

# Программалау негіздері

Кіріспе

# Мәліметтер және алгоритмдер

Алгоритм + Мәліметтер құрылымы =  
Программа

Алгоритм – есепті шығару жолында орындалатын әрекеттердің реттелген тізбегі

Программа – есепті шығару барысында орындалатын әрекеттердің (алгоритмнің) компьютерге түсінікті тілде жазылуы

# Ақпарат туралы

- Ақпарат– білім айқындылығы деңгейін азайту
- Ақпарат және оның мазмұны
- Ақпаратты нақты объектілермен сәйкестендіру **кодтау** деп аталады
- Ақпарат – дискретті ұғым болып саналады

# Ақпаратты өлшеу бірліктері

- Ең кіші өлшем бірлігі – 1 бит  
(екі дискретті қалыппен бейнеленеді, 1 және 0)
- 1 байт – 8 бит, мысалы, А – 65 – 01100101  
(256 қалып күй, бір символды бейнелеу үшін пайдаланылады)
- 1 сөз – 2-4-8 байт болады, компьютер платформасына байланысты өзгеріп отырады
- Туынды өлшем бірліктері
  - 1К = 1024 байт
  - 1М = 1024К = 1 048 576 байт
  - 1Г = 1024М = 1 073 741 824 байт

# Мәліметтердің компьютерлік модельдері

- Бүтін сандар  
(8-16-32-64 бит, таңбалы/таңбасыз)
- Адрестер  
(көбінесе 32 бит, бұлар да бүтін сандар)
- Символдар  
(8-16 бит, бұлар да бүтін сандар)
- Нақты (аралас) сандар  
(32-64-80 бит, мантисса + экспонента)

# Мәліметтер құрылымы

- Мәліметтердің скалярлық (жай) типтері
  - Бүтін
  - Нақты
  - Символдық
  - Логикалық
- Жиымдар (массивтер)
- Құрылымдар
- Объектілер

# Алгоритм

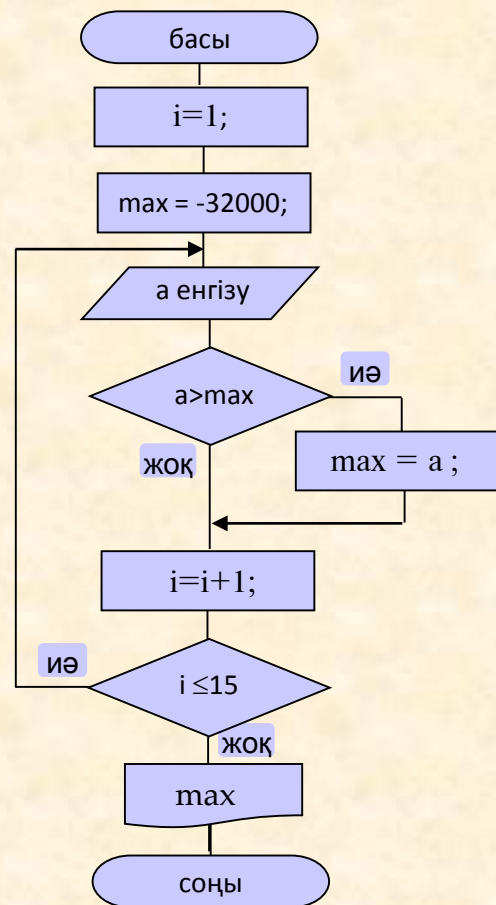
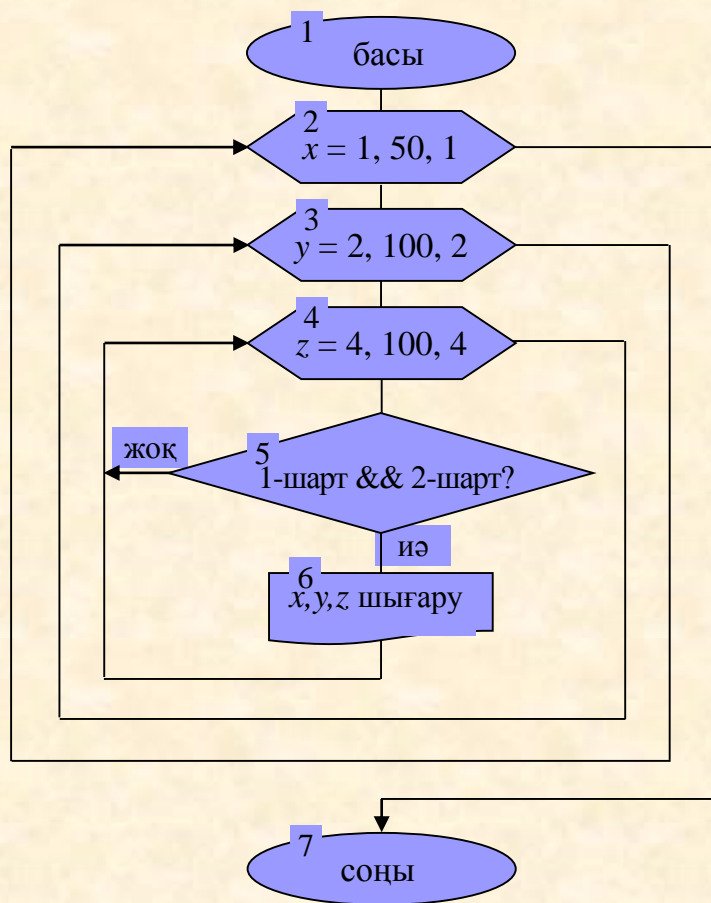
- Ис-әрекеттер (амалдар) тізбегі
  - Шектелген тізбек болуы тиіс
  - Бір мәнді болып табылады
- Алгоритмнің сипатталуы
  - Табиғи тілде
  - Графикалық түрде (блок-схемалар)
  - Алгоритмдік тілде

