c;
  10..15:     a := d;
  20, 21, 25..30: a := e;
  else writeln('Қате');
end;
```

қорғаумен)

"4": Айдың нөмірін енгізе отырып, осы айдағы күндердің санын және енгізу кезінде жіберілген қателердің санын есептеп шығару.

Мысал:

Ай нөмірін енгіз:

-2

Ай нөмірін енгіз:

11

Бұл айда 30 күн бар.

Сіз 1 рет қате енгіздіңіз.

Ай нөмірін енгіз:

2

Бұл айда 28 күн бар.

Сіз 0 рет қате енгіздіңіз.

"5": Айдың нөмірі мен күннің нөмірін енгізіп, Жаңа жылға дейін қалған күндердің санын анықтау.

Мысал:

Ай нөмірін енгіз:

12

Күнді енгіз:

25

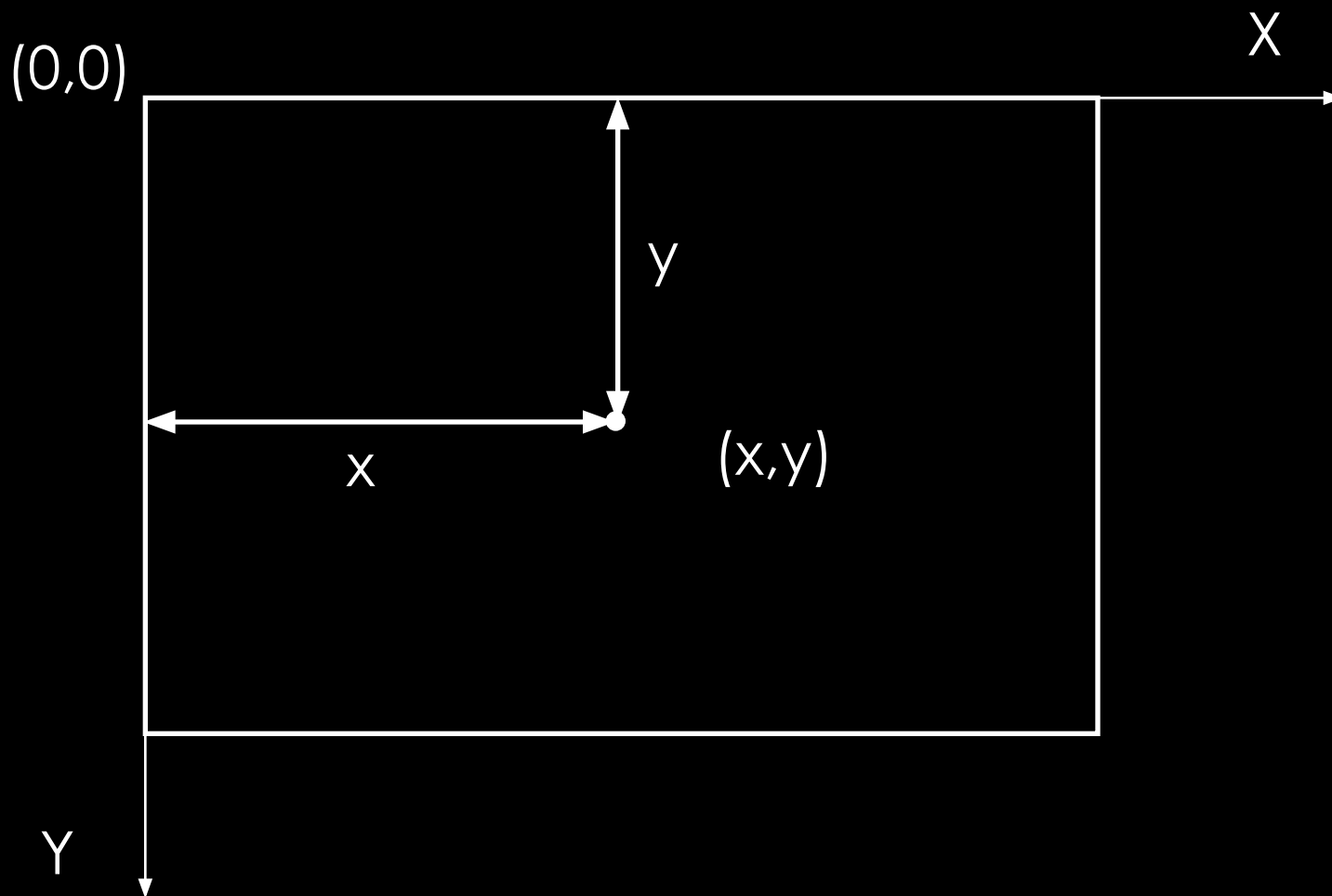
Жаңа жылға дейін 6 күн бар.

Pascal

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

37

Тақырып 7. Графика



Түстерді басқару

Сызықтың түсі мен қалыңдығы, нүктелердің түсі:

