

# Паскаль тілінде программалау

- |  |   |
|--|---|
| 1. <a href="#"><u>Кіріспе</u></a>          | 7. <a href="#"><u>Графика</u></a>               |
| 2. <a href="#"><u>Тармақталу</u></a>       | 8. <a href="#"><u>Функциялардың графигі</u></a> |
| 3. <a href="#"><u>Күрделі шарттар</u></a>  | 9. <a href="#"><u>Процедуралар</u></a>          |
| 4. <a href="#"><u>Циклдер</u></a>          | 10. <a href="#"><u>Рекурсия</u></a>             |
| 5. <a href="#"><u>Шартты циклдер</u></a>   | 11. <a href="#"><u>Анимация</u></a>             |
| 6. <a href="#"><u>Таңдау операторы</u></a> | 12. <a href="#"><u>Кездейсоқ сандар</u></a>     |
|  | 13. <a href="#"><u>Функциялар</u></a>           |

# Паскаль тілінде программалау

## Тақырып 1. Кіріспе

# Алгоритм

---

**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

---

**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?

# Программалау тілдері

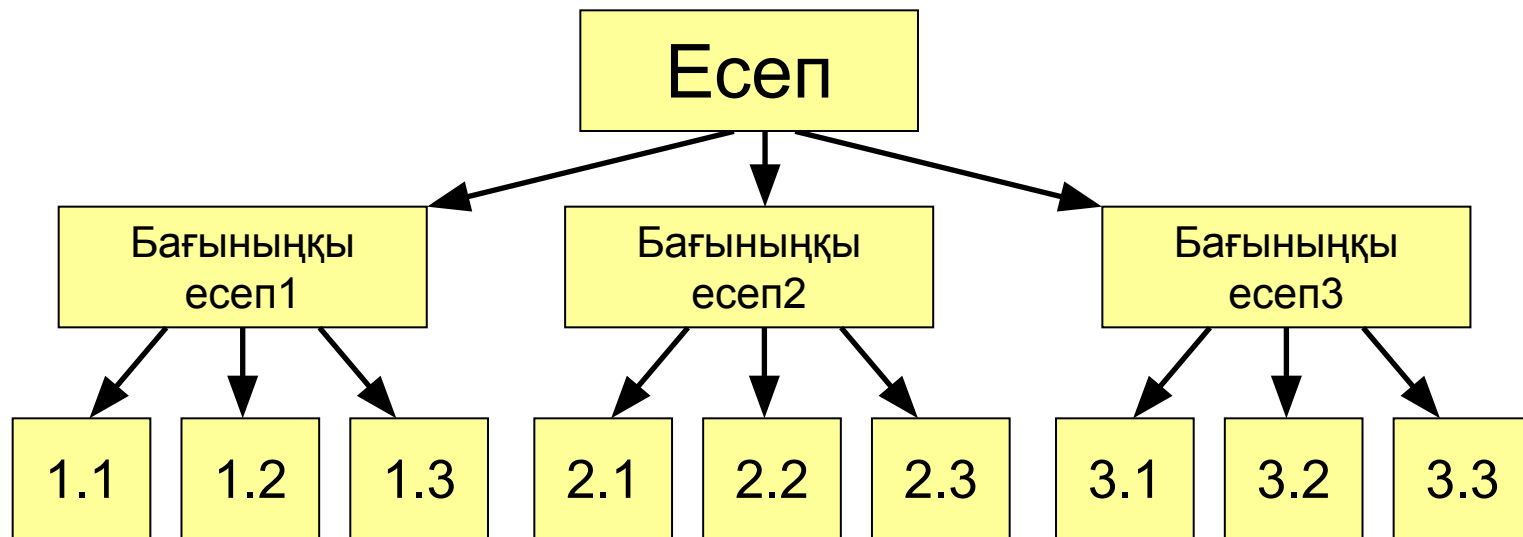
---

- **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

# Паскаль тілі

**1970** – Никлаус Вирт (Швейцария)

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



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

# Программа қандай бөліктерден тұрады?

```
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

## Атауларды сипаттағанда

- латын әріптерін (A-Z)

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

# Тұрақтылар

**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»)

# Айнымалылар

---

**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;
```

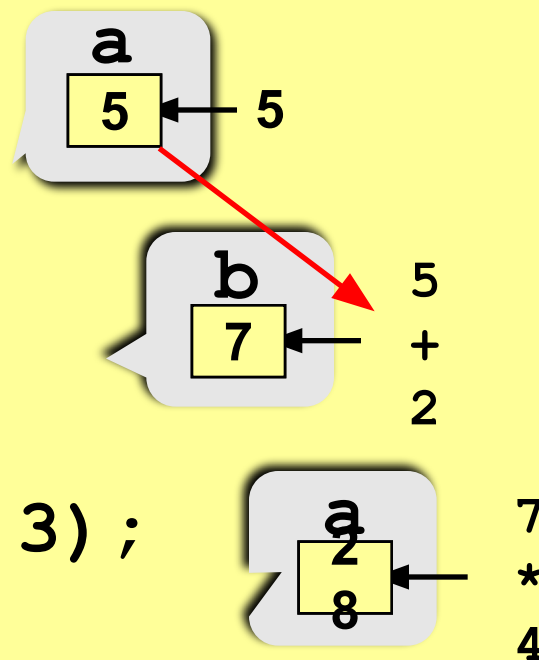
# Айнымалының мәнін қалай өзгертуге болады?

**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.
```



# Меншіктеу операторы

## 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 ( )

# Қай операторлар дұрыс жазылмаған?

```
program qq;  
var a, b: integer;  
    x, y: real;  
begin  
    a := 5;  
    10 := x;  
    y := 7,8;  
    b := 2.5;  
    x := 2*(a + y) ;  
    a := b + x;  
end.
```

айнымалы атауы := белгісінің  
сол жағында болуы керек

бүтін мен бөлшектің арасы  
нүктемен ажыратылуы керек

нақты мәнді бүтін  
айнымалыға меншіктеуге  
болмайды

# Амалдарды орындау тәртібі

- жақша ішіндегі өрнекті есептеу
- солдан оңға қарай көбейту, бөлу, **div**, **mod**
- солдан оңға қарай қосу және алу

2 3 5 4 1 7 8 6

**z := (5\*a\*c+3\*(c-d)) / a\*(b-c) / b;**

$$x = \frac{a^2 + 5c^2 - d(a+b)}{(c+d)(d-2a)}$$

$$z = \frac{5ac + 3(c-d)}{ab} (b-c)$$

2 6 3 4 7 5 1 12 8 11 10

**x := (a\*a+5\*c\*c-d\*(a+b)) / ((c+d)\*(d-2\*a)) ;**

# Екі санды қосу

---

**Есеп.** Екі бүтін санды қосып, қосындысын экранда шығару.

**Шығарылуы:**

```
program qq;  
var a, b, c: integer;  
begin  
    read ( a, b );  
    c := a + b;  
    writeln ( c );  
end.
```



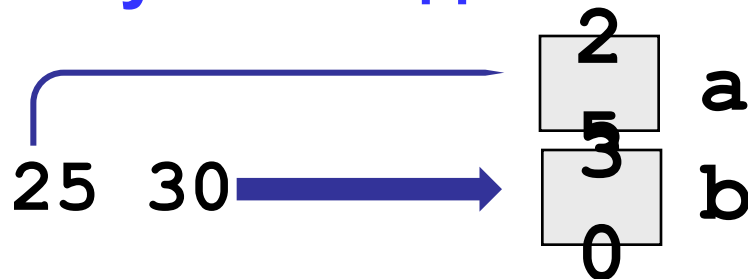
# Енгізу операторы

```
read ( a );      { a айнымалысының мәнін
                  енгізу}

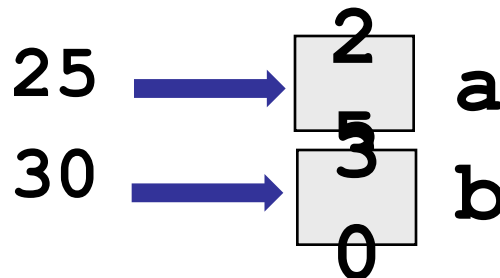
read ( a, b );   { a және b
                  айнымалыларының мәндерін
                  енгізу}
```

Екі санды қалай енгізуге болады?

бос орын арқылы:



*Enter* арқылы:



# Шығару операторы

---

```
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 );
```

# Шығару форматтары

```

program qq;
var i: integer;
    x: real;
begin
    i := 15;
    writeln ( '>', i, '<' );
    writeln ( '>', i:5, '<' );
    x := 12.345678;
    writeln ( '>', x, '<' );
    writeln ( '>', x:10, '<' );
    writeln ( '>', x:7:2, '<' );
end.

```

барлық  
символдар  
саны

```

>15<
>   15<

>1.234568E+001<
> 1.23E+001<
>  12.35<

```

барлық  
символдар  
саны

бөлшек бөлігіндегі  
символдар саны

# Толық шығарылуы

```
program qq;  
var a, b, c: integer;  
begin  
    writeln('Екі бҮтін сан енгізіңіз');  
    read ( a, b );  
    c := a + b;  
    writeln ( a, '+', b, '=', c );  
end.
```

бұны компьютер шығарады

**Протокол:**

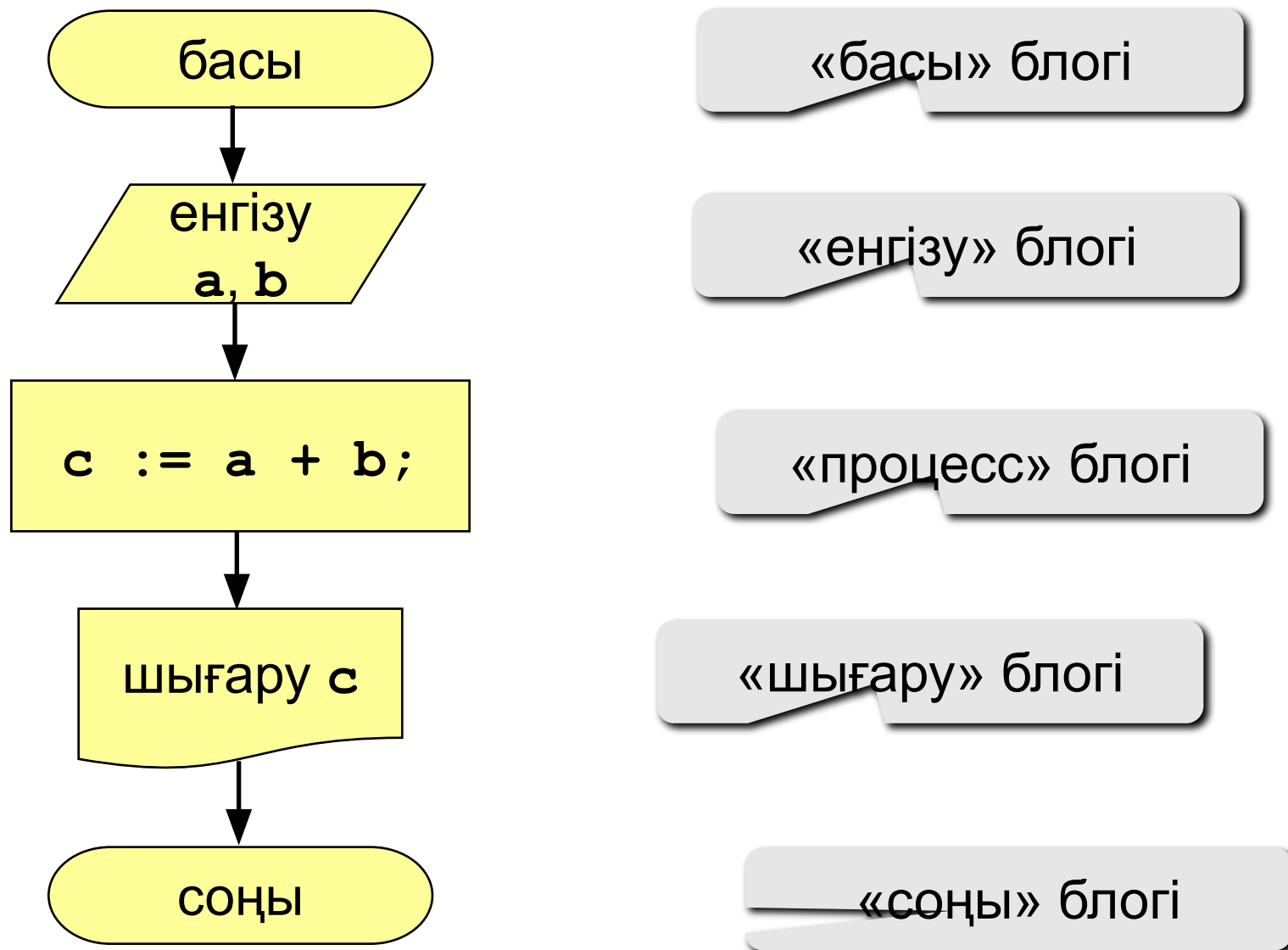
**Екі бҮтін сан енгізіңіз**

**25 30**

**25+30=55**

бұны программа жазушы өзі енгізеді

# Сызықтық алгоритмның блок-схемасы



# Тапсырмалар

---

**"4":** Үш сан енгізіп, олардың қосындысын және көбейтіндісін табу.

**Мысал:**

Үш сан енгіз:

4 5 7

$$4+5+7=16$$

$$4*5*7=140$$

**"5":** Үш сан енгізіп, олардың қосындысын, көбейтіндісін және арифметикалық ортасын табу.

**Мысал:**

Үш сан енгіз:

4 5 7

$$4+5+7=16$$

$$4*5*7=140$$

$$(4+5+7) / 3 = 5.33$$

# Паскаль тілінде программалау

## Тақырып 2. Тармақталу

# Тармақталу алгоритмдері

---

**Есеп.** Екі бүтін сан енгізіп, экранға олардың үлкенін шығару.

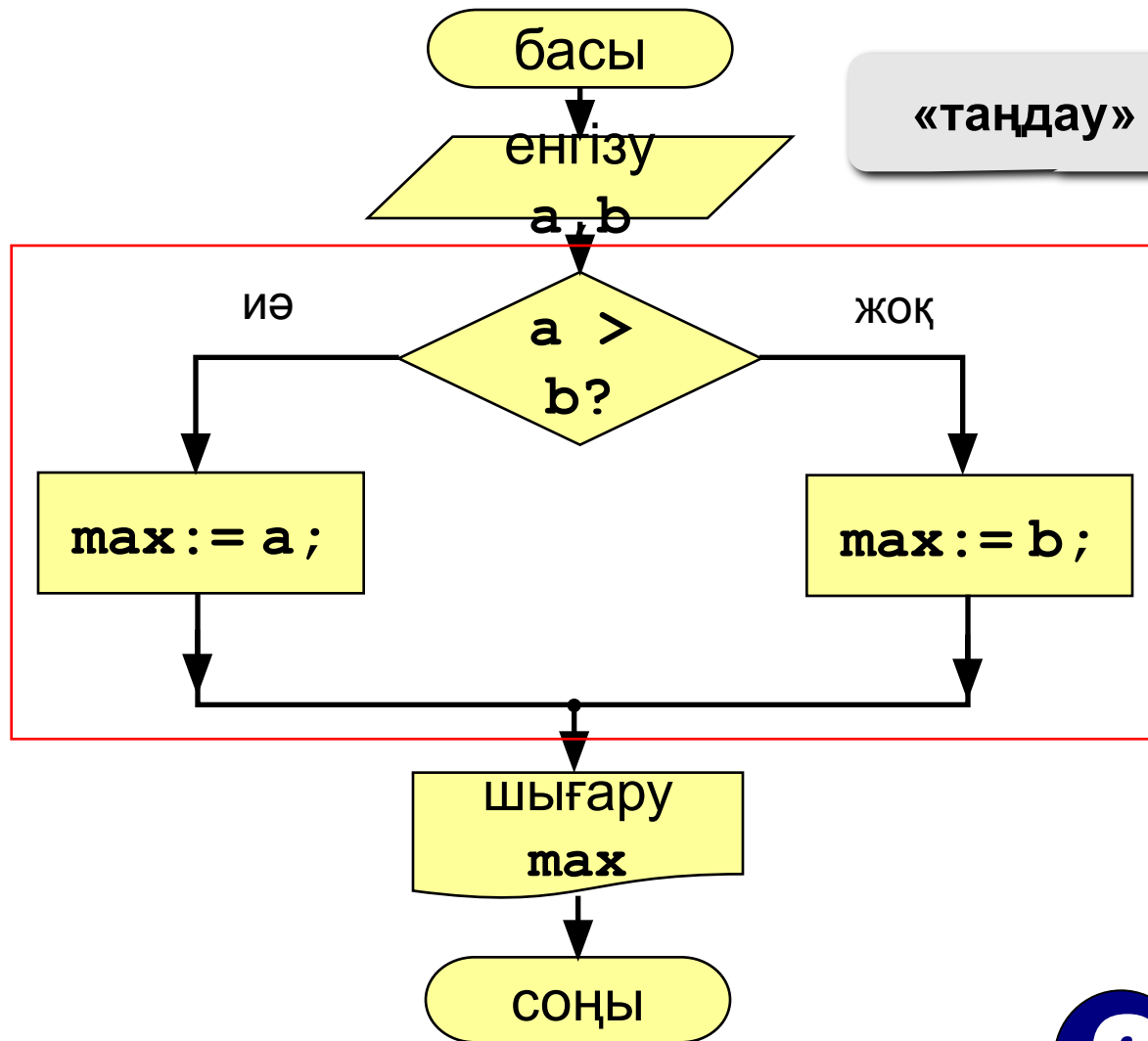
**Шығару амалы:** бірінші сан екінші саннан үлкен болса, біріншісін, ал егер кіші болса екіншісін экранға шығару.

**Ерекшелігі:** атқарушының әрекеті кейбір шарттардың орындалуына байланысты болады (**егер ...әйтпесе ...**).

Қадамдардың тізбектеп орындалуы кейбір шарттарға байланысты болатын алгоритмдерді **тармақталу** алгоритмдері деп атайды.



# Нұсқа 1. Блок-схема



«таңдау» блогі

Тармақталудың  
толымды түрі



Егер  $a = b$ ?

# Нұсқа 1. Программа

```
program qq;  
var a, b, max: integer;  
begin  
    writeln('Екі бҮтін сан енгізі');  
    read ( a, b );  
    if a > b then begin  
        max := a;  
    end  
    else begin  
        max := b;  
    end;  
    writeln ('Екі санның Үлкені', max);  
end.
```

шартты  
оператордың  
толымды түрі

# Шартты оператор

---