# Блок-схемалар

	Басы/соңы
	Әрекет (процесс)
	Енгізу/шығару
	Шарт, цикл басы/соңы (қайталау)
	Ішкі программаны шақыру
	Параметрлі цикл



# Блок-схема мысалдары



# Программалау тілдері

- Әмбебап тілдер
  - C/C++
  - C#
  - JAVA
  - FORTRAN
  - BASIC
  - PASCAL
  - PROLOG
- Арнайы (мамандандырылған)
  - HTML
  - JavaScript
  - SQL
  - UNIX-shell
  - VBA
  - VBScript
  - PHP

# Алгоритмдік тілдер

- C/C++
- C#
- Pascal/Delphi
- Visual Basic
- Java

# Программалау тілдерінің дамуы

- 1960-ж Fortran, Algol, Basic, Cobol, Lisp
- 1970-ж Pascal, C, Simula, Smalltalk, Prolog
- 1980-ж C++, Object Pascal, Ada, Occam
- 2000-ж C#, Java, PHP

# Тілді абстракциялау деңгейі

- Ассемблер
- Fortran
- C
- Pascal
- C++
- C#



# Машиналық код және ассемблер тілі

00	61		LD 1
01	62		LD 2
02	10		ADD
03	53	06	GT A
05	18		SGN
06	41		A: ST 1
07	50		STOP

# Трансляторлар

- Компилятор + біріктіргіш (компоновщик)  
(C/C++, Pascal)
- Интерпретатор  
(Basic, JavaScript, HTML)
- Аралық тілдерді қолдану және виртуалдық машинаны пайдалану  
(C#, Java)


# Программаны компиляциядан өткізу кезеңдері





# Программа құру кезеңдері

- Мәселенің (есептің) қойылымы
- Мәлімет құрылымдарын анықтау
- Алгоритмді жобалау
- Программалау (кодтау)
- Жөндеп түзету (отладка – debug)
- Тесттен өткізу
- Пайдалану



**Үзіліс 5 мин.**

# C тілі

## 1 дәріс

Айнымалылар және  
мәліметтер типтері,  
меншіктеу операциясы

# Программа мысалы

```
#include <stdio.h>
```

```
/* 1-мысал - қарапайым программа */
```

```
main ()
```

```
{
```

```
    int year, month;
```

```
    year = 2014;    month=2;
```

```
    printf ("Биыл %d жыл", year);
```

```
}
```

# Идентификаторлар жəне айнымалылар

- **Идентификатор** (атау) – әріптерден басталатын әріптер мен цифрлар тізбегі, (C/C++ тілінде атау регистрге тәуелді болады)
- **Айнымалы** – программаны орындау барысында өз мәнін өзгертетін шама, идентификатормен белгіленеді де, мәліметтердің белгілі бір типіне жатады
- **Мәліметтер типі** – айнымалы қабылдай алатын мәндер жиыны (және де оған қолданылатын операциялар жиыны)

# Мәліметтердің бүтін типтері

Типі	Ені	Диапазоны
unsigned char	1	0 ... 255
char	1	-128 ... +127
short int	2	-32 768 ... +32 767
unsigned int	4	0 ... 4 294 967 295
int	4	-2 147 483 648 ... 2 147 483 647
unsigned long	4	0 ... 4 294 967 295
long	4	-2 147 483 648 ... 2 147 483 647

# Мәліметтердің нақты типтері

Типі	Ені	Дәлдігі	Диапазоны
float	4	6-7	$\pm 3.4 \cdot 10^{\pm 38}$
double	8	15-16	$\pm 1.7 \cdot 10^{\pm 308}$
long double	10	19-20	$\pm 3.4 \cdot 10^{\pm 4932}$

