

# *Алгоритмдеу және программалау*

# 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

---

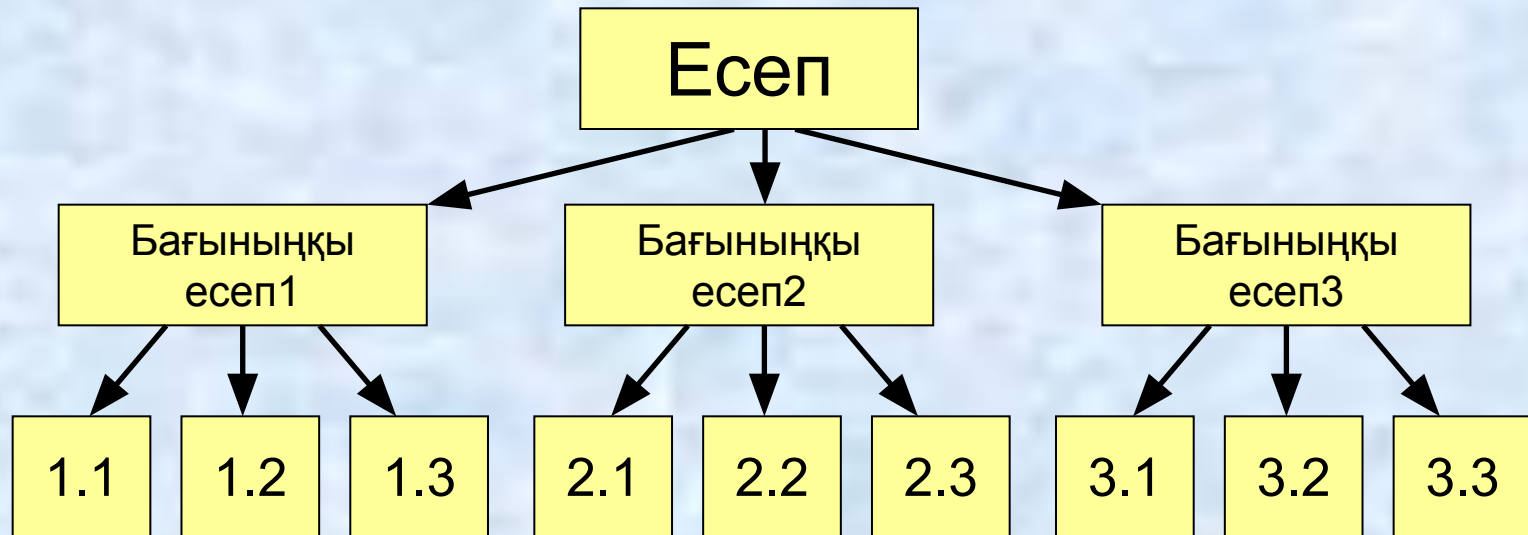
- **Машинаға бейімделген (төменгі деңгейлі)** - әрбір команда процессордың бір командасына сәйкес келеді (ассемблер)
- **Жоғарғы деңгейлі тілдер** – кәдімгі табиғи тілге (ағылшын тіліне) ұқсайды, адамның түсінуіне оңай, **бір ғана компьютерге тәуелді болмайды.**
  - *Жаңадан үйренушілерге*: Бейсик, ЛОГО, 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;
```

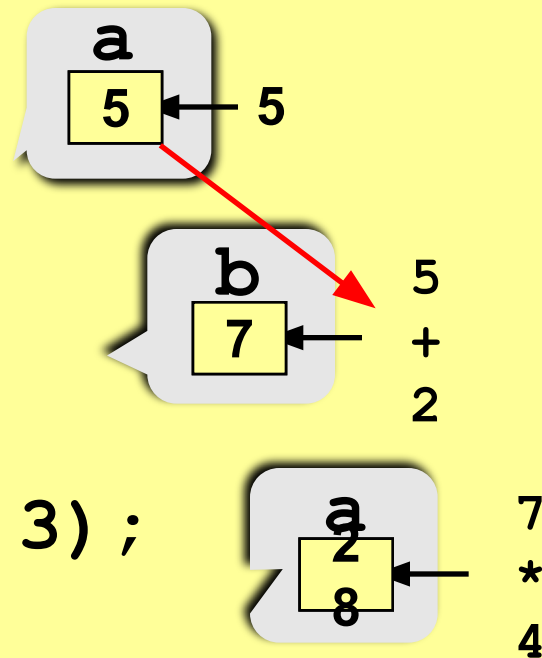
# Айнымалының мәнін қалай өзгертуге болады?

**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

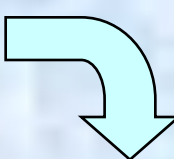
нақты мәнді бүтін  
айнымалыға меншіктеуге  
болмайды

# 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.
```