```
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

# Қай жері дұрыс емес?

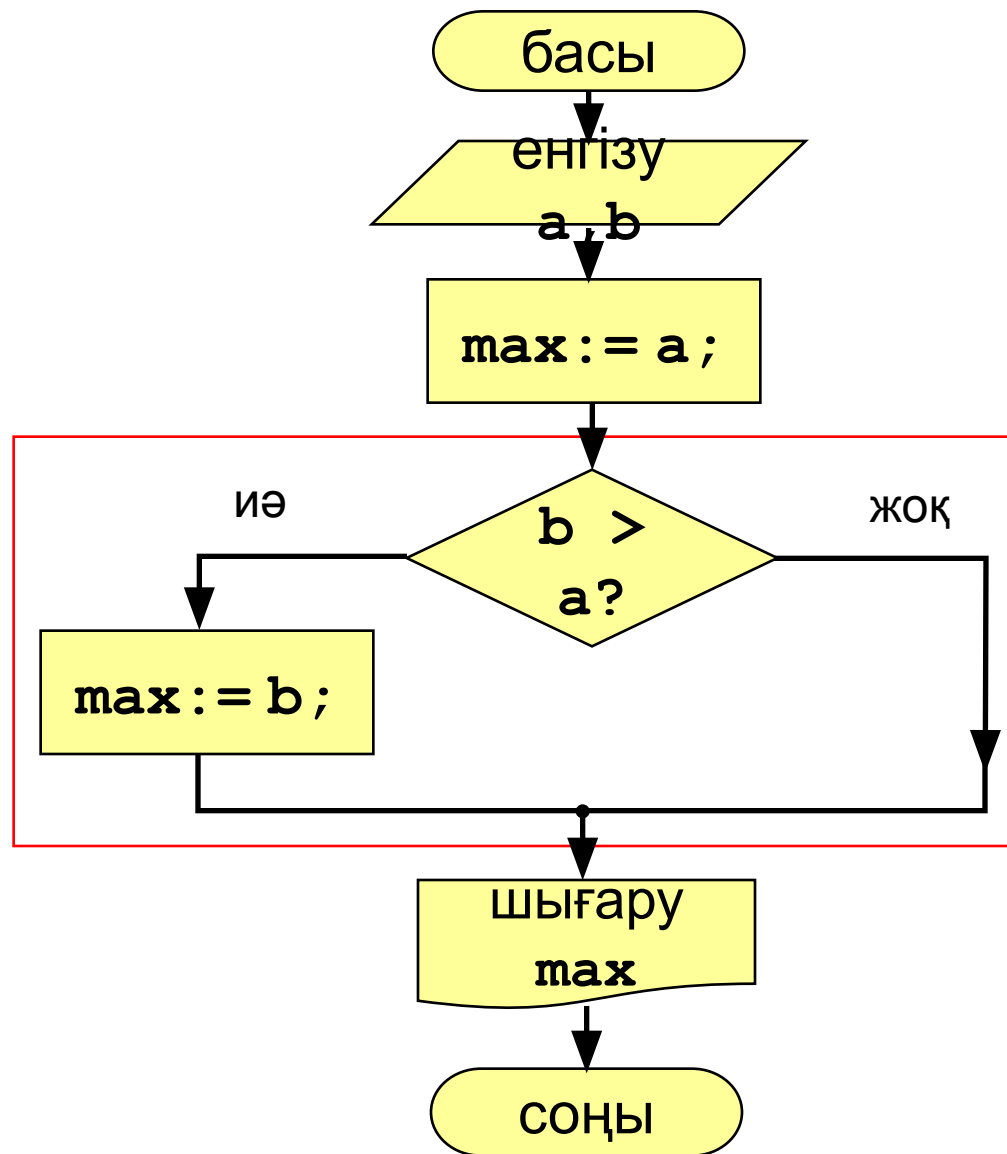
```
if a > b then begin
  a := b;
end
else begin
  b := a;
end;
```

```
if a > b then begin
  a := b; end
else begin
  b := a;
end;
```

```
if a > b then begin
  a := b;
end
else begin
  b := a;
end;
```

```
if a > b then begin
  a := b;
end
else begin
  b := a;
end;
```

## Нұсқа 2. Блок-схема



тармақталудың  
толымсыз түрі

## Нұсқа 2. Программа

```
program qq;  
var a, b, max: integer;  
begin  
    writeln('Екі бҮтін сан енгіз');  
    read ( a, b );  
    max := a;  
    if b > a then  
        max := b;  
    writeln ('Екі санның Үлкені', max);  
end.
```

шартты  
оператордың  
толымсыз түрі

## Нұсқа 2Б. Программа

---

```
program qq;  
var a, b, max: integer;  
begin  
    writeln('Екі бұтін сан енгіз');  
    read ( a, b );  
    max := b;  
    if a > b then  
        max := a;  
    writeln ('Екі санның Үлкені', max);  
end.
```

# Қай жері дұрыс емес?

---

```
if a > b then  
    a := b  
else b := a;
```

```
if a > b then begin  
    a := b;  
end  
else b := a;
```

```
if a > b then  
    a := b  
else b := a;
```

```
if b >= a then  
    b := a;
```



# Тапсырмалар

---

**"4":** Үш сан енгізіп олардың үлкенін табу.

**Мысал:**

Үш сан енгіз:

4      15      9

Ең Үлкен сан 15

**"5":** Бес сан енгізіп олардың үлкенін табу.

**Мысал:**

Бес сан енгіз:

4      15      9      56      4

Ең Үлкен сан 56

# Паскаль тілінде программалау

## Тақырып 3. Күрделі шарттар

# Күрделі шарттар

---

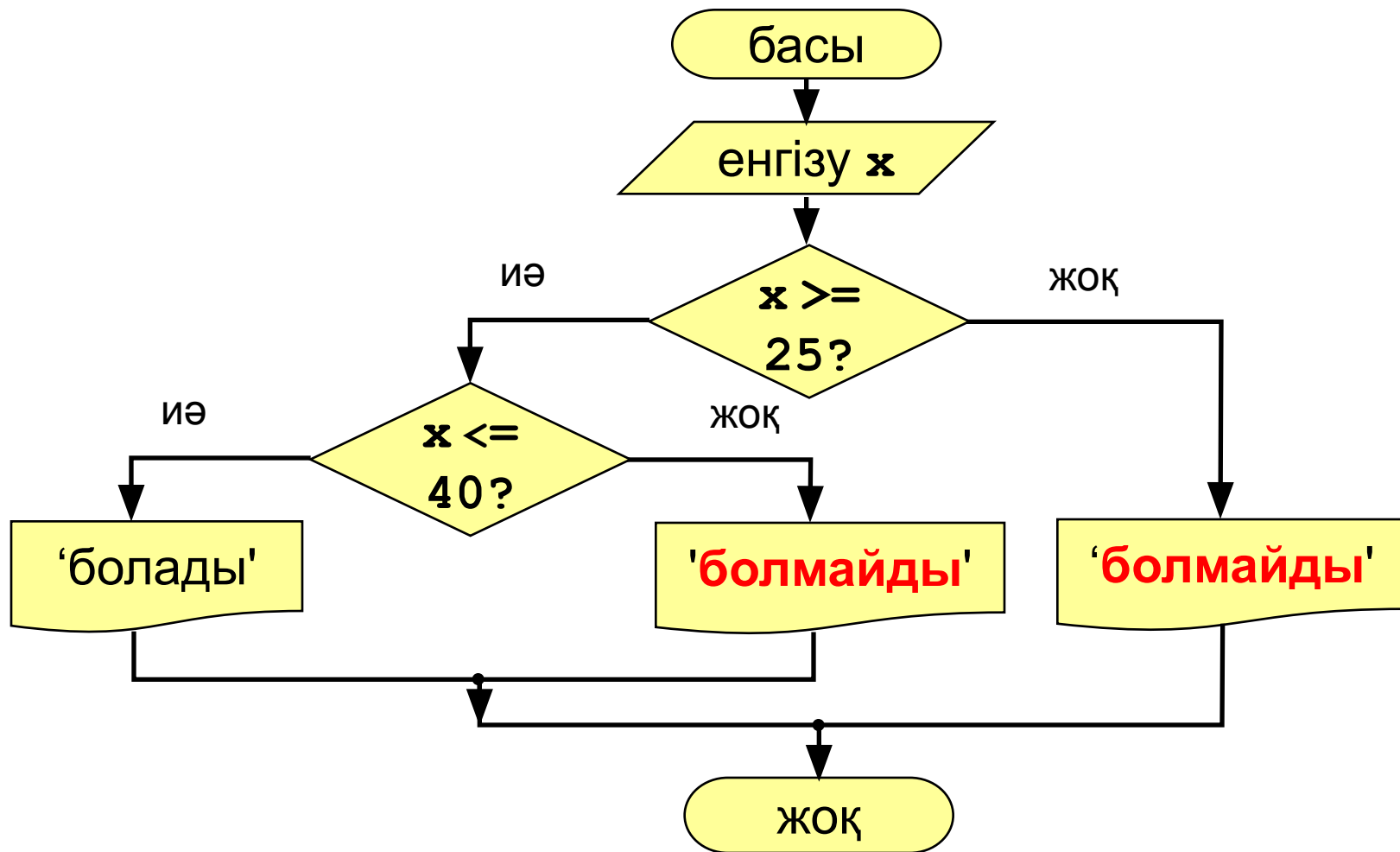
**Есеп.** Фирма 25 пен 40 жас арасындағы қызметкерлерді жұмысқа алуда. Адамның жасын енгізіп, оны фирмаға алуға болатындығы немесе болмайтындығы туралы мәлімет шығару («болады» немесе «болмайды» деген жауап шығару).

**Ерекшелігі:** екі шарттың бір мезгілде орындалатынын немесе орындалмайтын тексеру.



**Белгілі әдістермен шешуге бола ма?**

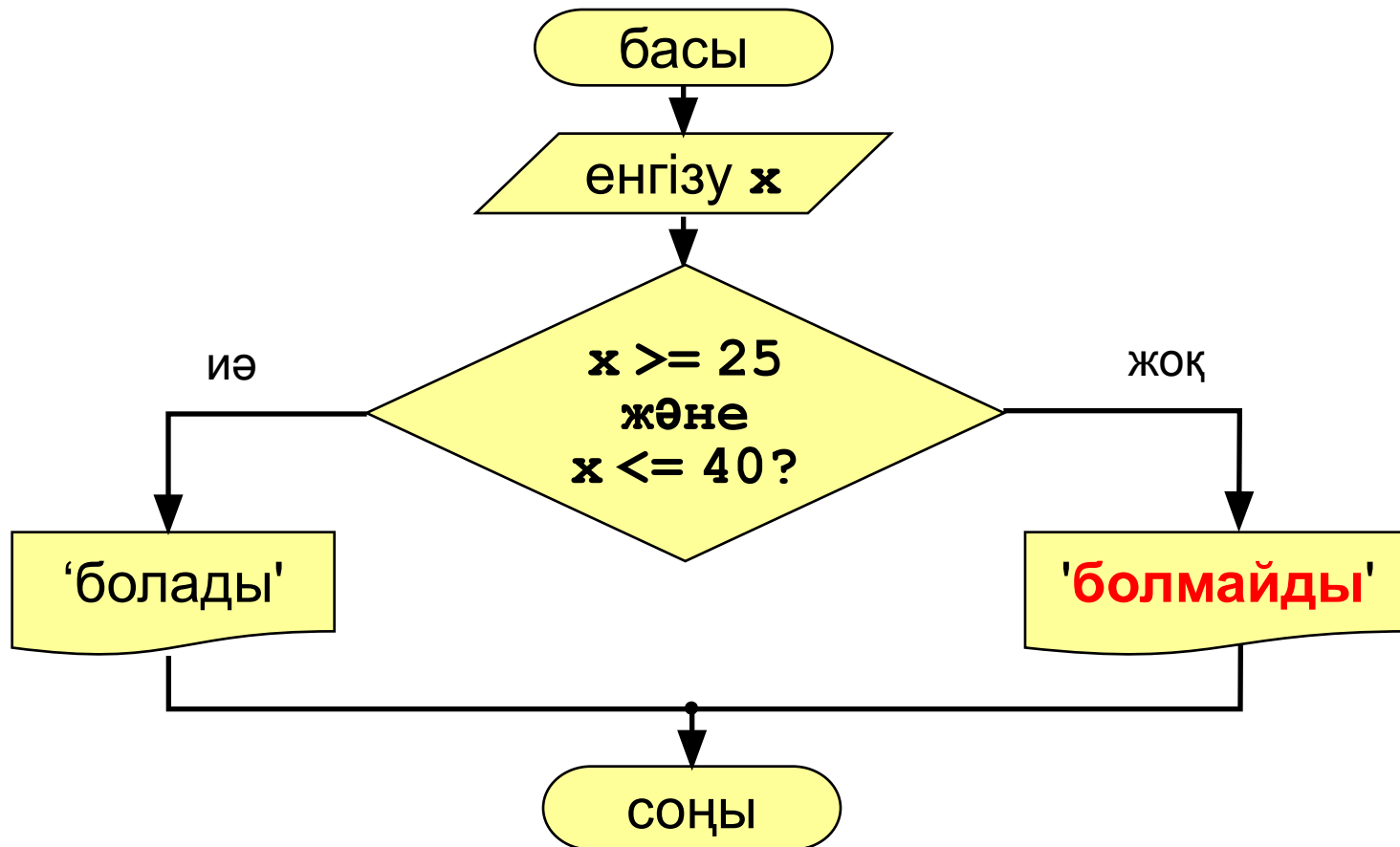
# Нұсқа 1. Алгоритм



# Нұсқа 1. Программа

```
program qq;  
var x: integer;  
begin  
    writeln('Жасты енгіз');  
    read ( x );  
    if x >= 25 then  
        if x <= 40 then  
            writeln ('Болады')  
        else writeln ('Болмайды')  
    else  
        writeln ('Болмайды') ;  
end.
```

## Нұсқа 2. Алгоритм



## Вариант 2. Программа

```
program qq;  
var x: integer;  
begin  
    writeln('Жасты енгіз');  
    read ( x );  
    if (x >= 25) and (x <= 40) then  
        writeln ('Болады')  
    else writeln ('Болмайды')  
end.
```

күрделі  
шарт

# 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)**

&lt;

&lt;=

&gt;

&gt;=

тең

=

тең емес

&lt;&gt;



# Күрделі шарттар

---

## 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
```

# Күрделі шарттар

$a := 2; b := 3; c := 4;$  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

х-тің қандай мәндерінде шарт ақиқат болады:

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

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

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

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

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

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

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

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

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

# Тапсырмалар

---

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

**Мысал:**

Ай нөмірін енгіз:

**4**

көктем

# Паскаль тілінде программалау

## Тақырып 4. Циклдер

# Циклдер

---

**Цикл** – белгілі бір әрекеттер тізбегінің бірнеше рет қайталанып орындалуы.

- қадамдар саны **белгілі** цикл
- қадамдар саны **белгісіз** цикл (шартты цикл)

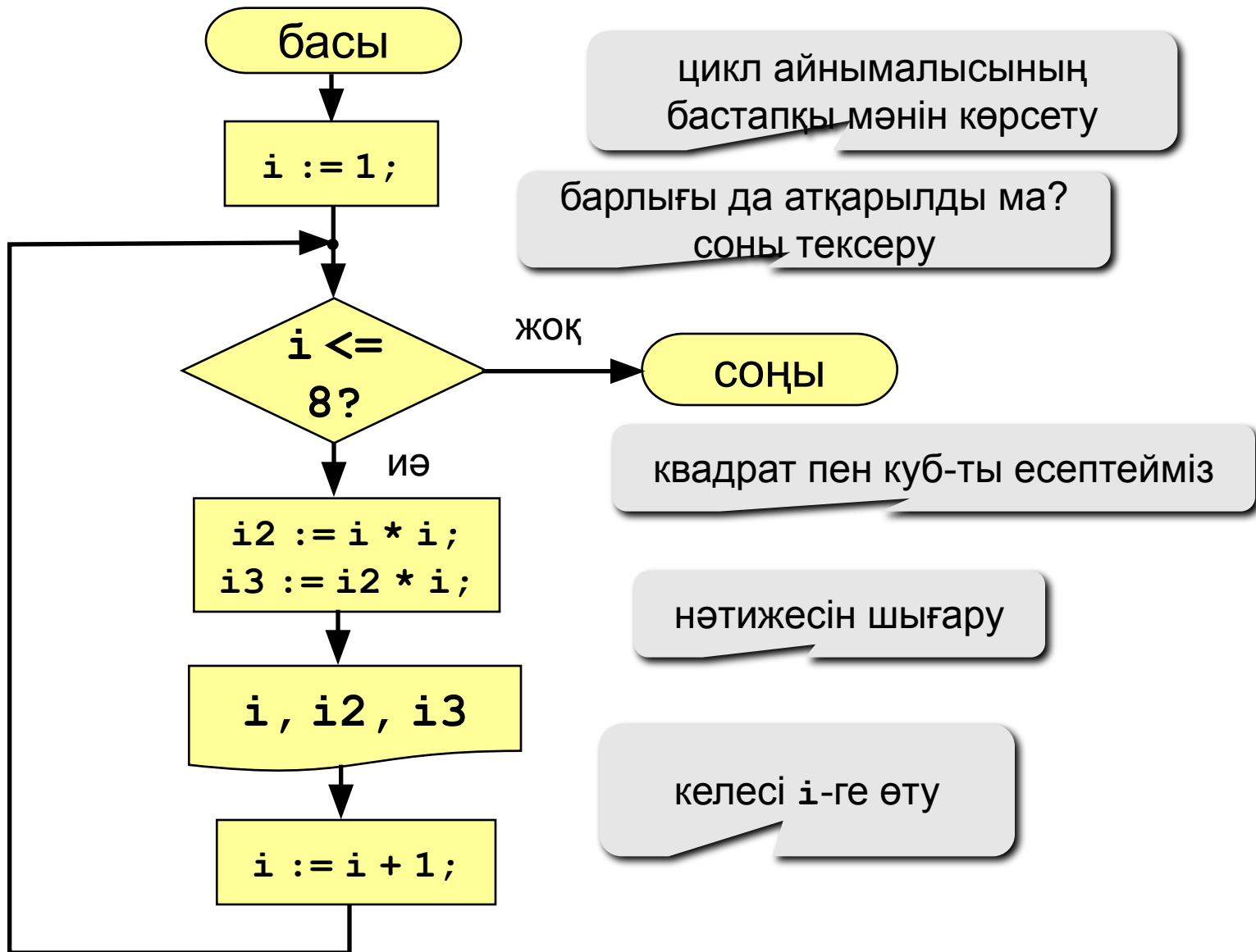
**Есеп.** 1-ден 8-ге дейінгі бүтін сандардың квадраттары мен кубтарын есептеп шығару (**a**-дан **b**-ға дейін).

**Ерекшелігі:** бірдей амалдар 8 рет орындалады.

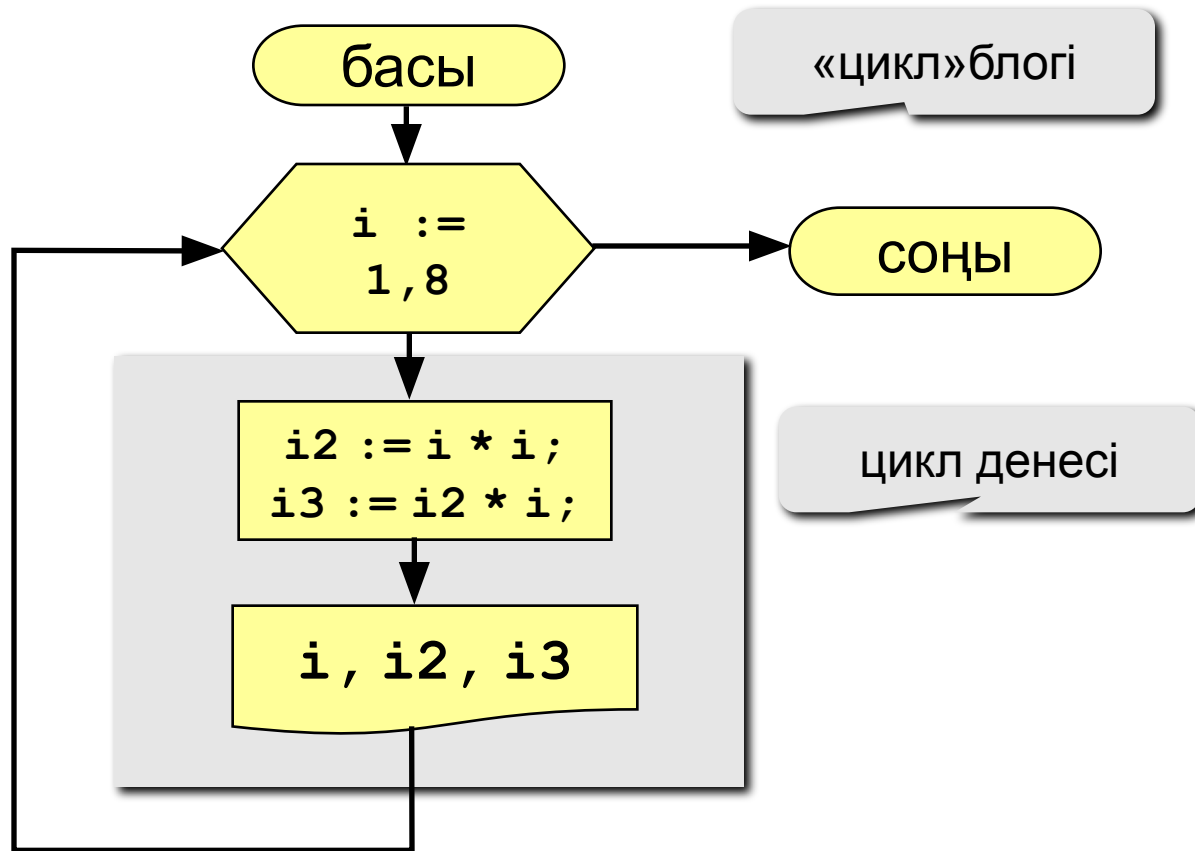


Белгілі әдістермен шешуге бола ма?

# Алгоритм



# Алгоритм ("цикл" блогімен)



# Программа