Мантисса 1234567890123456	Дәреже +15
------------------------------	---------------

0.1234567890123456 · 10<sup>+15</sup>

# Айнымалыларды сипаттау

- `int i, j;`
- `float x;`
- `double a, b, c;`



## Тікелей константаларды жазу

Константа типі	Мысалы
int	12, +45, -1 567 439
int	055 (8-дік жүйе), 0x1A (16-лық жүйе)
unsigned int	600000, 123432135
long int	5L, 100000
short int	10, 32H, -10h
float	2.12, 1.0, 1234.0f, 3E+6, 2.1E-4f
double	123.43, 6.67E-34
long double	1.39L, 12.0l
char	'a', '\n', '1'

# Арнайы константаларды жазу

<code>\n</code>	жаңа жол
<code>\r</code>	каретканы қайтару
<code>\t</code>	көлденең (горизонтал) табуляция
<code>\"</code>	қостырнақша
<code>\'</code>	апостроф
<code>\\</code>	кері қиғаш сызық
<code>\0</code>	нөлдік символ
<code>\123</code>	123 - сегіздік кодтағы символ
<code>\xA1</code>	A1 - оналтылық кодтағы символ

# Меншіктеу операциялары

- `n=1;`
- `n=k*5;`
- `n=n+1;`
- `a=2.99792E+10;`
- `x=a/1000.0`
- `i=j=0;`            `i=(j=0);`

# Мәліметтер типінің сәйкестігі

- Меншіктеу таңбасының сол жағындағы айнымалы типі оң жақтағы өрнек мәніне сәйкес келуі тиіс!
- Бірақ бірнеше ерекшеліктер бар
- Типтерді тікелей бір түрге келтіруге болады

# Типтерді автоматты түрде келтіру

```
int i;  
float x;  
double y;  
char c;
```

```
x = i; // бұлай жазуға болады
```

```
i = x; // ескерту немесе қате
```

```
c = i; // болады, бірақ биттерді жоғалту  
// мүмкіндігі бар
```

```
x = y; // болады, бірақ дәлдікті жоғалтамыз
```

# Типтерді тікелей түрде келтіру

```
int i;
```

```
float x;
```

```
double y;
```

```
char c;
```

```
x = (float) i;
```

```
i = (int) x;
```

```
i = int(y);
```

Типтерді келтіруді дұрыс түсінбеу –  
программадағы қиын анықталатын қателер көзі

# Айнымалыларды инициалдау

- **int** i = 0, j = 0;
- **float** x = 1.0;
- **double** a,b,c = 2.718281828;

Типтерді келтіруді дұрыс түсінбеу –  
программадағы қиын анықталатын қателер көзі

# Бинарлық операциялар

+

– қосу

–

– азайту

\*

– көбейту

/

– бөлу

%

– модуль бойынша бөлу  
(қалдық табу)



## Бүтінсандық арифметикалық өрнектер

- $n$
- $25*n+3$
- $25*(n+3)$
- $n/2$  – бүтіндерді бөлу!  $5/2 = 0$
- $n\%2$  – бөлгендегі қалдық  $5\%2 = 1$

# Унарлық операциялар

- – унарлық минус
- ++ – бірге арттыру
- – бірге кеміту

```
i=1;  
j=i++;           // j=1  
k=++i;          // k=3  
n=--i;          // n=2
```

## Нақты арифметикалық өрнектер

- $x$
- $25.0 * x + 3$
- $25 * (x + 3.0)$
- $x / 2$  – нақты сандық бөлу  $5.0 / 2 = 2.5$
- $2 * \sin(x)$

Өрнектер құрамындағы функциялар, айнымалылар, сандар, константалар **операндтар** болып табылады

# Меншіктеу операциялары

- =** Сол жақ операндқа оң жақтағы мәнді тікелей меншіктеу
- +=** Сол жақтағы операнд пен оң жақтағы операнд мәндерін қосып, нәтижені сол жақтағы операндқа меншіктеу
- Сол жақтағы операнд мәнінен оң жақтағы операнд мәнін алып, нәтижені сол жақтағы операндқа меншіктеу
- \*=** Сол жақтағы операнд пен оң жақтағы операндты көбейтіп, нәтижені сол жақтағы операндқа меншіктеу
- /=** Сол жақтағы операнд мәнін оң жақтағы операндқа бөліп, нәтижені сол жақтағы операндқа меншіктеу

# Меншіктеу операциялары мысалдары

```
x=a+b=c-d++;
```

```
m*=2;
```

```
max = (x>y) ? x : y;
```

```
a=sin(x)*cos(y);
```

```
c=sqrt(2.0);
```

# Қарапайым программа

```
#include <stdio.h>
```

```
void main()
```

```
{ float R;
```

```
float L, S;
```

```
printf("Enter R ");
```