```
Pen ( 1, 255, 0, 0 );
```

СЫЗЫҚТЫҢ
ҚАЛЫҢДЫҒЫ

R(red)
)
0..255

G(green)
)
0..255

B(blue)
0..255

Бояудың түсі мен стилі:

```
Brush ( 1, 0, 255, 0 );
```

0 – өшіру
1 - қосу

R

G

B

Мәтін түсі:


```
TextColor ( 0, 0, 255 );
```

R

G

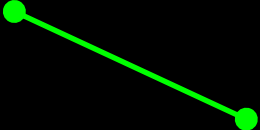
B

(x, y)



```
Pen (1, 0, 0, 255);  
Point (x, y);
```

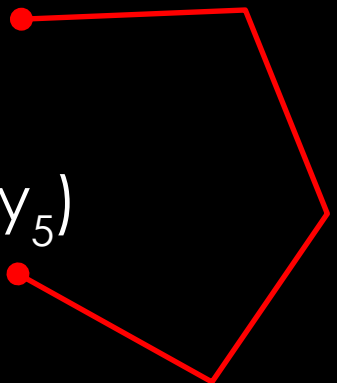
(x_1, y_1)



(x_2, y_2)

```
Pen (1, 0, 255, 0);  
Line (x1, y1, x2, y2);
```

(x_1, y_1)



(x_2, y_2)

(x_3, y_3)

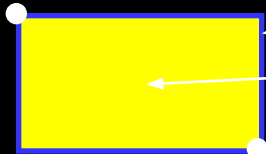
(x_4, y_4)

(x_5, y_5)

```
Pen (1, 255, 0, 0);  
MoveTo (x1, y1);  
LineTo (x2, y2);  
LineTo (x3, y3);  
LineTo (x4, y4);  
LineTo (x5, y5);
```

Фигуралардың ішін бояу

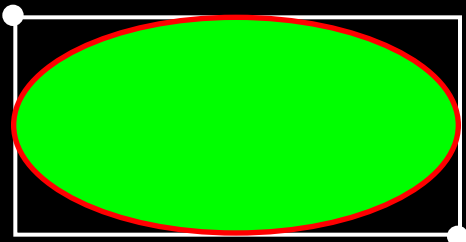
(x_1, y_1)



(x_2, y_2)

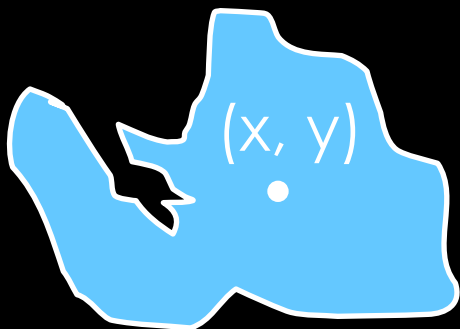
```
Pen (1, 0, 0, 255);  
Brush (1, 255, 255, 0);  
Rectangle (x1, y1, x2, y2);
```

(x_1, y_1)



(x_2, y_2)