```
program qq;  
var i, i2, i3: integer;  
begin
```

бастапқы мән

цикл айнымалысы

ең соңғы мән

```
  for i:=1 to 8 do begin  
    i2 := i*i;  
    i3 := i2*i;  
    writeln(i:4, i2:4, i3:4);  
  end;
```

```
end.
```



# Айнымалысы кеміп отыратын цикл

---

**Есеп.** 8-ден 1-ге дейінгі бүтін сандардың квадраттары мен кубтарын есептеп шығару (кері қарай есептеу).

**Ерекшелігі:** цикл айнымалысы кеміп отыруы керек.

**Шығарылуы:**

```
for i:=8 downto 1 do begin
    i2 := i*i;
    i3 := i2*i;
    writeln(i:4, i2:4, i3:4);
end;
```

# Айнымалысы бар цикл

---

## Айнымалының 1-ге артуы:

```
for <айнымалы> := <алғашқы мән> to  
    <ең соңғы мән> do begin  
    {цикл денесі}  
end;
```

## Айнымалының 1-ге кемуі:

```
for <айнымалы> := <алғашқы мән>  
    downto  
    <ең соңғы мән> do begin  
    {цикл денесі}  
end;
```

# Айнымалысы бар цикл

---

## Ерекшеліктері:

- цикл айнымалысының мәні бүтін болуы керек (**integer**)
- цикл айнымалысының өзгеру қадамы қашанда 1-ге (**to**) немесе -1-ге (**downto**) тең
- егер цикл денесінде бір ғана оператор болса, **begin** мен **end** сөздерін жазбай кетуге болады:

```
for i:=1 to 8 do  
    writeln( 'Привет' ) ;
```

- егер соңғы мән алғашқы мәннен кіші болса, (**to**) циклы бірде бір рет орындалмайды (шартты циклдың басында тексеру, шартты алдын ала тексеретін цикл)

# Айнымалысы бар цикл

---

## Ерекшеліктері:

- Цикл денесінде цикл айнымалысын өзгертуге рұқсат етілмейді (неліктен?)
- Алғашқы және соңғы мәндер өзгертілгенімен цикл ішіндегі қадамдар саны өзгермейді:

```
n := 8;  
for i:=1 to n do begin  
    writeln('Салем');  
    n := n + 1;  
end;
```

мүдіріп қалу жоқ

# Цикл неше рет орындалған?

---

```
a := 1;  
for i:=1 to 3 do a := a+1;
```

~~a = 4~~

```
a := 1;  
for i:=3 to 1 do a := a+1;
```

~~a = 1~~

```
a := 1;  
for i:=1 downto 3 do a := a+1;
```

~~a = 1~~

```
a := 1;  
for i:=3 downto 1 do a := a+1;
```

~~a = 4~~

# Қадамдарды қалай өзгертуге болады?

**Есеп.** Экранға 1-ден 9-ға дейінгі тақ сандардың квадраттары мен кубтарын есептеп шығару.

**Ерекшелігі:** цикл айнымалысы 2-ге артып отыру керек.

**Мәселе:** Паскальде қадамдар саны 1 немесе -1 болуы мүмкін.

**Шығарылуы:**

```
for i:=1 to 9 do begin
  if i mod 2 = 1 then begin
    i2 := i*i;
    i3 := i2*i;
    writeln(i:4, i2:4, i3:4);
  end;
end;
```

Тек қана тақ  
**i**-лермен  
орындалады

# Қадамдарды қалай өзгертуге болады? – II

---

**Идея:** Тек 5 санды ғана шығару керек, **k** айнымалысы 1-ден 5-ке дейін өзгереді. **i**-дің бастапқы мәні 1-ге тең, циклдың әр қадамы сайын **i** 2-ге артып отырады.

**Шығарылуы:**

```
i := 1;  
for k:=1 to 5 do begin  
    i2 := i*i;  
    i3 := i2*i;  
    writeln(i:4, i2:4, i3:4);  
    i := i + 2;  
end;
```

# Қадамдарды қалай өзгертуге болады? – III

**Идея:** Тек 5 санды ғана шығару керек, **k** айнымалысы 1-ден 5-ке дейін өзгереді. **k-ны біле отырып, i-ді есептеу қажет.**

k	1	2	3	4	5
i	1	3	5	7	9

$$i = 2k - 1$$

**Шығарылуы:**

```
for k:=1 to 5 do begin
  i := 2*k - 1;
  i2 := i*i;
  i3 := i2*i;
  writeln(i:4, i2:4, i3:4);
end;
```



# Тапсырмалар

---

**"4":** а мен Ъ-ны енгізіп, осы екеуінің арасындағы сандардың квадраттары мен кубтарын есептеп шығару.

**Мысал:**

Интервал шекараларын енгіз:

**4 6**

4    16    64

5    25    125

6    36    216

**"5":** 1, 2, 4, 7, 11, 16, ... тәртібімен орналасқан 10 санның квадраттары мен кубтарын есептеп шығару

**Мысал:**

1            1            1

2            4            8

4            16            64

...

46    2116    97336

# Паскаль тілінде программалау

## Тақырып 5. Шартты циклдер

# Қадамдар саны белгісіз цикл

---

**Мысал:** Бөренеден ағаштың бұтағын кесіп алу керек.  
Қолараны әрі-бері неше рет қозғалтуымыз керек, . ?

**Есеп:** (<2000000) бүтін сан енгізіп, ондағы цифрлардың санын анықтау.

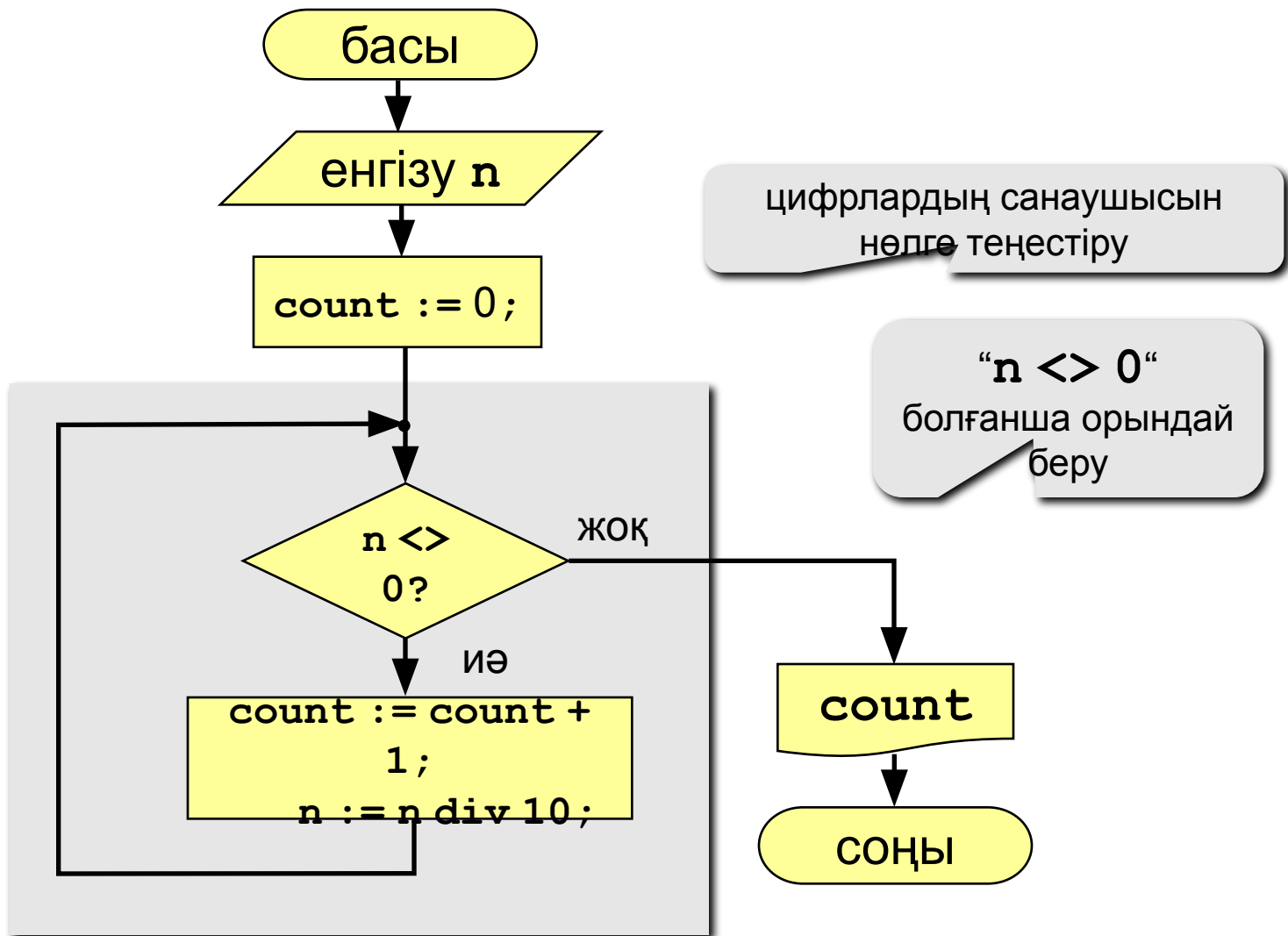
**Шығару амалы:** Ең соңғы орналасқан цифрларды кезекпен алып тастай отырып, санаушыны арттырамыз.

n	count
123	0
12	1
1	2
0	3

**Мәселе:** неше қадам жасау керектігі белгісіз.

**Шығарылуы:**  $n = 0$  болғанда тоқтау керек, яғни " $n \neq 0$ " болғанша орындай беру керек.

# Алгоритм



# Программа

```
program qq;  
var n, count, n1: integer;  
begin  
    writeln('Бүтін сан енгіз');  
    read(n); n1 := n;  
    count := 0;  
  
    while n <> 0 do begin  
        count := count + 1;  
        n := n div 10;  
    end;  
    writeln(n1, ' санында', count, ' цифр табылды');  
end.
```

"n <> 0"  
болғанша  
орындай беру

# Шартты цикл

---

```
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;
```

# Шартты цикл

## 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;
```

# Цикл неше рет орындалған?

```
a := 4; b := 6;
while a < b do a := a + 1;
```

2 рет  
a = 6

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

1 рет  
a = 10

```
a := 4; b := 6;
while a > b do a := a + 1;
```

0 рет  
a = 4

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

1 рет      b  
= -2

```
a := 4; b := 6;
while a < b do a := a - 1;
```

дағдарысқа  
ұшырау



# For-ны while-ға ауыстыру және керісінше

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

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

```
for i:=a downto b do  
begin  
    {цикл денесі}  
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.**

# Тапсырмалар

---

**"4":** Бүтін сан енгізіп, ондағы цифрлардың қосындысын табу.

**Мысал:**

Бүтін сан енгіз:

**1234**

1234 санындағы цифрлардың қосындысы 10-ға тең.

**"5":** Бүтін сан енгізіп, оның жазылуында бірдей цифрлардың бар-жоғын анықтау.

**Мысал:**

Бүтін сан енгіз:

**1234**

Жоқ.

Бүтін сан енгіз:

**1224**

Бар.

# Тізбектер

## Мысалдар:

• 1, 2, 3, 4, 5, ...

$$a_n = n$$

$$a_1 = 1, \quad a_{n+1} = a_n + 1$$

• 1, 2, 4, 7, 11, 16, ...

$$a_1 = 1, \quad a_{n+1} = a_n + n$$

• 1, 2, 4, 8, 16, 32, ...

$$a_n = 2^{n-1}$$

$$a_1 = 1, \quad a_{n+1} = 2a_n$$

•  $\frac{1}{2}, \frac{1}{2}, \frac{3}{8}, \frac{1}{4}, \frac{5}{32}, \dots$

$\frac{1}{2}, \frac{2}{4}, \frac{3}{8}, \frac{4}{16}, \frac{5}{32}, \dots$

$$a_n = \frac{b_n}{c_n}$$

$$b_1 = 1, \quad b_{n+1} = b_n + 1$$

$$c_1 = 2, \quad c_{n+1} = 2c_n$$

# Тізбектер

**Есеп:** модулі бойынша 0,001-ден асып түсетін

$$1, -\frac{1}{2}, \frac{2}{4}, -\frac{3}{8}, \frac{4}{16}, -\frac{5}{32}, \dots$$

тізбектегі барлық элементтердің қосындысын табу:

$$S = 1 - \frac{1}{2} + \frac{2}{4} - \frac{3}{8} + \frac{4}{16} - \frac{5}{32} + \dots$$

**Тізбек элементтері (№2-ден бастап ):**

$$a = z \frac{b}{c}$$

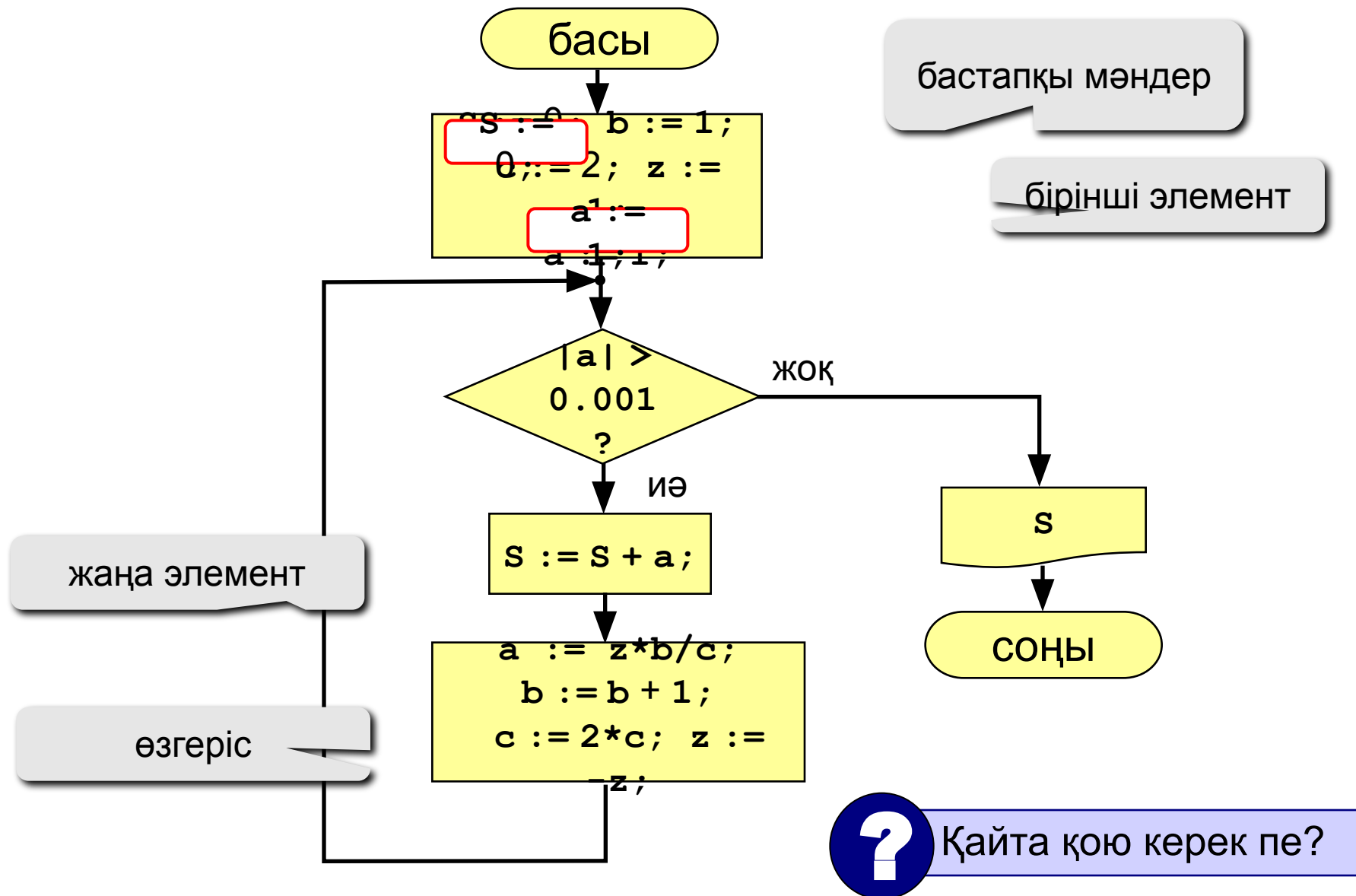
n	1	2	3	4	5	...
b	1	2	3	4	5	...
c	2	4	8	16	32	...
z	-1	1	-1	1	-1	...

`b := b+1;`

`c := 2*c;`

`z := -z;`

# Алгоритм



# Программа