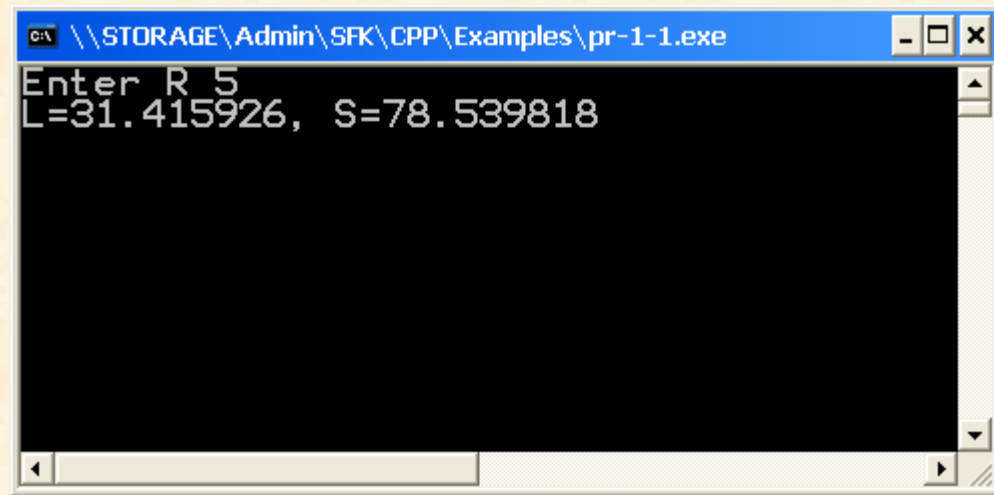
```
scanf("%f", &R);
```

```
L=3.1415926*2*R;
```

```
S=3.1415926*R*R;
```

```
printf("L=%f, S=%f", L, S);
```

```
}
```



```
C:\ \\STORAGE\Admin\SFK\CPP\Examples\pr-1-1.exe
Enter R 5
L=31.415926, S=78.539818
```

# printf функциясы

```
printf("Hello, World!\n");  
printf("x=%f, y=%f\n", x, y);  
printf("%05d", 15); // 00015 мәнін береді
```

**%c** – символ

**%d** – бүтін ондық сан

**%i** – бүтін ондық сан

**%o** – бүтін сегіздік сан

**%x** – бүтін он алтылық сан (a1)

**%X** – бүтін он алтылық сан (A1)

**%u** – таңбасыз ондық сан

**%f** – нақты сан *xx.xxx*

**%F** – нақты сан *xx.xxx*

**%e** – нақты сан *x.xx e+xx*

**%E** – нақты сан *x.xx E+xx*

**%G** – **%F** немесе **%E** (қысқалауын береді)

**%g** – **%f** немесе **%e** (қысқалауын береді)

**%s** – символдар тіркесі

**%p** – нұсқауыш

**%%** – символ %

**l, h (L, H)** **%ld** – long int модификаторлары, **%hu** – short unsigned, **%Lf** – long double

**%8.2F, %16E, %8d** дәлдік модификаторлары

# scanf функциясы

`%d` – бүтін ондық сан

`%i` – бүтін ондық сан

`%o` – бүтін сегіздік сан

`%x` – бүтін он алтылық сан

`%h` – `short int` типін оқу

`%e` – `float` типіндегі санды оқу

`%c` – СИМВОЛ

`%s` – СИМВОЛДАР ТІРКЕСІН ОҚУ

`%p` – НҰСҚАУЫШ

```
scanf("%d%*c%d", &i, &j);
```

20+50

$i \rightarrow 20, j \rightarrow 50, *$  – көрсетілген типтегі мәліметтерді оқу,  
бірақ оларды есепке алмау

```
scanf("%5s", &s); // өрістің ең үлкен мәні
```

```
scanf("%dplus%d", &i, &j); // plus мәтінін есепке алмау
```



Мысалы:

Әрбір спецификация % символынан басталып, түрлендіру символымен аяқталады. Ол екеуінің ортасында мыналар тұруы мүмкін:

**-минус** таңбасы, аргумент мәні сол жақ шетке ығыстырылып жазылады.

**-цифрлар**, бүтін санның жалпы орналасу енін анықтайды. Сан осы енге немесе одан артық болып шығарылады. Егер аргумент ені көрсетілген еннен аз болса, онда ол бос орындармен толтырылып жазылады.

**%9i** – бүтін сан ені 9 цифрдан тұрады, сан ені аз болса, оның сол жағында бос орындар орналасады.

**%9.3f** – нақты сан ені 9 цифрдан тұрады, оның 3 таңбасы бөлшекке беріледі, сан ені аз болса, оның сол жағында бос орындар орналасады.



**Рахмет!**