```
Pen (1, 255, 0, 0);  
Brush (1, 0, 255, 0);  
Ellipse (x1, y1, x2, y2);
```



```
Brush (1, 100, 200, 255);  
Fill (x, y);
```

Pascal

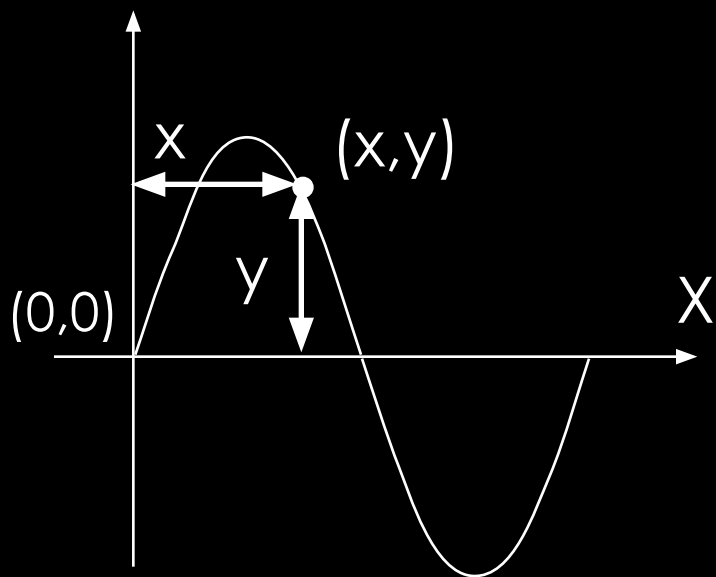
ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

42

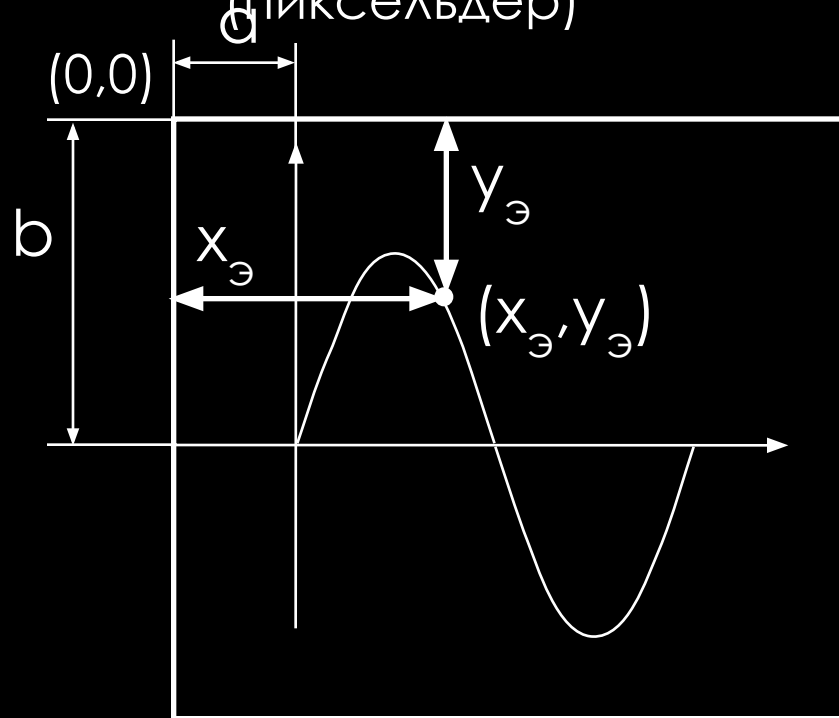
Тақырып 8. Функциялардың графигі

Координаталардың түрленуі

Координаталардың
математикалық
жүйесі



Координаталардың
экрандық жүйесі
(пиксельдер)



k — масштаб (экрандағы
жекелеген кесінді кескінінің
ұзындығы)

$$x_{\text{э}} = a + kx$$

$$y_{\text{э}} = b - ky$$

Нүктелерді қалай біріктіруге болады?

Алгоритм:

Егер бірінші нүкте болса
(x_3, y_3) нүктесіне өту
әйтпесе
(x_3, y_3) нүктесіне кесінді

әрекеттерді
ң нұсқасын
таңдау

Программа:

ЛОГИКАЛЫҚ
АЙНЫМАЛЫ