```
program qq;  
var b, c, z: integer;  
    S, a: real;  
begin  
    S := 0; z := -1;  
    b := 1; c := 2; a := 1;  
    while abs(a) > 0.001 do begin  
        S := S + a;  
        a := z * b / c;  
        z := - z;  
        b := b + 1;  
        c := c * 2;  
    end;  
    writeln('S =', S:10:3);  
end.
```

бастапқы  
мәндер

қосындының  
артуы

тізбек элементін есепке  
алу

келесі  
қосылғышқа өту

# Тапсырмалар

---

**"4":** Тізбектегі элементтердің қосындысын 0,001 дәлдікпен табу :

$$S = 1 + \frac{2}{3 \cdot 3} - \frac{4}{5 \cdot 9} + \frac{6}{7 \cdot 27} - \frac{8}{9 \cdot 81} + \dots$$

**Жауап:**

$$S = 1.157$$

**"5":** Тізбектегі элементтердің қосындысын 0,001 дәлдікпен табу :

$$S = 1 + \frac{2}{2 \cdot 3} - \frac{4}{3 \cdot 9} + \frac{6}{5 \cdot 27} - \frac{8}{8 \cdot 81} + \frac{10}{13 \cdot 243} - \dots$$

**Жауап:**

$$S = 1.220$$

# Соңғышартты цикл

---

**Есеп:** Бүтін **оң** сан енгізіп(<2000000), ондағы цифрлардың санын анықтау.

**Мәселе:** Теріс санның немесе нөлдің енгізілмеуін қалай қадағалауға болады?

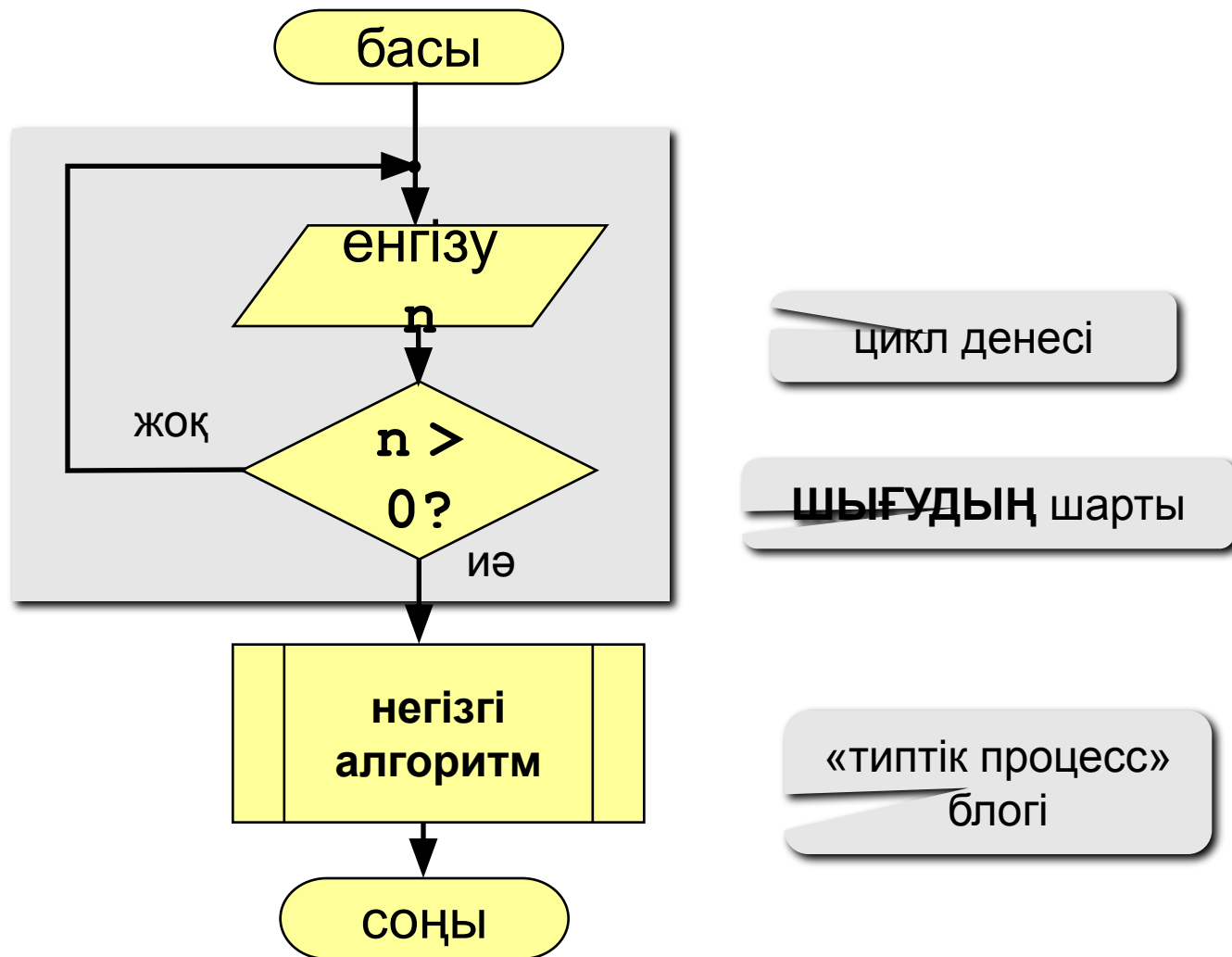
**Шығарылуы:** Егер бұрыс сан енгізілсе, берілгендерді енгізу үшін кері оралу (цикл!).

**Ерекшелігі:** Цикл денесі кемінде бір рет орындалады.  
=> шартын тексеруді циклдың соңында орындау керек.  
(**соңғышартты цикл**).

**Соңғышартты цикл** – шарттың тексерілуі циклдың соңында орындалатын цикл түрін айтады.



# Соңғышартты цикл: алгоритм



# Программа

```
program qq;  
var n: integer;  
begin
```

```
  repeat  
    writeln('Оң сан енгіз');  
    read(n);  
  until n > 0;
```

ШЫҒУДЫҢ шарты

```
  ... { негізгі алгоритм }
```

```
end.
```

## Ерекшеліктері:

- цикл денесі кемінде бір рет орындалады
- **until** ("...болмайынша") сөзінен кейін циклдан **ШЫҒУДЫҢ** шарты қойылады

# Цикл неше рет орындалған?

```
a := 4; b := 6;  
repeat a := a + 1; until a > b;
```

3 рет  
a = 7

```
a := 4; b := 6;  
repeat a := a + b; until a > b;
```

1 рет  
a = 10

```
a := 4; b := 6;  
repeat a := a + b; until a < b;
```

**Мүдіріп Қалу**

```
a := 4; b := 6;  
repeat b := a - b; until a < b;
```

2 рет  
b = 6

```
a := 4; b := 6;  
repeat a := a + 2; until a < b;
```

**Мүдіріп Қалу**

# Тапсырмалар (бұрыс енгізуден қорғау)

**"4":** Натурал сан енгізіп, сол сандағы цифрлардың қосындысы 10-ға тең екендігінің дұрыстығын анықтау.

**Мысал:**

`>= 0 сан енгіз:`

**-234**

Оң сан енгізу керек.

`>= 0 сан енгіз:`

**1234**

Иә

`>= 0 сан енгіз:`

**1233**

Жоқ

**"5":** Натурал сан енгізіп, қандай цифрлардың бірнеше рет кездесетіндігін анықтау.

**Мысал:**

`>= 0 сан енгіз:`

**2323**

2, 3 Қайталанады

`>= 0 сан енгіз:`

**1234**

Қайталану жоқ.

# Паскаль тілінде программалау

## Тақырып 6. Таңдау операторы

# Таңдау операторы

---

**Есеп:** Ай нөмірін енгізіп, осы айдағы күндердің санын шығару.

**Шығарылуы:** Айдағы күндер саны

**28 күн** – 2 (ақпан)

**30 күн** – 4 (сәуір), 6 (маусым), 9 (қыркүйек), 11 (қараша)

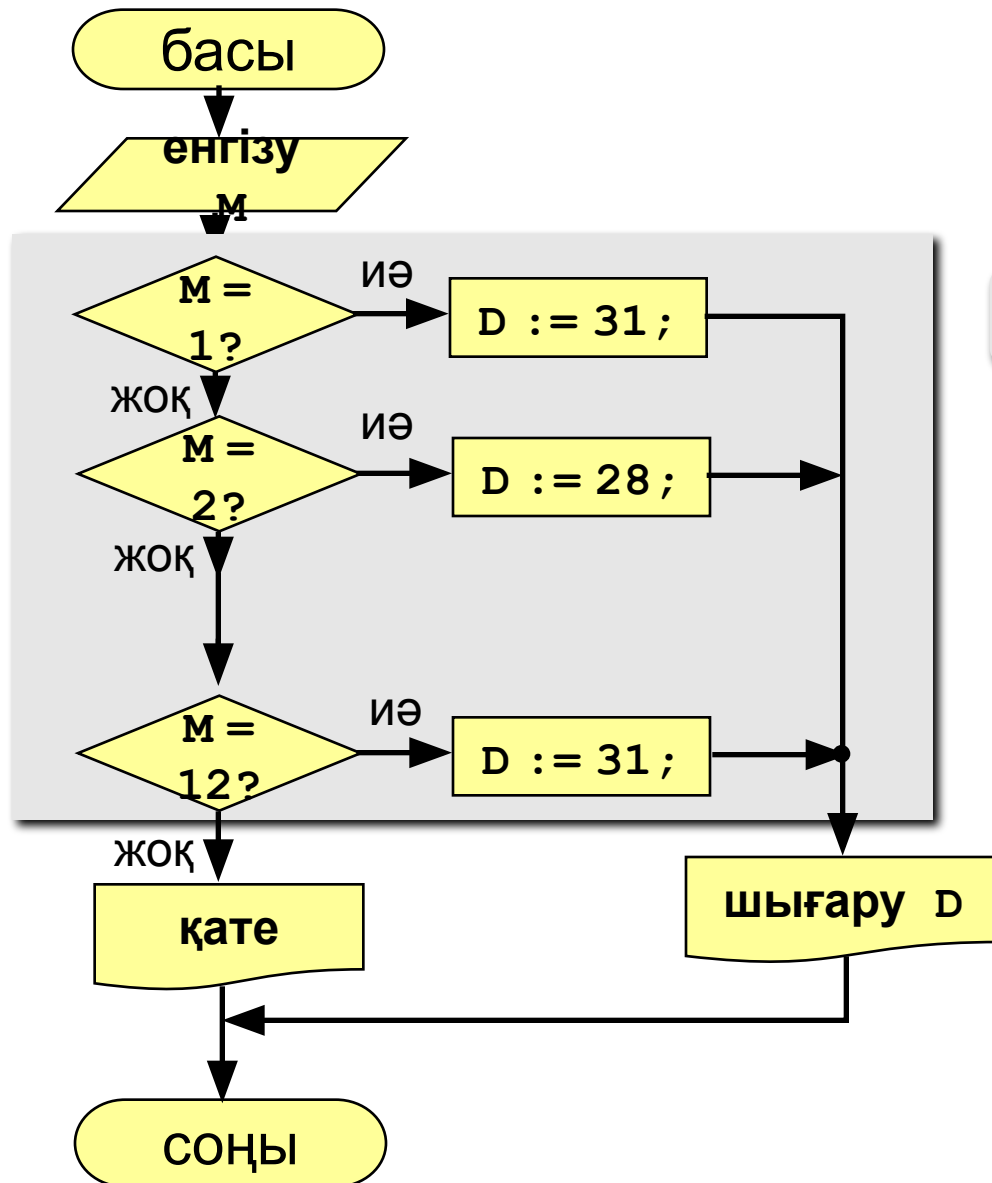
**31 күн** – 1 (қаңтар), 3 (наурыз), 5 (мамыр), 7 (шілде),  
8 (тамыз), 10 (қазан), 12 (желтоқсан)

**Ерекшелігі:** Екі нұсқаның емес, айдың нөміріне сәйкес бірнеше нұсқаның бірін таңдау.



Бұған дейінгі қолданған әдістермен шешуге бола ма?

# Алгоритм



таңдау операторы

ешбір  
нұсқасы  
сәйкес  
келген жоқ

# Программа

```
program qq;  
var M, D: integer;  
begin  
    writeln('Айдың нөмірін енгіз:');  
    read ( M );  
  
    case M of  
        2:          begin D := 28; end;  
        4,6,9,11:   begin D := 30; end;  
        1,3,5,7,8,10,12: D := 31;  
        else          D := -1;  
    end;  
  
    if D > 0 then  
        writeln('Бұл айда ', D, ' күн.')    else  
        writeln('Айдың нөмірі дұрыс емес');  
end.
```

ешбір нұсқасы  
сәйкес келген жоқ



# 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;
```

болуы мүмкін

# Таңдау операторы

---

## 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;
```

# Таңдау операторы

## 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;
```

# Қай жері дұрыс емес?

```
case a of
  2: begin a := b;
  4: a := c;
end;
```

```
case a of
  2: a := b;
  4: a := c;
end;
```

```
case a of
  2..5: a := b;
  4: a := c;
end;
```

```
case a of
  0..2: a := b;
  3..6: a := c;
end;
```

```
case a+c/2 of
  2: a := b;
  4: a := c;
end;
```

```
begin
case a of
  2: a := b; d := 0; end;
  4: a := c;
end;
```

# Тапсырмалар (бұрыс енгізуден 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**

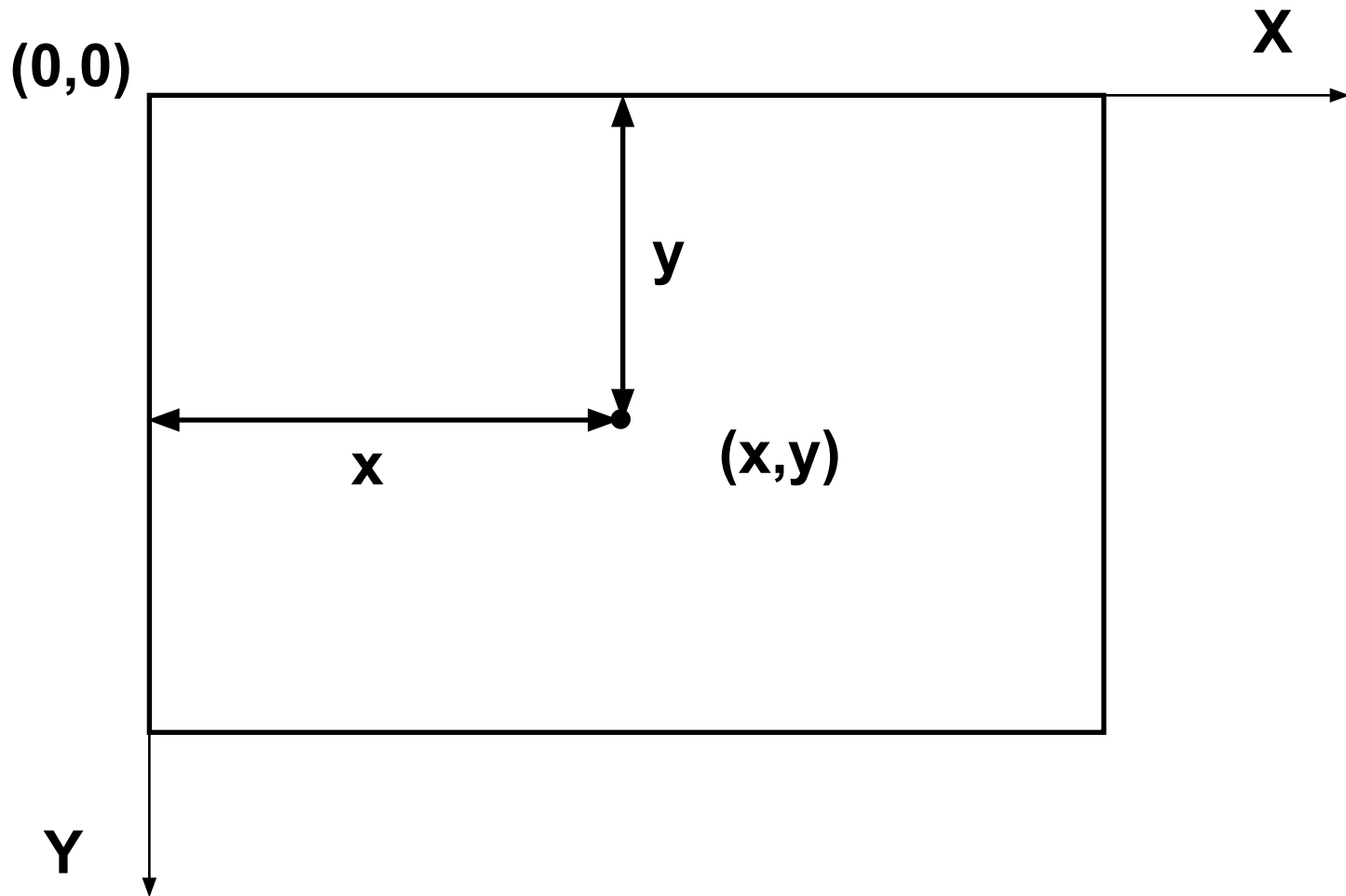
Жаңа жылға дейін 6 күн бар.