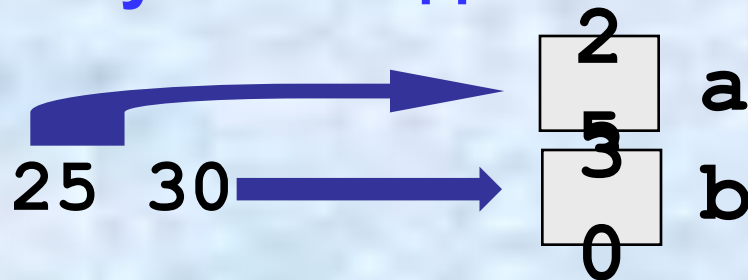
# Енгізу операторы

---

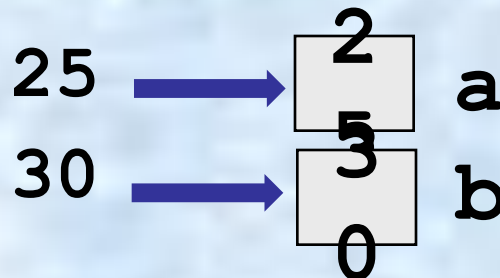
```
read ( a );      { a айнымалысының мәнін  
                  енгізу}  
read ( a, b );   { a және b  
                  айнымалыларының мәндерін  
                  енгізу}
```

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**

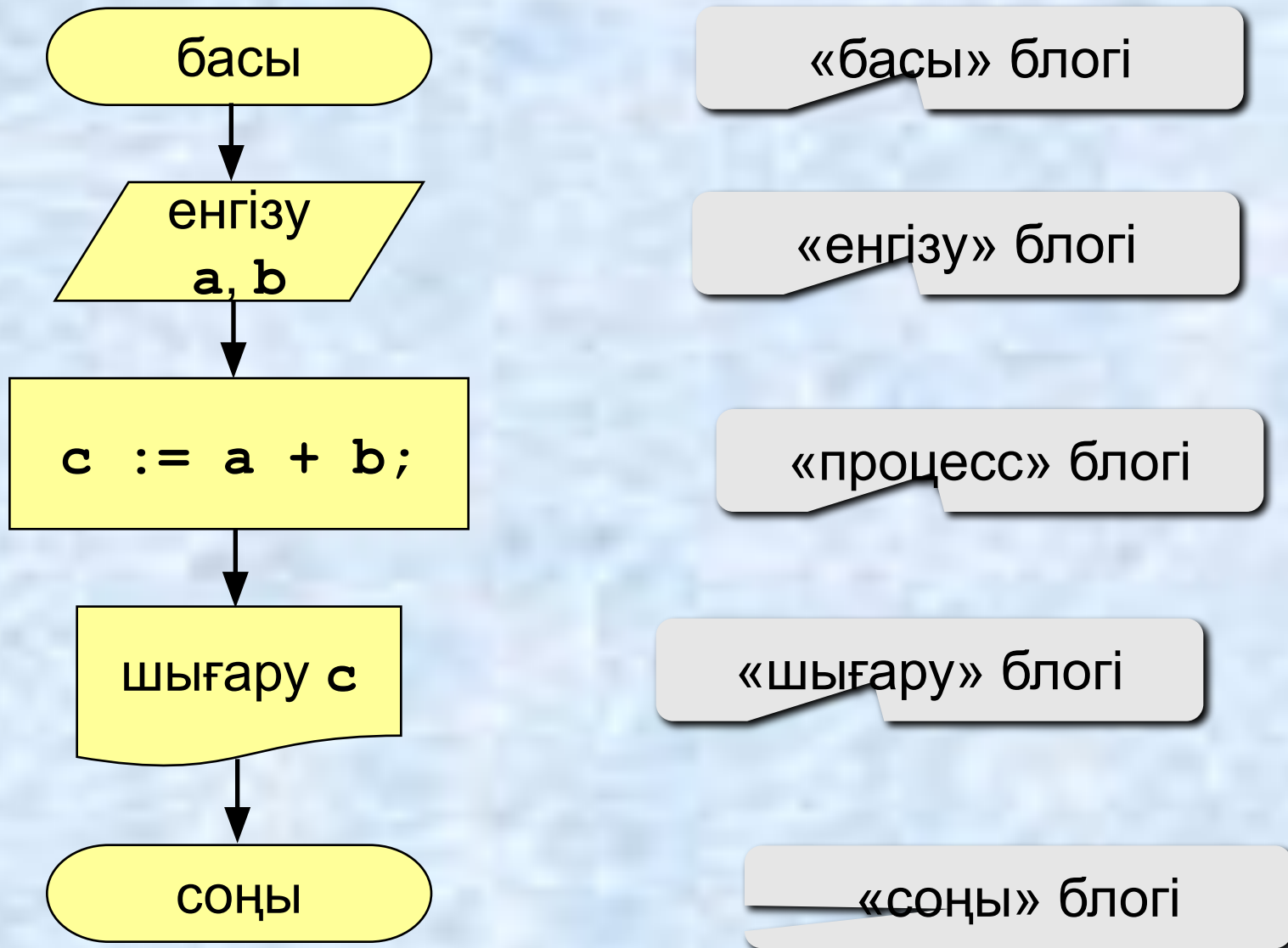
**25 30**

**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

$$4+5+7=16$$

$$4*5*7=140$$

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