```
var first: boolean;  
    ...  
begin  
    ...  
    first := True;  
    while x <= xmax do begin  
        ...  
        if first then begin  
            MoveTo(xe, ye);  
            first := False;  
        end  
        else LineTo(xe, ye);  
        ...  
    end;  
end.
```

бастапқы мән

Pascal

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

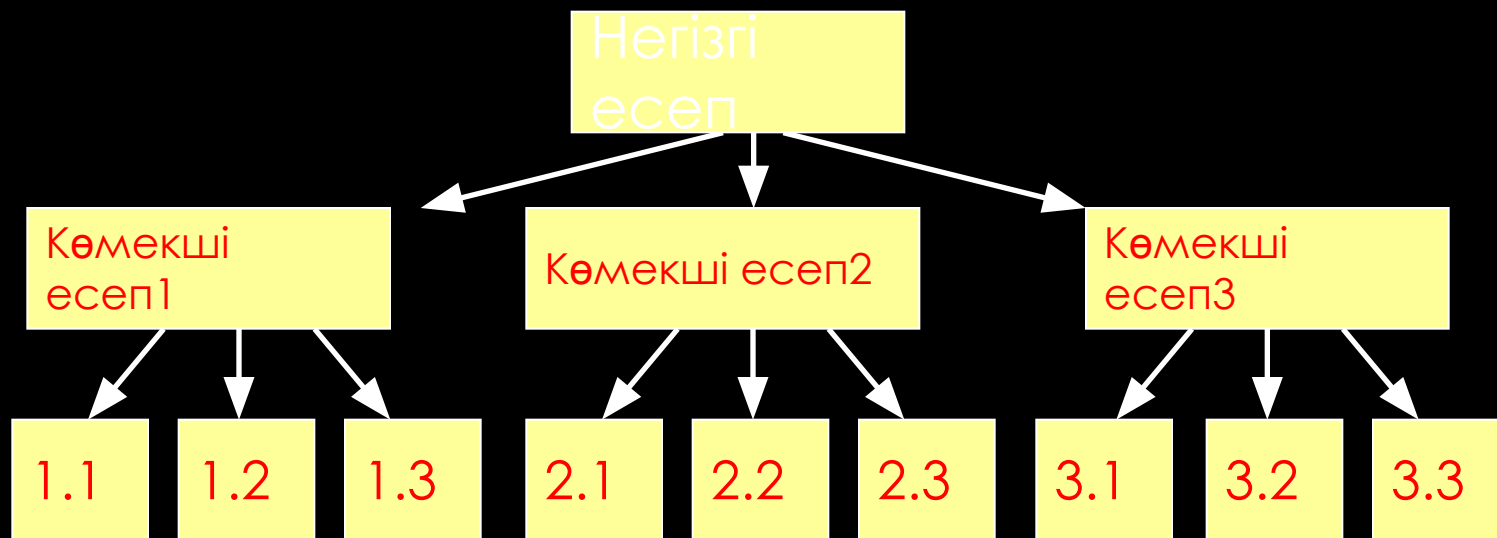
45

Тақырып 9. Процедуралар

Процедура – кейбір әрекеттерді орындау үшін қолданылатын көмекші.

Қолданылуы:

- программаның әр жерінде орналасқан бірдей әрекеттерді орындау
- программаны түсіну оңай болу үшін оны бірнеше қосалқы бөліктерге бөлу



Ерекшеліктері:

- әрбір формалды параметрдің қос нүктеден кейін типі көрсетілуі қажет

```
procedure A (x: real; y: integer; z: real);
```

- егер типтері бірдей параметрлер бірінен кейін бірі орналасса, олардың арасы үтірмен ажыратылады

```
procedure A (x, z: real; y, k, l: integer);
```

- процедураның ішіндегі параметрлер айнымалылар секілді қолданылады

Ерекшеліктері:

- процедурада қосымша айнымалы ретінде жергілікті (локальді) айнымалыларды пайдалануға болады, басқа процедуралар оған қатынай алмайды

```
program qq;
```

```
  procedure A(x, y: integer);  
    var a, b:  
        real;  
    begin  
      a := (x + y) / 6;  
      ...  
    end;
```

```
begin
```

```
  ...
```

```
end.
```

локальді
айнымалылар

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

49

Тақырып 10. Рекурсия

```
procedure Pifagor(x0, y0, a, L: real;  
                 N: integer);  
const k = 0.6;    { ұзындықтың өзгеруі }  
var x1, y1: real; { жергілікті айнымалылар }  
begin  
    if N > 0 then begin  
        x1 := x0 + L*cos(a);  
        y1 := y0 - L*sin(a);  
        Line (round(x0), round(y0),  
             round(x1), round(y1));  
  