# Паскаль тілінде программалау

## Тақырып 7. Графика

# Координаталар жүйесі

---



# 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

`TextColor ( 0, 0, 255 );`

R

G

B



# Нүктелер, кесінділер және қисық сызықтар

$(x, y)$



```
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);
```

$(x_1, y_1)$



$(x_2, y_2)$



$(x_5, y_5)$



$(x_3, y_3)$



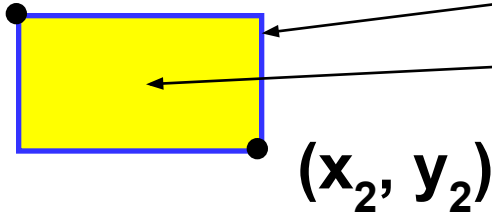
$(x_4, y_4)$



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

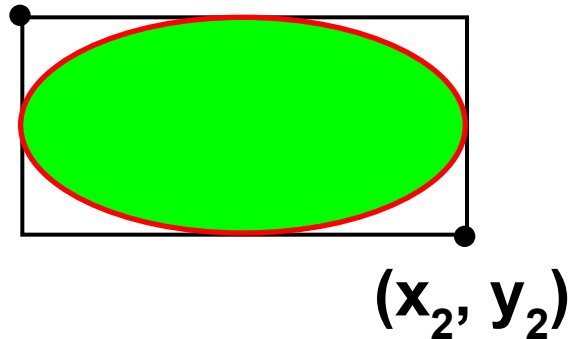
# Фигуралардың ішін бояу

$(x_1, y_1)$

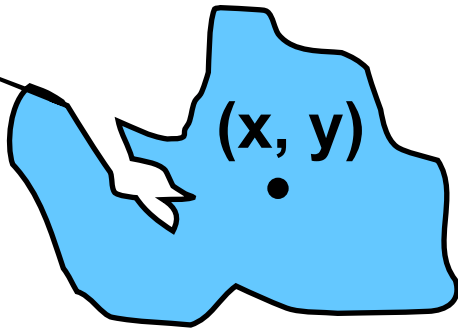


```
Pen (1, 0, 0, 255);  
Brush (1, 255, 255, 0);  
Rectangle (x1, y1, x2, y2);
```

$(x_1, y_1)$

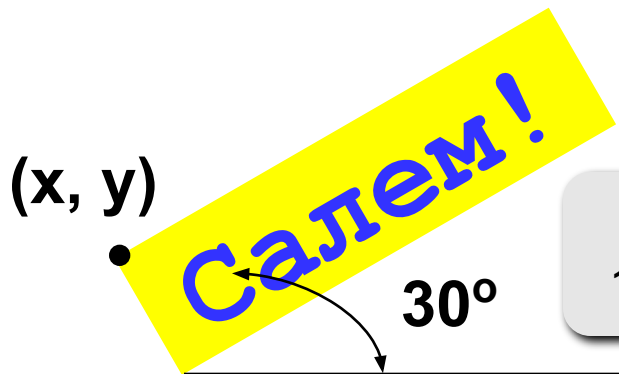


```
Pen (1, 255, 0, 0);  
Brush (1, 0, 255, 0);  
Ellipse (x1, y1, x2, y2);
```



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

# Мәтін



```
TextColor (0, 0, 255);  
Brush (1, 255, 255, 0);  
Font (20, 30, 600);
```

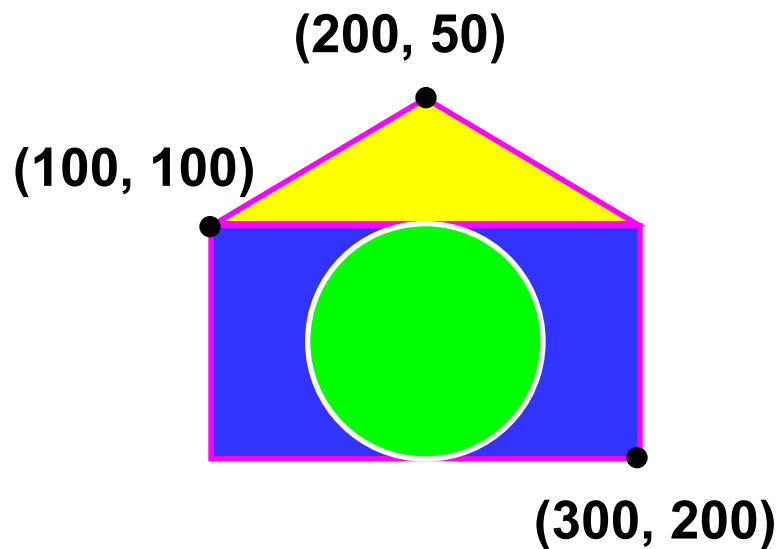
өлшемі  
10 пиксель

айналу  
бұрышы

қанықтылық:  
400 – қалыпты  
600 – қалың

```
MoveTo (x, y);  
writeln ('Сәлем!');
```

# Мысал

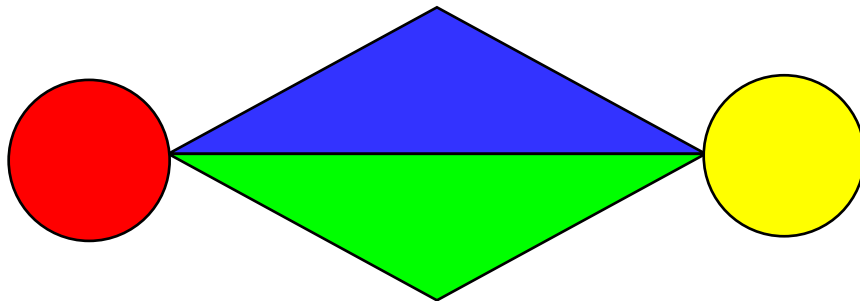


```
program qq;  
begin  
    Pen(2, 255, 0, 255);  
    Brush(1, 0, 0, 255);  
    Rectangle(100, 100, 300, 200);  
    MoveTo(100, 100);  
    LineTo(200, 50);  
    LineTo(300, 100);  
    Brush(1, 255, 255, 0);  
    Fill(200, 75);  
    Pen(2, 255, 255, 255);  
    Brush(1, 0, 255, 0);  
    Ellipse(150, 100, 250, 200);  
end.
```

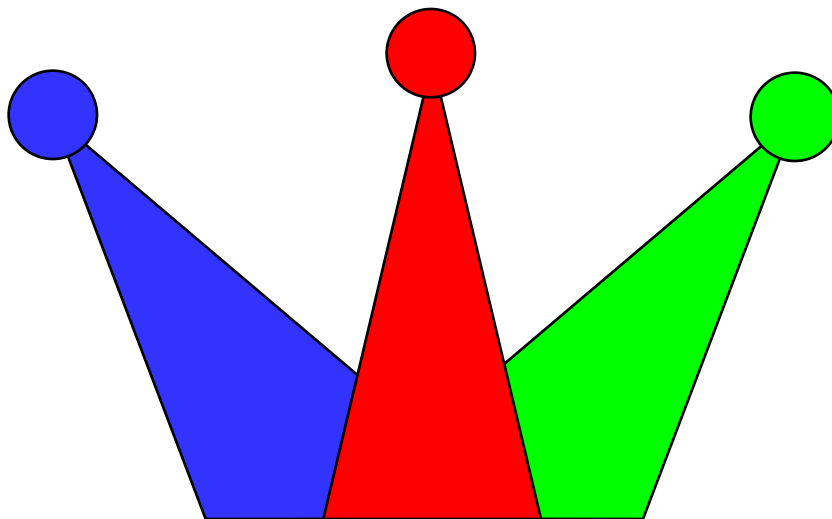
# Тапсырмалар

---

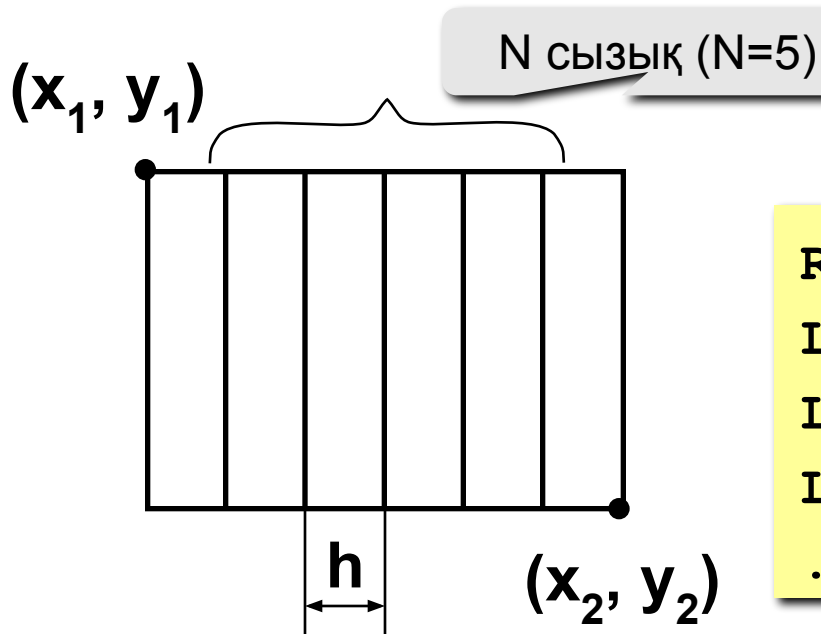
"4": «Бақа»



"5": «Тәж»



# Штрихтер салу



$$h = \frac{x_2 - x_1}{N + 1}$$

```
Rectangle (x1, y1, x2, y2);
Line ( x1+h, y1, x1+h, y2);
Line ( x1+2*h, y1, x1+2*h, y2);
Line ( x1+3*h, y1, x1+3*h, y2);
...
```

x

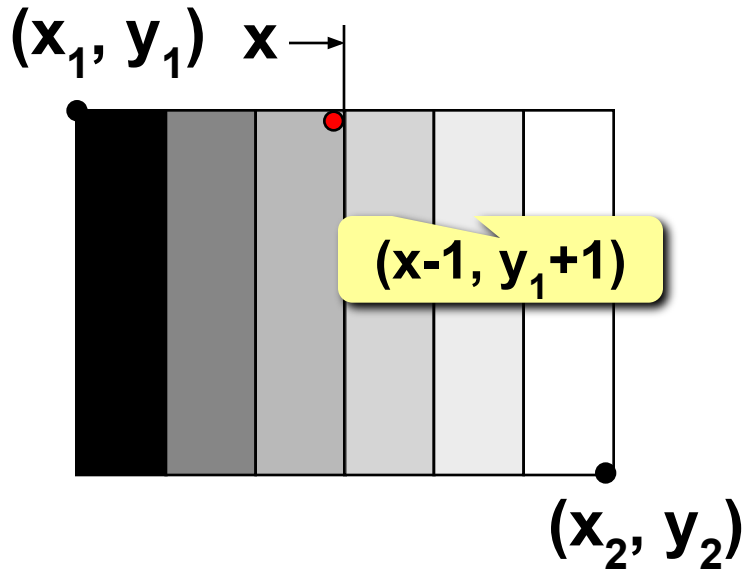
x

```
h := (x2 - x1) / (N + 1);
Rectangle (x1, y1, x2, y2);
x := x1 + h;
for i:=1 to N do begin
    Line( round(x), y1, round(x), y2);
    x := x + h;
end;
```

var x, h: real;

жақын орналасқан бүтін санға дейін дөңгелектеу

# Түсті қалай өзгертуге болады?



сұр: R = G = B

```
Brush ( 1, c, c, c );
Fill ( ???, ??? );
```

Өзгеру қадамы c:

$$h_c = \frac{255}{N+1}$$

```
hc := 255 div (N + 1);
c := 0;
```

```
for i:=1 to N+1 do begin
```

```
  Line (round(x), y1, round(x), y2);
```

```
  Brush (1, c, c, c);
```

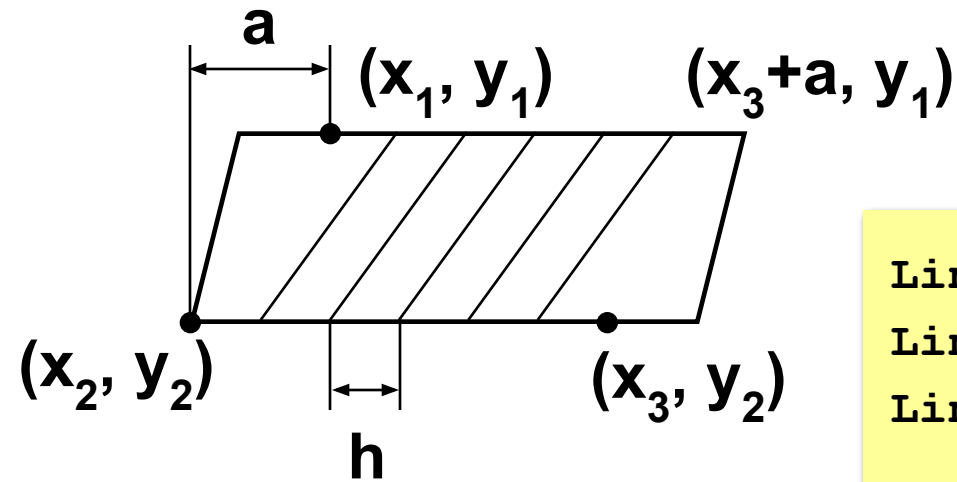
```
  Fill (round(x)-1, y1+1);
```

```
  x := x + h; c := c + hc;
```

```
end;
```

```
var c, hc: integer;
```

# Штрихтер салу



$$a = x_1 - x_2$$

$$h = \frac{x_3 - x_2}{N + 1}$$

```
Line( x1+h, y1, x1+h-a, y2);
Line( x1+2*h, y1, x1+2*h-a, y2);
Line( x1+3*h, y1, x1+3*h-a, y2);
...
```

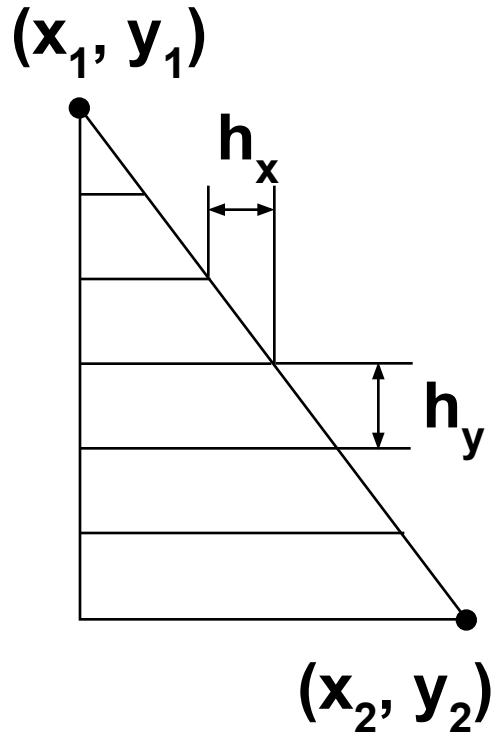
$x$

$x - a$

```
h := (x3 - x2) / (N + 1);
a := x1 - x2;
x := x1 + h;
for i:=1 to N do begin
    Line( round(x), y1, round(x-a), y2);
    x := x + h;
end;
```



# Штрихтер салу



$$h_x = \frac{x_2 - x_1}{N + 1}$$

$$h_y = \frac{y_2 - y_1}{N + 1}$$

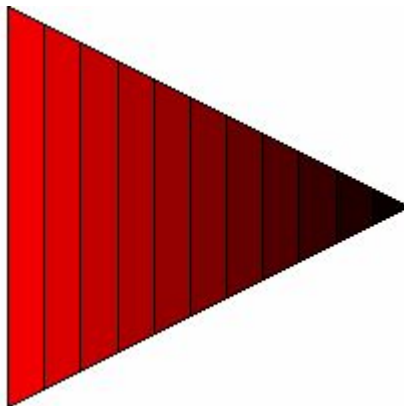
```
Line( x1, y1+hy, x1+hx, y1+hy)
;
Line( x1, y1+2*hy, x1+2*hx,
y1+2*hy);
Line( x1, y1+3*hy, x1+3*hx,
y1+3*hy);
```

```
hx := (x2 - x1) / (N + 1);
hy := (y2 - y1) / (N + 1);
x := x1 + hx; y := y1 + hy;
for i:=1 to N do begin
    Line( x1, round(y), round(x), round(y));
    x := x + hx; y := y + hy;
end;
```

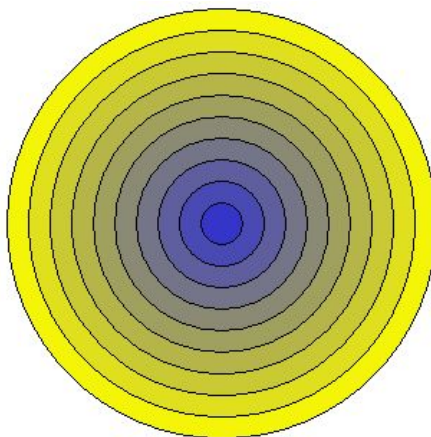
# Тапсырмалар

---

**"4":** Пернетақтадан штрих сызықтарының санын енгізе отырып фигура салу және оның ішін әр түрлі түстермен бояу.



**"5":** Пернетақтадан шеңберлердің санын енгізу арқылы шеңбердің суретін салу және оның ішіндегі әр шеңберді әр түрлі түске бояу.