        Pifagor (x1, y1, a+pi/4, L*k, N-1);  
        Pifagor (x1, y1, a-pi/4, L*k, N-1);  
    end;  
end;
```

егер N=0 болса, аяқтау

рекурсивті
шақырулар

Рекурсивті процедура деп өзін өзі
шақыратын процедураны айтады

```
program qq;
```

```
  procedure Pifagor(x0, y0, a, L: real;  
                  N: integer);
```

```
  ...  
end;
```

а бұрышы

баған ұзындығы

```
begin
```

```
  Pifagor (250, 400, pi/2, 150, 8);
```

```
end;
```

x_0

y_0

деңгейлер саны



Ағашты 30° оңға қалай еңкейтуге болады?

```
Pifagor (250, 400, 2*pi/3, 150, 8);
```


Pascal

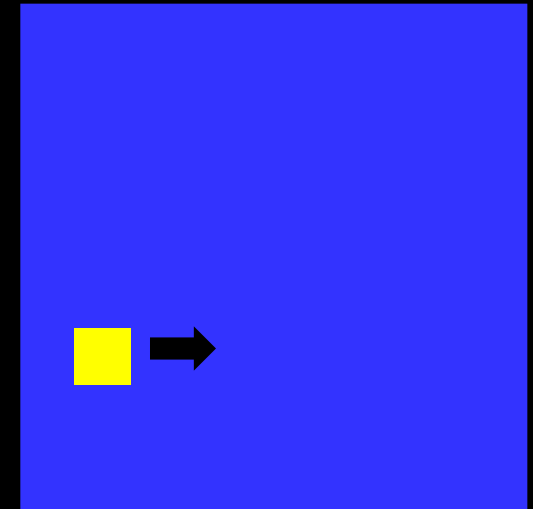
ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

52

Тақырып 11. Анимация

Анимация (ағылш. *animation*) – экрандағы кескінге жан бітіру.

Есеп: 400x400 пиксельді көк квадраттың ішімен 20x20 пиксельді сары квадрат солдан оңға қарай жылжып келеді. Программа өз жұмысын *Esc* пернесін басқанда немесе сары квадрат көк квадраттың оң жақ жетіне жеткен кезде тоқтатады.



Проблема: объектінің экранда қозғалуын қалай кескіндеуге болады?

Қосымша: объектінің қозғалысы (x,y) координаталарымен белгіленеді

Анимация қағидасы:

1. объектіні (x,y) нүктесінде саламыз
2. бірнеше миллисекундқа кідіру
3. объектіні өшіреміз
4. (x,y) координаталарын өзгертеміз
5. 1-қадамға көшеміз

Пернелермен басқару

54

Есеп: көк квадраттың ішінде орналасқан сары квадрат бағыттаушы пернелермен басқарылатындай болуы керек. Пернелердің коды:

солға – 37 жоғары – 38 Esc – 27

оңға – 39 төмен – 40

Проблема: қозғалыс бағытын қалай өзгертуге болады?

Шығарылуы:

```
if IsEvent            then begin
  Event ( k, code, i);
  if k = 1 then begin
    case code of
      37: x := x - 1; 38: y := y - 1;
      39: x := x + 1; 40: y := y + 1;
      27: stop := True;
    end;
  end;
end;
```

егер перне
басылған болса,
...

```
program qq;  
const rSun = 60;      { КҮННІҢ радиусы }  
      L  = 150;      { Жер орбитасының радиусы }  
      x0 = 200;     { КҮН центрінің координаталары }  
      y0 = 200;  
  
var x, y,             { Жердің координаталары }  
    k, code, i: integer; { Event үшін }  
    a, ha: real;      { айналу бұрышы, қадам }  
    stop: boolean;   { программаны тоқтату белгісі }  
procedure Draw(x, y: integer; flag:  
Boolean);  
begin  
    ...  
end;  
begin  
    ...  
end.
```

Pascal

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

56

Тақырып 12. Кездейсоқ сандар

Кездейсоқ көріністер: барлық жерде...

- тиынды лақтыру ("орел" немесе "решка")
- қардың түсуі
- броундық қозғалыс
- телефон байланысында болатын ақаулар
- радиоэфирдің шуы

Кездейсоқ сандар – бұған дейінгі сандардың ретін біліп отырсақ та, келесі санның қандай екені белгісіз болатын сандардың тізбегін айтады.

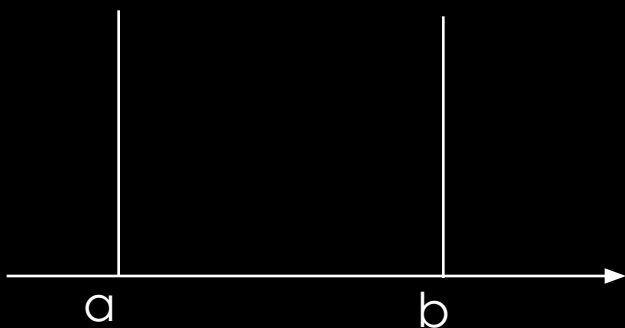
Проблема: компьютерде осыны қалай алуға болады?

Мүмкін болатын шешімдері:

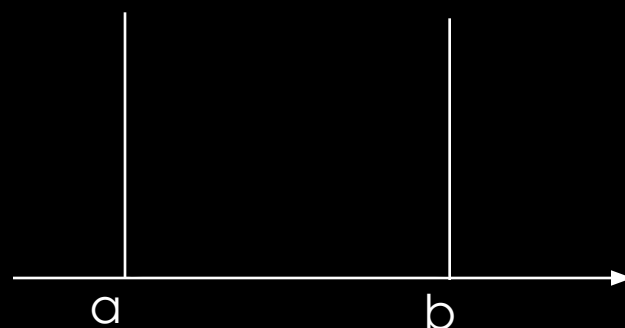
- шу ақауларының сыртқы көздерін пайдалану
- математикалық түрлендірулерді пайдалану

Ерекшеліктері:

- орналасу – бұл бір ғана санның емес бүкіл тізбектің сипаттамасы
- бірқалыпсыздардың саны – көп
- кез келген бірқалыпсыз орналасуды бірқалыпты орналасудың көмегімен алуға болады.



бірқалыпты орналасу



бірқалыпсыз орналасу

Паскальдегі кездейсоқ сандардың генераторы⁵⁹

[0,N] интервалындағы бүтін сандар:

```
var x: integer;
```

```
...
```

```
x := random ( 100 );    { интервал [0,99] }
```

[0,1] интервалындағы бүтін сандар:

```
var x: real;
```

```
...
```

```
x := random;           { интервал [0,1] }
```



Pascal

ПАСКАЛЬ ТІЛІНДЕ ПРОГРАММАЛАУ

60

Тақырып 13. Функциялар

Функция – нәтижесі қандай да бір мәнге ие болатын көмекші алгоритм (қосалқы программа).

Мысалдар:

- \sin , \cos , \tan есептеу
- өрнектерді күрделі формулалармен есептеу
- сұраққа жауап (жай сан ба, әлде жай сан емес пе?)

Неліктен? 

- программаның әр жерінде орналасқан бірдей есептеулерді орындау үшін
- функциялардың жалпыға бірдей кітапханаларын құру үшін

Ерекшеліктері:

- тақырыбы `function` сөзімен басталады

```
function Max (a, b: integer): integer;
```

- функцияда формальды параметрлер процедурадағыдай сипатталады

```
function qq (a, b: integer; x: real): real;
```

- айнымалы параметрлерді пайдалануға болады

```
function Max (var a, b: integer): integer;
```

- тақырыптың соңында қоспункттен кейін нәтиженің типі көрсетіледі

- функция `function Max (a, b: integer): integer;` пасады

Ерекшеліктері:

- жергілікті айнымалыларды жариялауға және пайдалануға болады

```
function qq (a, b: integer): float;  
  var x, y:  
    float;  
begin  
  ...  
end;
```

- нәтиже өзіндік меншік функциямен қатысты әйкес келетін айнымалыға жазылады; оны жариялаудың ҚАЖЕТІ ЖОҚ:

```
function Max (a, b: integer): integer;  
begin  
  ...  
  Max :=  
end;
```

НАЗАРЛАРЫҢЫЗГА РАХМЕТ!