# Паскаль тілінде программалау

**Тақырып 8. Функциялардың графигі**

# Функция графиктерін құру

---

**Есеп:** 0 мен  $2\pi$  интервалы арасындағы  $y = 3 \sin(x)$  функциясының графигін құру.

**Анализ:**

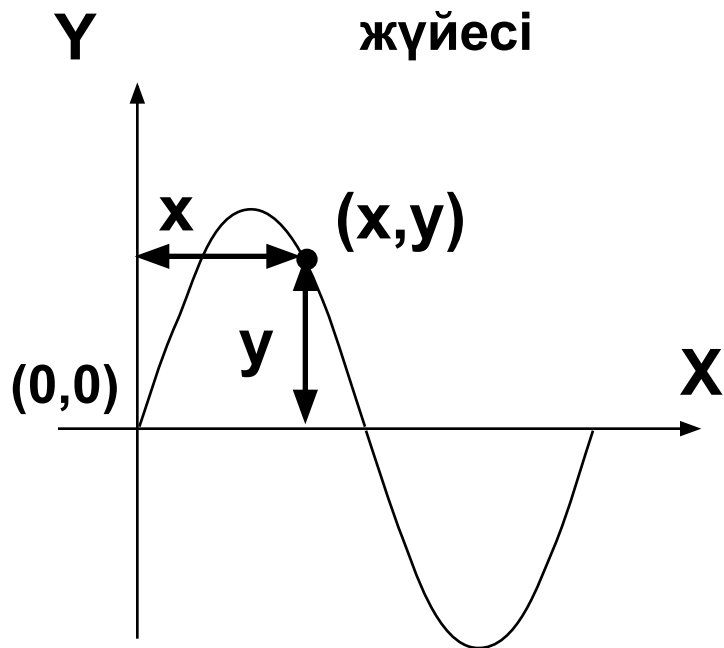
$x = \pi/2$  болғанда максималды мән  $y_{\max} = 3$

$x = 3\pi/2$  болғанда минималды мән  $y_{\min} = -3$

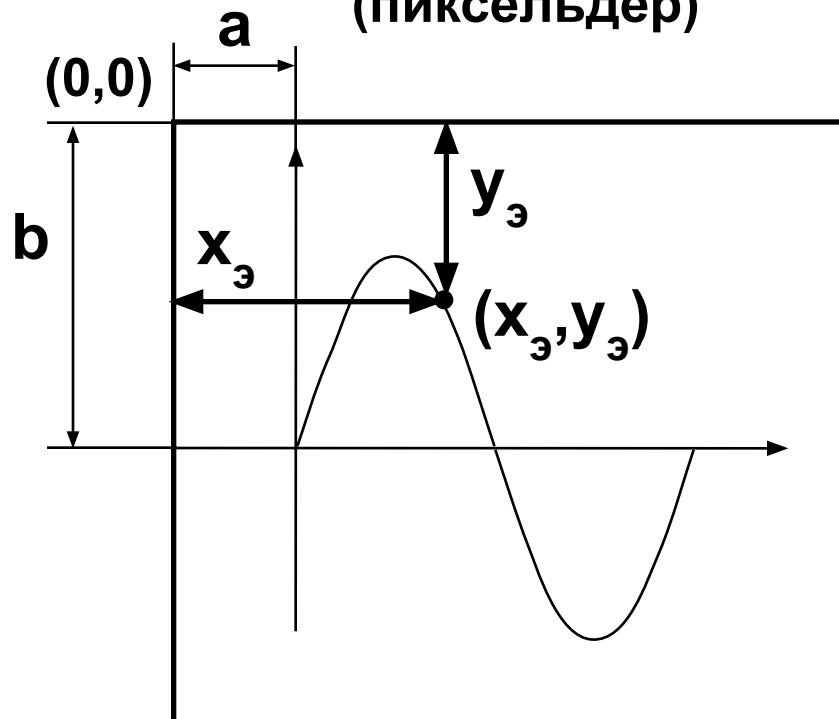
**Мәселе:** функция координалардың математикалық жүйесінде көрсетілген, ал графикті координаталарын пиксельдермен көрсете отырып компьютер экранында шығару керек.

# Координаталардың түрленуі

Координаталардың  
математикалық  
жүйесі



Координаталардың  
экрандық жүйесі  
(пиксельдер)



$k$  — масштаб (экрандағы  
жекелеген кесінді кескінінің  
ұзындығы)

$$x_э = a + kx$$

$$y_э = b - ky$$

# Программа

```

program qq;
const a = 50; b = 200; k = 50;
      xmin = 0; xmax = 6.2832;
var x, y, h: real;
      xe, ye, w: integer;
begin
  w := round((xmax - xmin)*k);
  Line(a-10, b, a+w, b);
  Line(a, 0, a, 2*b);
  x := xmin; h := 0.05;
  while x <= xmax do begin
    y := 3*sin(x);
    xe := a + round(k*x);
    ye := b - round(k*y);
    Point (xe, ye);
    x := x + h;
  end;
end.

```

2π

экранда

h – x-тің өзгеру қадамы

w – OX осінің ұзындығы  
(пиксельдер саны)координаталар  
осіграфиктің  
құрылу циклі

Кемшілігі ?

# Нүктелерді қалай біріктіруге болады?

## Алгоритм:

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

әрекеттердің  
нұсқасын  
таңдау

## Программа:

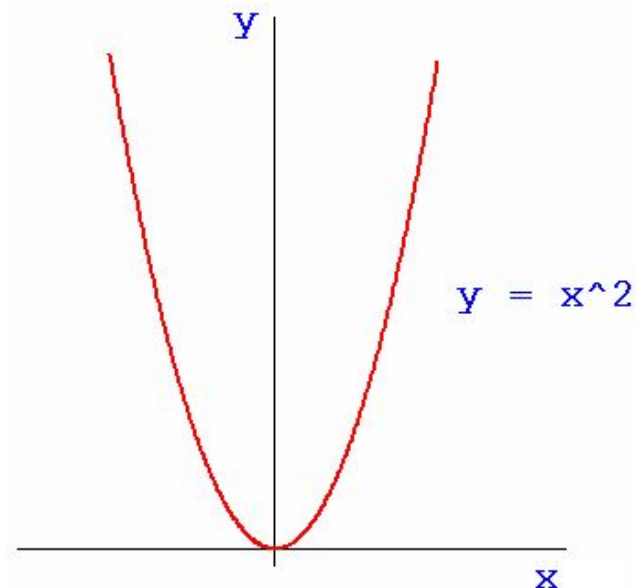
логикалық  
айнымалы

```
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

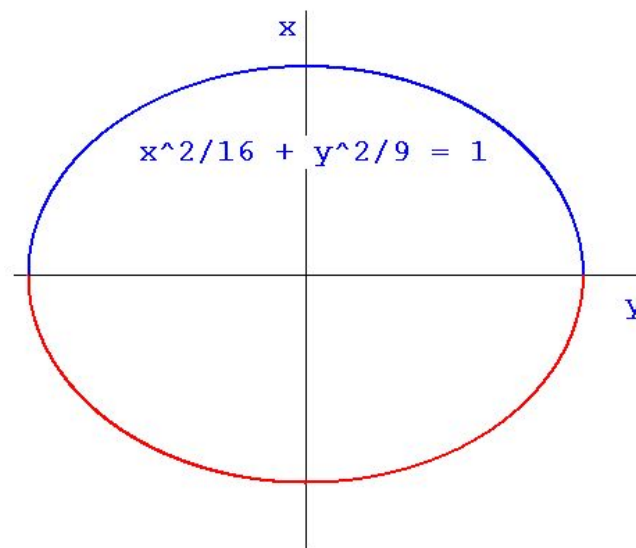
# Тапсырмалар

**"4":**  $[-3,3]$  интервалы арасында  $y = x^2$  функциясының графигін құру.



**"5":** Функция графигін құру (эллипс)

$$\frac{x^2}{16} + \frac{y^2}{9} = 1$$





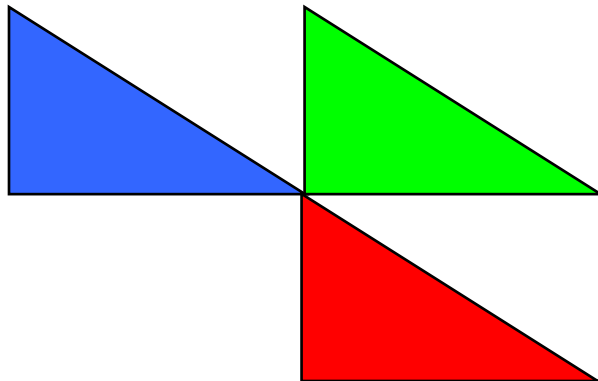
# Паскаль тілінде программалау

## Тақырып 9. Процедуралар

# Процедуралар

---

**Есеп:** Мына фигураны құру:



**Белгілі әдістермен шешуге бола ма?**

**Ерешілімі:** Үш ұқсас фигура.

**жалпы:** өлшемдері, айналу бұрышы

**айырмашылықтары:** координаталары, түстері



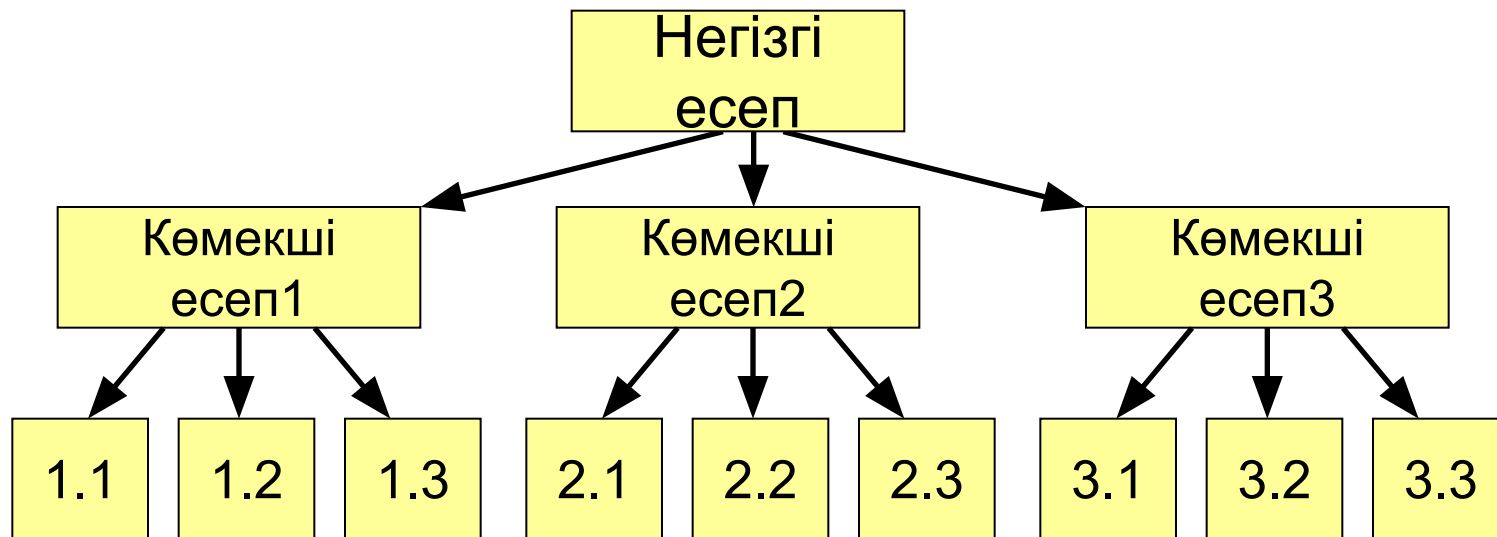
**Неше координата белгілеу керек?**

# Процедуралар

**Процедура** – кейбір әрекеттерді орындау үшін қолданылатын көмекші.

## Қолданылуы:

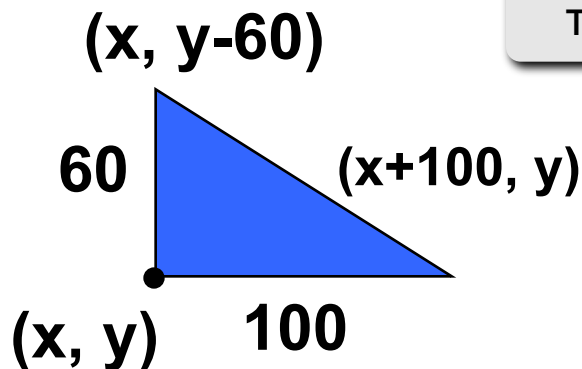
- программаның әр жерінде орналасқан бірдей әрекеттерді орындау
- программаны түсіну оңай болу үшін оны бірнеше қосалқы бөліктерге бөлу



# Процедуралар

## Есепті орындау тәртібі:

- бірдей немесе ұқсас әрекеттерді ерекшелеп алу (үш фигура)
- олардың **ұқсас** жерлері (өлшемі, пішіні, айналу бұрышы) мен **айырмашылықтарын** (координаталары, түсі) табу
- айырмашықтарын белгісіз айнымалылар түрінде өрнектеу, осы айнымалылар процедураның **параметрлері** болады



тақырыбы

параметрлер

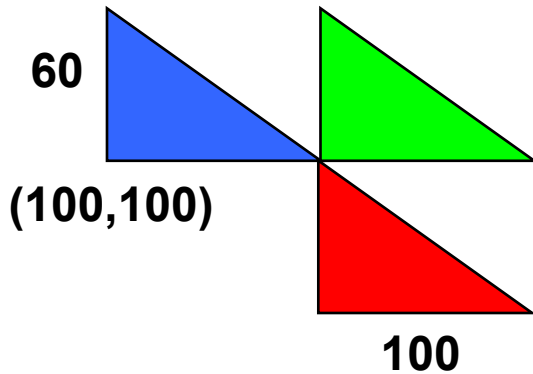
```
procedure Tr( x, y, r, g, b: integer);
begin
  MoveTo(x, y);
  LineTo(x, y-60);
  LineTo(x+100, y);
  LineTo(x, y);
  Brush(1, r, g, b);
  Fill(x+20, y-20);
end;
```

түс

координаталар

процедура тұлғасы

# Программа



процедураны  
шақырулар

формальды параметрлер

```
program qq;  
  procedure Tr( x, y, r, g, b: integer);  
  begin  
    ...  
  end;  
begin  
  Pen(1, 255, 0, 255);  
  Tr(100, 100, 0, 0, 255);  
  Tr(200, 100, 0, 255, 0);  
  Tr(200, 160, 255, 0, 0);  
end.
```

процедура

нақтылы параметрлер

# Процедуралар

---

## Ерекшеліктері:

- барлық процедуралар негізгі программаның **үстінде** орналасқан
- процедураның тақырыбында **формальды** параметрлер жазылады, олар кейде өзгеріп отыратындықтан атаулармен белгіленеді

```
procedure Tr( x, y, r, g, b: integer);
```

- процедураны шақыру кезінде жақша ішінде **нақтылы** параметрлер(сандар немесе арифметикалық өрнектер) көрсетіледі

```
Tr (200, 100, 0, 255, 0);
```

x

y

r

g

b

# Процедуралар

---

## 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

# Процедуралар

## 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



# Параметр-айнымалылар

**Есеп:** екі айнымалы мәндерінің орындарын ауыстыратын процедураны құру.

**Ерекшеліктері:**

Процедурада жасалған өзгертулер шақырушы программаға белгілі болуы қажет.

```
program qq;  
var x, y: integer;  
  
  procedure Exchange ( a, b: integer );  
  var c: integer;  
  begin  
    c := a; a := b; b := c;  
  end;  
  
begin  
  x := 1; y := 2;  
  Exchange ( x, y );  
  writeln ( 'x = ', x, ' y = ', y );  
end;
```

мына процедура  
параметрлердің  
**көшірмесімен**  
жұмыс істейді

**x = 1 y = 2**

# Параметр-айнымалылар

параметрлер өзгере алады

```
procedure Exchange ( var a, b: integer );
var c: integer;
begin
    c := a; a := b; b := c;
end;
```

## Қолданылуы:

процедура(және функция) осындай әдіс арқылы бірнеше мән бере алады.

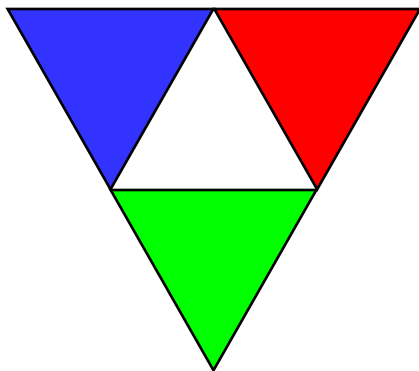
## Шақыртудың тыйым салынған нұсқалары

Exchange ( ~~2~~, ~~3~~ );      { сандар }

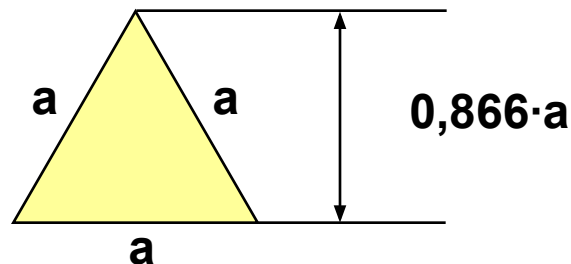
Exchange ( ~~x+2~~, ~~y+2~~ );      { өрнектер }

# Тапсырмалар

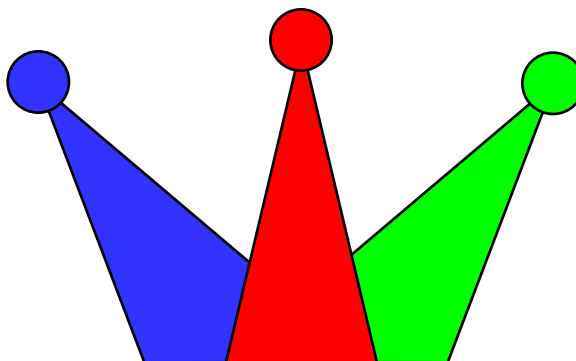
"4": Процедураларды пайдалана отырып, мына фигураны құрастыру.



Тең қабырғалы үшбұрыш



"5": Процедураларды пайдалана отырып, мына фигураны құрастыру.



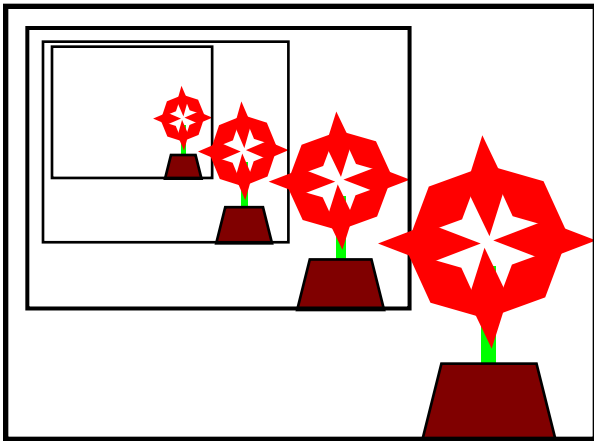
# Паскаль тілінде программалау

## Тақырып 10. Рекурсия

# Рекурсивті объектілер

## Мысалдар:

Рекурсияланған сурет:



Факториал:

$$N! = \begin{cases} 1, & \text{егер } N = 1, \\ N \cdot (N-1)!, & \text{егер } N > 1. \end{cases}$$

$$1! = 1, \quad 2! = 2 \cdot 1! = 2 \cdot 1, \quad 3! = 3 \cdot 2! = 3 \cdot 2 \cdot 1$$

$$4! = 4 \cdot 3! = 4 \cdot 3 \cdot 2 \cdot 1$$

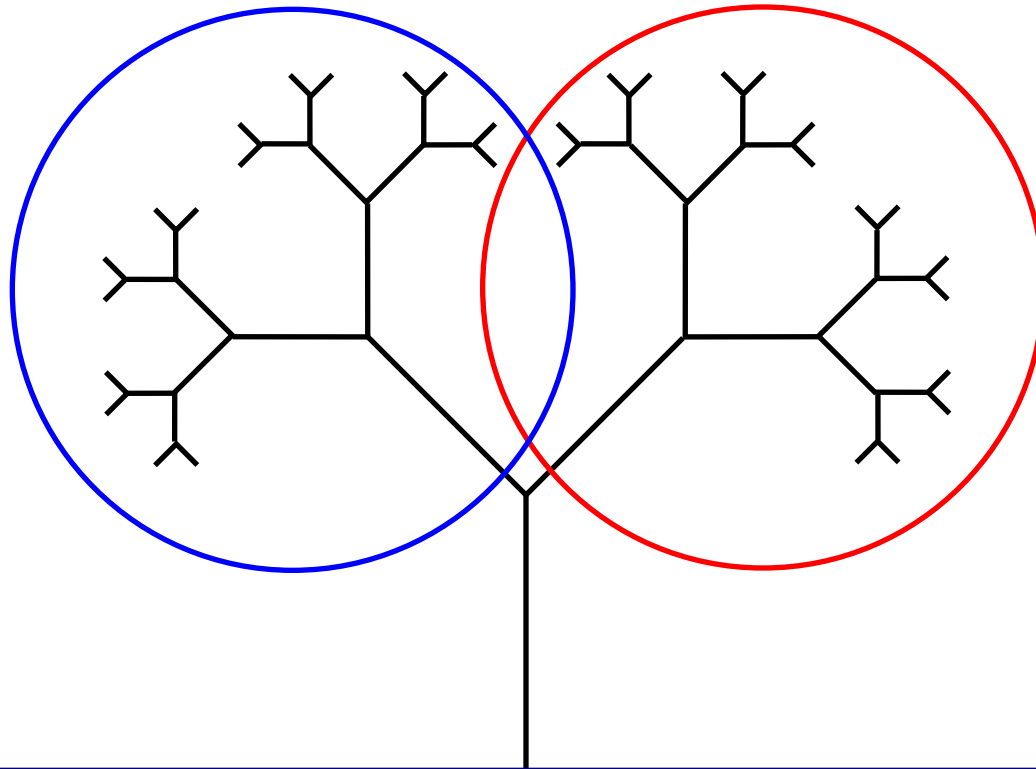
$$N! = N \cdot (N-1) \cdot \dots \cdot 2 \cdot 1$$

**Рекурсивті объект** деп - бір объект немесе дәл осындай бірнеше объект арқылы анықталатын объектіні айтады.

# Пифагор ағашы

**N деңгейлі Пифагор ағашы** – бұл ағаштың діңгегі мен осы діңгектен симметриялы түрде өсіп шыққан **N-деңгейлі екі Пифагор ағашы**, бұл ағаштардың бағанының ұзындығы 2 есе кем және олардың арасындағы бұрыш  $90^\circ$ -қа тең.

**6 деңгей:**



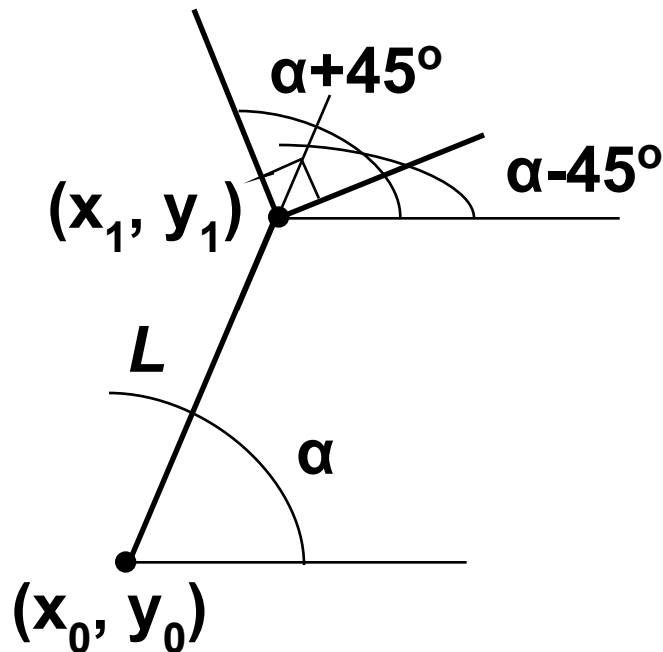
Бұның рекурсивті фигура екенін қалай дәлелдеуге болады?

# Пифагор ағашы

## Ерекшеліктері:

- қашан тоқтау керек?
- ағаштардың иілу бұрышы әр түрлі

**Қалған деңгейлердің саны нөлге тең болғанда!**



$$x_1 = x_0 + L \cdot \cos(\alpha)$$

$$y_1 = y_0 - L \cdot \sin(\alpha)$$

Еншілес ағаштардың иілу бұрышы

$$\alpha + \pi/4$$

$$\alpha - \pi/4$$

# Процедура

$\alpha$  бұрышы

баған ұзындығы

```

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



# Программа

```
program qq;
```

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

```
  ...  
end;
```

$\alpha$  бұрышы

баған ұзындығы

```
begin
```

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

```
end;
```

$x_0$

$y_0$

деңгейлер саны

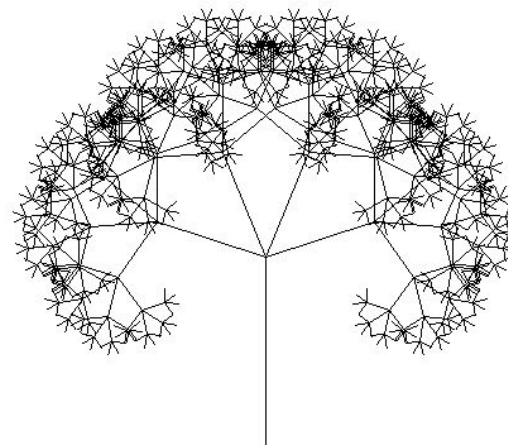
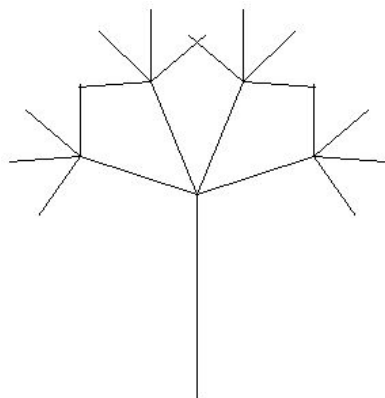


Ағашты  $30^\circ$  оңға қалай еңкейтуге болады?

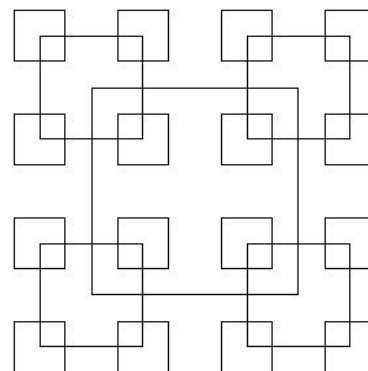
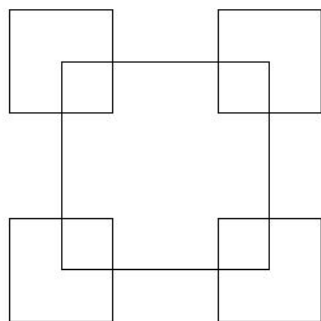
```
Pifagor (250, 400, 2*pi/3, 150, 8);
```

# Задания

**"4":** Рекурсивті процедураны қолдана отырып, мына фигураны құрастыру:



**"5":** Рекурсивті процедураны қолдана отырып, мына фигураларды құрастыру:



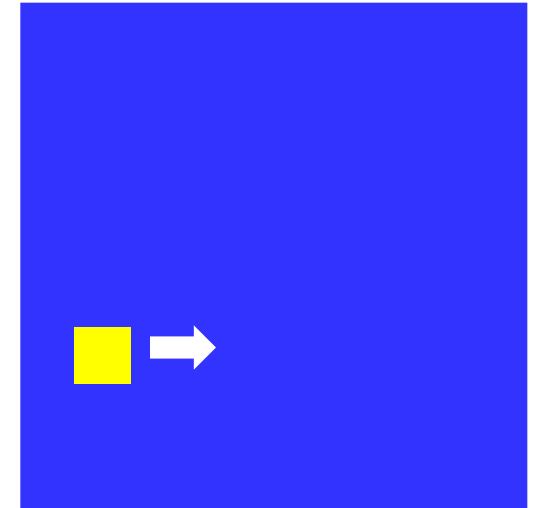
# Паскаль тілінде программалау

## Тақырып 11. Анимация

# 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

## «Пернелерді шертуді» қалай іске асыруға болады?

**Оқиға** деп қандай да бір объектінің қалпында өзгерістің болуын немесе пайдаланушының әрекетін айтады. (пернені басу, тышқанды шерту).

**IsEvent** – қолданушы тарапынан қандай да бір әрекеттің болған-болмағандығын анықтайтын логикалық функция.

**Event** – нақты қандай оқиғаның болғандығын анықтайтын процедура.

```
if IsEvent then begin
    Event(k, x, y);
    if k = 1 then
        writeln('Коды бар перне ', x)
    else { k = 2 }
        writeln('Тышқан: x=', x, ' y=', y);
end;
```

```
var k, x, y: integer;
```

# Esc пернесін басып циклдан қалай шығуға болады?

```
program qq;  
var stop: boolean;  
    k,code,i: integer;  
begin  
    stop := False;  
    repeat  
        if IsEvent then begin  
            Event(k, code, i);  
            if (k = 1) and (code = 27) then  
                stop := True;  
        end;  
        ...  
    until stop;  
end;
```

егер тоқтау қажет болса, True

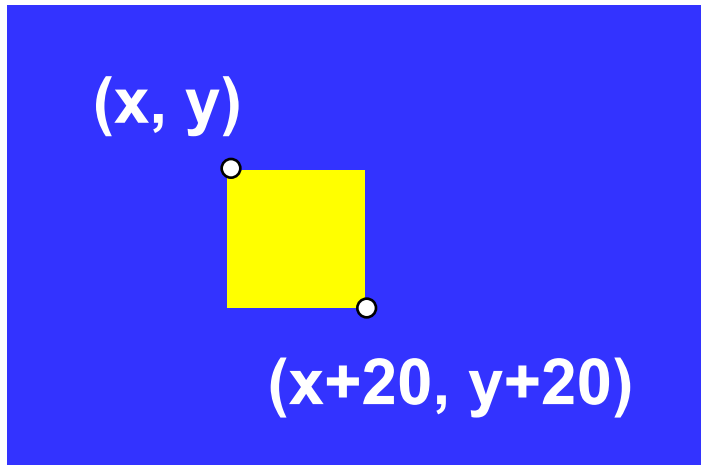
циклді іске қосу

егер бір нәрсе бола қалса...

қандай оқиға  
болды?

егер коды 27 (Esc)  
болатын перне басылса,  
онда стоп

# Процедура (сурет салу және өшіру)



## Идеялар

- бір процедура суретті салады және өшіреді
- өшіру = фонның түсімен бояу
- квадраттың шекарасын өшіріп тастау (негізгі программада)

сурет салу (**True**) немесе салмау (**False**)?

```
procedure Draw(x, y: integer; flag: boolean);  
begin  
  if flag then  
    Brush(1, 255, 255, 0)  
  else  
    Brush(1, 0, 0, 255);  
  Rectangle(x, y, x+20, y+20);  
end;
```

сурет саламыз: қылқаламның түсі  
– сары

суретті өшіреміз: қылқаламның  
түсі – көк

тек қана құю!

# Толық программасы

```

program qq;
var x, y, k, code, i: integer;
    stop: boolean;
procedure Draw(x,y: integer; flag: Boolean);
begin
    ...
end;
begin
    Brush(1, 0, 0, 255);
    Rectangle(10, 10, 400, 400);
    Pen(0, 0, 0, 255);
    x := 10; y := 200; stop := false;
    repeat
        if IsEvent then begin
            ...
        end;
        Draw(x, y, True);
        Delay(10);
        Draw(x, y, False);
        x := x + 1;
        if x >= 400-20 then stop := true;
    until stop;
end.

```

процедура

көк фон

шекараны өшіріп тастау

бастапқы  
жағдайлары

*Esc* пернесімен  
шығу

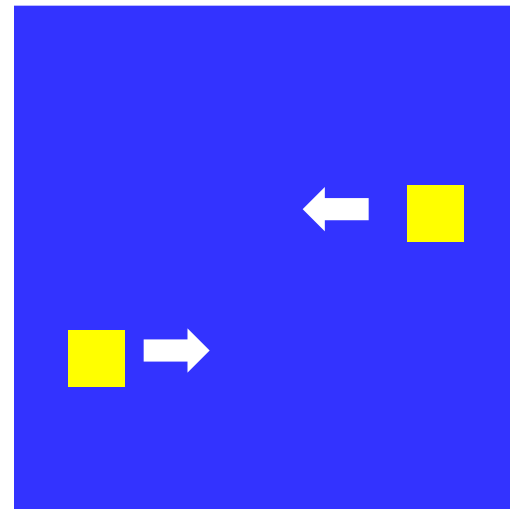
10 мс күтеміз

шекараға тақалған  
кезде шығу

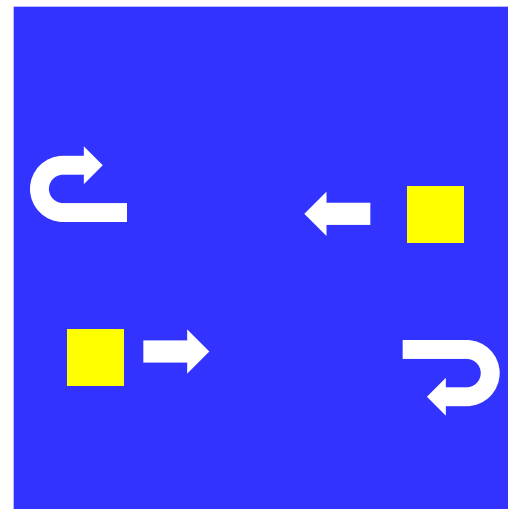


# Тапсырмалар

**"4":** Екі квадрат бір біріне қарама-қарсы бағытта қозғалады:



**"5":** Екі квадрат бір біріне қарама-қарсы бағытта қозғалады және көк квадраттың қабырғасына соқтығысып кері бағытта жылжиды:



# Пернелермен басқару

**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,  
...

# Программасы

```
program qq;  
var x, y, k, code, i: integer;  
    stop: boolean;
```

процедура

```
procedure Draw(x,y: integer; flag: Boolean);  
begin  
    ...  
end;
```

```
begin
```

```
    ...
```

негізгі цикл

```
repeat
```

```
    Draw(x, y, True);
```

```
    Delay(20);
```

```
    Draw(x, y, False);
```

```
    if IsEvent then begin
```

```
        ...
```

```
    end;
```

```
until stop;
```

оқиғаларды  
өңдеу

```
end.
```



Кемшілігі?

# Жыпылықтауды қалай кетіруге болады?

---

**Проблема:** ешбір перне басылмағанның өзінде де квадрат әр 20 мс сайын басқа түске боялып тұрады(жыпылықтау!)

**Бізге қажеттісі:** ешқандай оқиға болмаса квадратты басқа түстерге боямау (жыпылықтатпау)

**Шығарылуы:** квадратты бояп тастап, оқиғаны күту

**Жаңа проблема:** оқиғаны қалай күтеміз ?

**Жаңа проблеманың шығарылуы:** бос цикл «оқиға болмайынша ештеме істеме»:

```
while not IsEvent do;
```

# Программасы

```
program qq;  
var x, y, k, code, i: integer;  
    stop: boolean;
```

процедура

```
procedure Draw(x,y: integer; flag: Boolean);  
begin  
    ...  
end;
```

```
begin
```

```
    ...
```

квадраттың суретін саламыз

```
repeat
```

```
    Draw(x, y, True);
```

оқиғаны күтеміз

```
    while not IsEvent do;
```

енді өшіруге болады

```
        Draw(x, y, False);
```

```
        Event(k, code, i);
```

```
        ...
```

```
until stop;
```

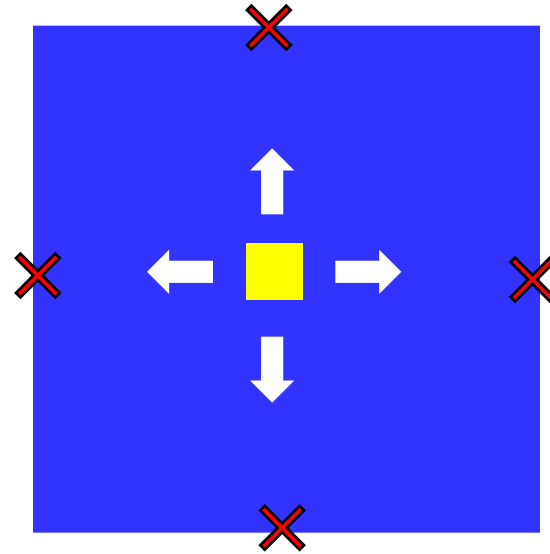
```
end.
```



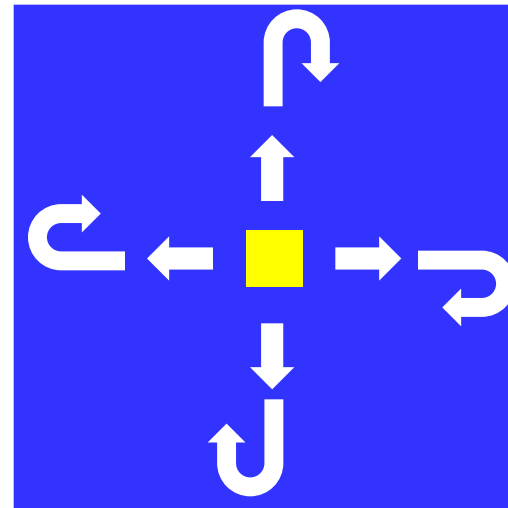
Нені жақсартуға болады?

# Тапсырмалар

**"4":** Квадрат бағыттауыш пернелерді басқан бағытта жылжып отырады, алайда көк квадраттың шекарасынан асып кете алмайды:



**"5":** Квадрат үздіксіз қозғалып отырады, бағыттауыш пернелер басылған кезде өз бағытын өзгертеді және көк квадраттың қабырғаларына соқтығысып кері жылжып отырады:

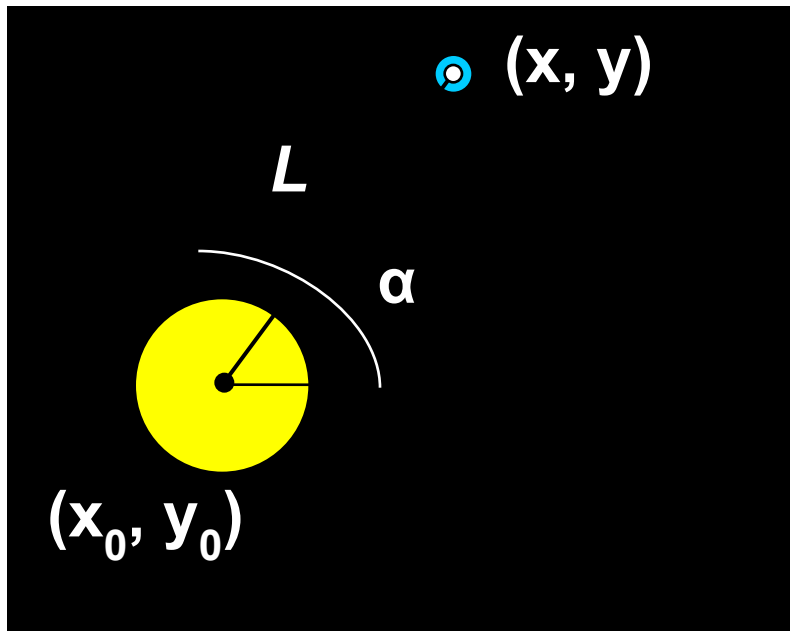


# Айналу

**Есеп:** Жердің Күнді айналу моделін бейнелеу.

**Проблема:** шеңбер бойымен айналу, координаталарды қалай өзгертіп отыруға болады?

**Шығарылуы:** тәуелсіз айнымалы ретінде  $\alpha$  –ның айналу бұрышын қолдану (циклде өзгерту)



$$x = x_0 + L \cdot \cos(\alpha)$$

$$y = y_0 - L \cdot \sin(\alpha)$$

# Процедура

сурет салу (**True**) немесе салмай (**False**)?

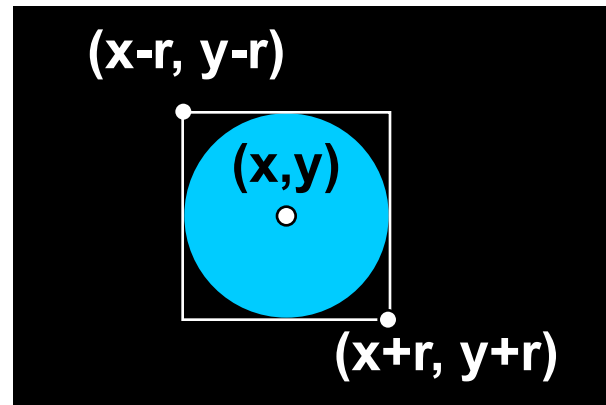
```
procedure Draw(x, y: integer; flag: boolean);  
const r = 10;  
begin  
    if flag then  
        Brush(1, 100, 100, 255)  
    else  
        Brush(1, 0, 0, 0);  
    Ellipse(x-r, y-r, x+r, y+r);  
end;
```

Жердің радиусы

суретін саламыз: қылқалам түсі  
– көгілдір

суретті өшіреміз: қылқалам түсі  
қара

тек қана құю!





# Тұрақтылар мен айнымалылар

```
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.
```

# Негізгі программа

```
program qq;
```

```
...
```

```
begin
```

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

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

```
  Ellipse(x0-rSun, y0-rSun, x0+rSun, y0+rSun);
```

```
  a := 0; ha := 1*pi/180; { бастапқы бұрыш, 100 мс-та 1° Қадам }
```

```
  stop := false;
```

```
  Pen(0,0,0,0); { контурларды өшіріп тастаймыз }
```

```
  repeat
```

```
    x := round(x0 + L*cos(a));
```

```
    y := round(y0 - L*sin(a));
```

```
    Draw(x, y, True);
```

```
    Delay(100);
```

```
    Draw(x, y, False);
```

```
    if IsEvent then begin
```

```
      Event(k, code, i);
```

```
      if (k = 1) and (code = 27) then stop := true;
```

```
    end;
```

```
    a := a + ha;
```

```
  until stop;
```

```
end.
```

фонды қара түске бояу

Күннің суретін саламыз

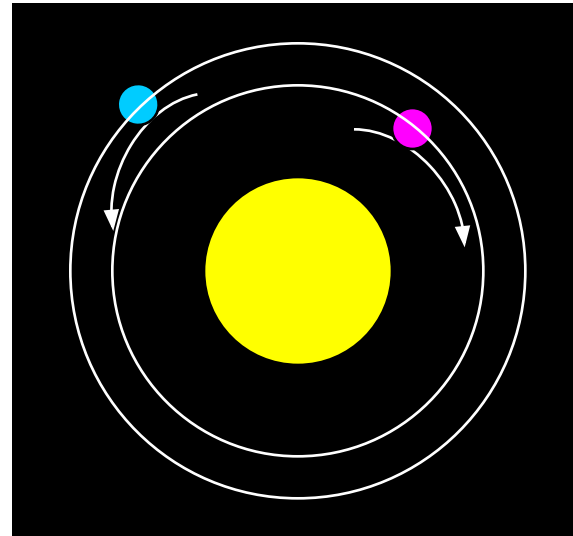
жаңа координаталар

100 мс күтеміз

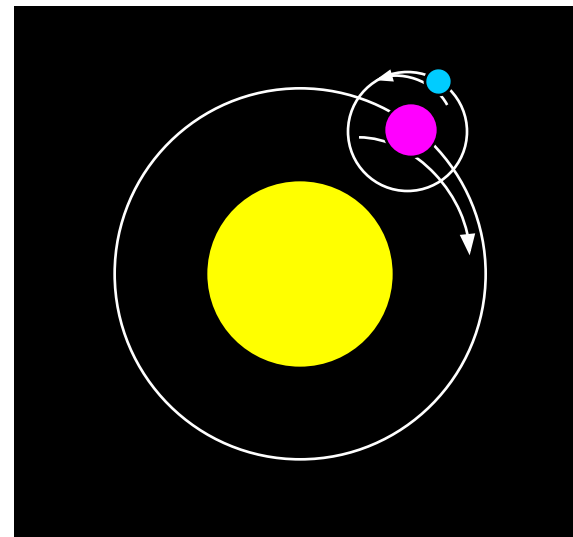
ha бұрышына бұрылу

# Тапсырмалар

**"4":** Күнді айналып жүрген екі планетаның моделін бейнелеу, планеталар бір-біріне қарама-қарсы бағытта қозғалатын болсын:



**"5":** Күн-Жер-Ай жүйесінің моделін бейнелеу:



# Паскаль тілінде программалау

**Тақырып 12. Кездейсоқ сандар**

# 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

# Жалған кездейсоқ сандар

**Жалған кездейсоқ сандар** – бойында кездейсоқ сандардың қасиеті бар, бірақ әрбір келесі сан алдын ала берілген формула бойынша есептелетін сандардың тізбегін айтады.

## Мысалдар:

1. Кездейсоқ бүтін сандар  $[0, m)$  (сызықты конгруэнтті әдіс)

$$x_n = (a \cdot x_{n-1} + c) \bmod m$$

а, с, m - бүтін сандар

$$x_n = (16807 \cdot x_{n-1} + 12345) \bmod 1073741823$$

жай сан

$$2^{30}-1$$

2. Кездейсоқ нақты сандар  $[0, 1]$

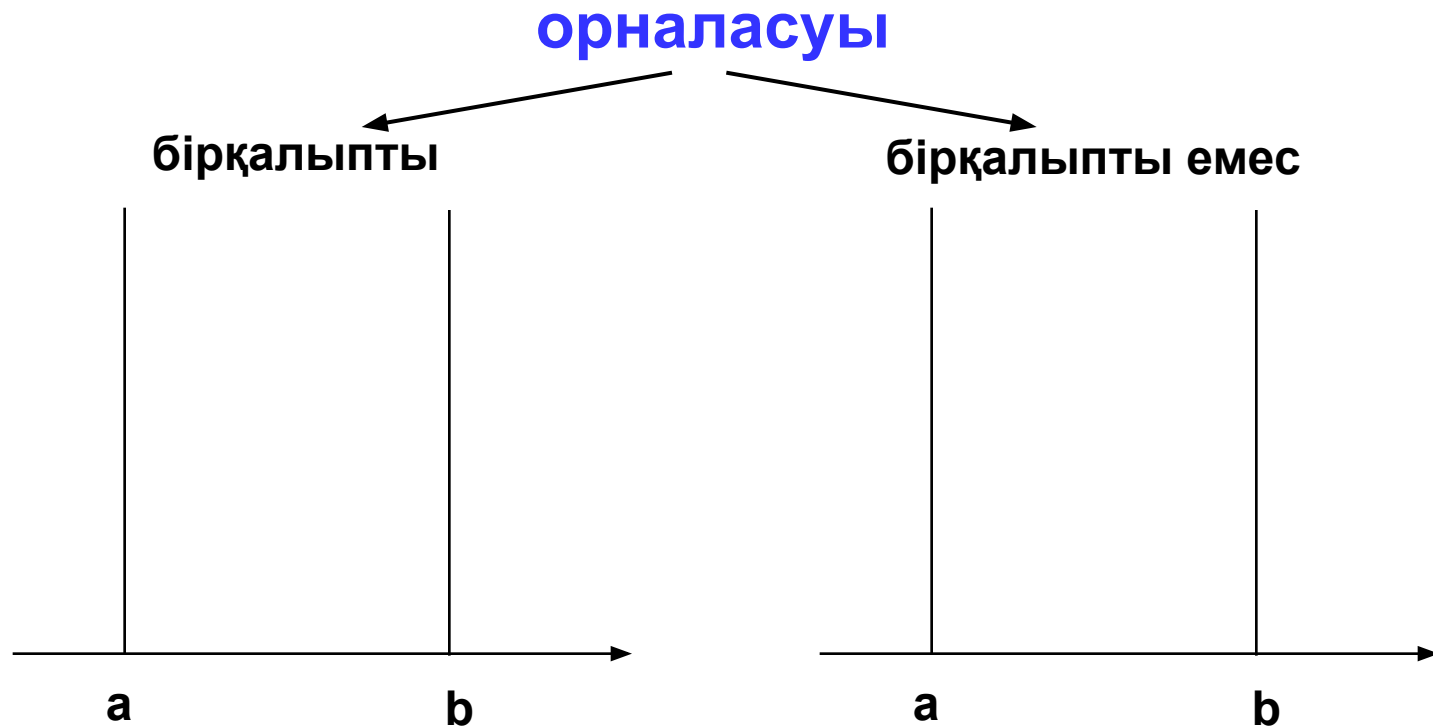
$$x_n = \left\{ (\pi + x_{n-1})^k \right\}$$

мысал,  $k = 5$

санның бөлшек бөлігі

# Кездейсоқ сандардың орналасуы

**Модель:** қар бұршақтары  $[a,b]$  кесіндісінің аралығына түседі




Әр түрлі орналастырулардың саны қанша болуы мүмкін?

# Кездейсоқ сандардың орналасуы


## Ерекшеліктері:

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



$$x = \frac{x_1 + x_2}{2}$$

бірқалыпты орналасу



$$x = \frac{x_1 + x_2 + \dots + x_{12}}{12}$$

бірқалыпсыз орналасу



# Паскальдегі кездейсоқ сандардың генераторы

---

**[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] }
```

# Кездейсоқ сандар

---

**Есеп:** 400x300 пиксельді тіктөртбұрыштың ішін түстері кездейсоқ нүктелермен бірқалыпты етіп толтыру.



**Нүктенің кездейсоқ координаталарын қалай алуға болады?**

```
x := random ( 400 );  
y := random ( 300 );
```

**Бірқалыптылықты қалай алуға болады?**

`random` функциясы қолданылса автоматты түрде қамтамасыз етіледі

**Кездейсоқ түстерді қалай алуға болады?**

```
Pen (1, random(256), random(256), random(256)) ;  
Point ( x, y ) ;
```

# Программа

```
program qq;  
var x, y, k, code, i: integer;  
    stop: boolean;  
begin  
    stop := False;  
    repeat  
        x := random(400);  
        y := random(300);  
        Pen(1, random(256), random(256), random(256));  
        Point(x, y);  
        if IsEvent then begin  
            Event(k, code, i);  
            if (k = 1) and (code = 27) then stop := True;  
        end;  
    until stop;  
end.
```

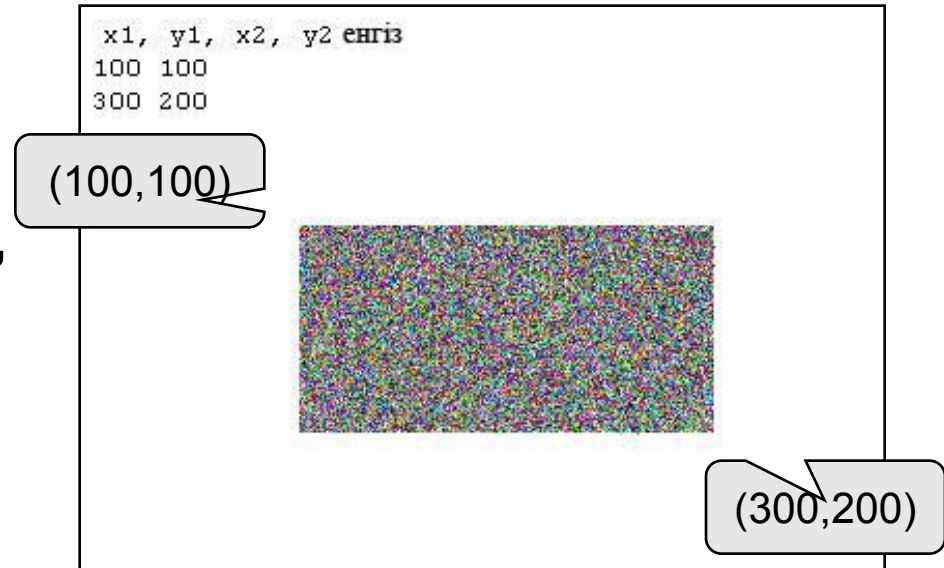
кездейсоқ координаталар

кездейсоқ түс

**Esc** пернесін басып шығып кету

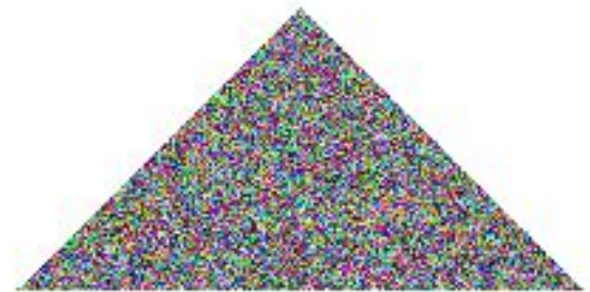
# Тапсырмалар

**"4":** Пернетақтадан тіктөртбұрыш бұрыштарының координаталарын енгізіп, тіктөртбұрыштың ішін түстері кездейсоқ нүктелермен толтыру керек.



**"5":** Үшбұрыштың ішін түстері кездейсоқ болатын нүктелермен толтыру. (бірқалыпты немесе бірқалыпты емес).

**Қосымша көмек:** бұрышы  $45^\circ$  болатын теңбүйірлі үшбұрышты алу керек.



# Паскаль тілінде программалау

## Тақырып 13. Функциялар

# Функциялар

---

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

## Мысалдар:

- $\sin x$ ,  $\cos x$ ,  $\sqrt{x}$  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



Процедурадан айырмашылығы неде?

# Функциялар

**Есеп:** екі санның үлкенін анықтайтын функция құру және оны қолдануға мысал келтіру

**Функция:**

формальды параметрлер

```
function Max (a, b: integer): integer;  
begin  
  if a > b then Max := a  
  else       Max := b;  
end.
```

бұл функцияны нәтижесі

# Функциялар

## 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 қоспүктеден кейін **нәтиженің типі** көрсетіледі

- функция **function Max (a, b: integer): integer;** ласады



# Функциялар

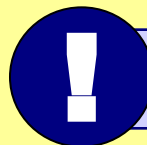
## Ерекшеліктері:

- жергілікті айнымалыларды жариялауға және пайдалануға болады

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

- Нәтиже сол атымен аты функцияның атымен сәйкес келетін айнымалыға жазылады; оны жариялаудың **ҚАЖЕТІ ЖОҚ**:

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



**Delphi-де:** Result := a;

# Программа

```
program qq;  
var a, b, c: integer;  
  
function Max (a, b: integer): integer;  
begin  
    ...  
end;  
  
begin  
    writeln('Екі сан енгіз');  
    read(a, b);  
    c := Max ( a, b );  
    writeln('Санның Үлкені ', max c  
end.
```

нақты параметрлер

функцияны шақыру



**Айнымалылардың, функциялардың және процедуралардың аттары бірдей болмауы керек!**

# Логикалық функциялар

**Есеп:** енгізілген санның - жай сан екендігін анықтайтын функция құру.

**Ерекшеліктері:**

- жауабы – логикалық мән (True немесе False)
- функцияның нәтижесін `if`, `while` шарттарында логикалық шама ретінде қолдануға болады

**Алгоритм:** 2-ден N-1-ге дейінгі аралықтағы бөлгіштердің санын есептейміз, егер олардың саны нөлге тең болмаса енгізілген санның - құрама сан болғаны.

```
count := 0;
```

```
for i := 2 to N-1 do
```

```
  if N mod i = 0 then
```

```
    count := count + 1;
```

```
if count = 0 then
```

```
  { N саны - жай сан }
```

```
else { N саны - құрама сан }
```



Қалай жақсартуға болады?

# Логикалық функциялар

```
program qq;  
var N: integer;
```

нәтижесі – логикалық мән

```
function Prime (N: integer): boolean;
```

```
var count, i: integer;
```

```
begin
```

```
  i := 2; count := 0;
```

```
  while i*i <= N do
```

```
    if N mod i = 0 then count := count + 1;
```

```
    i := i + 1;
```

```
  end;
```

```
  Prime := (count = 0);
```

```
end;
```

таңдап алу тек  $\sqrt{N}$ -ге дейін

шарт дегеніміз – логикалық мән

```
begin
```

```
  writeln('Бүтін сан енгіз');
```

```
  read(N);
```

```
  if Prime(N) then
```

```
    writeln(N, ' - жай сан')
```

```
  else writeln(N, ' - Құрама сан');
```

```
end.
```

функцияны шақыру

# Тапсырмалар

---

**"4":** 1-ден N-ге дейінгі сандардың қосындысын анықтайтын функция құру және оны қолдануға мысал келтіру.

**Мысал:**

Сан енгіз:

100

Қосынды = 5050

**"5":** Шахматты ойлап шығарған адамның N-ші торға неше бидайдың қоюын сұрағанын анықтайтын функция құру. (1-шісіне – 1 дән, 2-шісіне – 2 дән, 3-шісіне – 4 бидай дәнін, ...)

**Мысал:**

Тордың нөмірін енгіз:

28

28-шісі торда 134217728 бидай дәні.

# Тапсырмалар (2-нұсқа)

**"4":** Екі натурал санның ең үлкен ортақ бөлгішін анықтайтын функция құру және оны қолдануға мысал келтіру.

**Мысал:**

Екі сан енгіз:

**14 21**

$$\text{ЕҮОБ}(14, 21) = 7$$

**"5":** Синус функциясын қатардың қосындысы секілді есептейтін функция құру. (0.001 дәлдікпен)

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

**Мысал:**

**x радиан түрінде!**

Градус түріндегі бұрышты енгіз:

**45**

$$\sin(45) = 0.707$$

# Фильмнің соңы

---