

# Тақырыбы: C++ тілінде бағдарламалау

Құрастырушы: аға оқытушы Саринова А.Ж.  
С.Торайгырова атындағы ПМУ «ЭЖА» кафедрасы

Павлодар 2019

# C тілі туралы

- C тілінің бірнеше маңызды ерекшелігі бар, олар соны басқа программалау тілдерден бөледі.
- C тілі құрылымдық программалаудың толық конструкциялар жиынтығын, модульдігін, бағдарламаның блоктық құрылымын қолдайды.

# С тілдің кейбір ерекшеліктері.

- С тілінде кейбір төмен деңгейлі операциялар тізімі жүзеге асырылған. Осы операциялардың кейбіреулері машиналық бұйрықтарға тікеле сәйкес болады, мысалы, *бірлікті операциялар* немесе ++ и -- операциялары .
- *С мәліметтердің базалық типтері* бағдарламада Ассемблерде қолданылатын объектілерді бейнелейді – байттар, машиналық сөздер және т.б.
- С тілі айнымалыларға және функцияларға *бағыттауыш* механизмін қолдайды.
- **Бағыттауыш** – белгілі бір айнымалы немесе функцияның машиналық адресін сақтау үшін арналған айнымалы.
- Бағыттауыштардың арифметикасы қолданылады, ол Ассемблердегідей тікелей қолжетімділігін және жады адрестерімен жұмыс жасауды жүзеге асыруға мүмкіндік береді.
- Бағыттауыштарды пайдалану жоғарытиімді бағдарламаларды құрастыруға мүмкіндік береді, бірақ программалаушыдан ерекше абайлылықты талап етеді.

# Ескерту:

- Кейбір маңызды емес жағдаларды ескермей, C++ тілі C тілінің бір түрі болып табылады.
- C тілі ұсынатын мүмкіндіктерден басқа, C++ тілі жаңа типтерді анықтаудың икемді және тиімді құралдарын қамтамасыз етеді.

# Мынаны білу жөн:

- C++ тіліндегі басты ұғым – **класс**.
- Класстар ақпаратты сақтау, мәліметтердің кепілдендірілген инициализациясын, пайдаланушылар анықтайтын типтертің айқын емес түрлендірілуі, типтің динамикалық анықталуы, пайдаланушының жадыны басқаруды бақылау және операторлардың қайта жүктеу механизмін қамтамасыз етеді.

# Препроцессор

- **Препроцессор** – сіздің бағдарламаңыздың мәтінін компиляторға дейін өңдейтін бағдарлама.
- Препроцессордың жұмысы директивалармен басқарылады. Препроцессордың көмегімен келесі операцияларды орындауға болады:
- Бағдарламаға көрсетілген файлдардан мәтіндерді қосу;
- Идентификаторларды таңбалардың жүйелілікпен ауыстыру;
- Макроқойылым, яғни нақты аргументтерді ескеру арқылы қалыптасатын препроцессормен параметрленген мәтінмен белгілеуді ауыстыру;
- Бағдарламадан мәтіннің кейбір бөліктерін алып тастау (шартты компиляция).

# Файлдарды қосу

- Файлдарды қосу *#include* директивасы арқылы жүргізілуі мүмкін, оның келесі синтаксисі бар:
- **#include** <жол>
- **#include** «жол»
- Мұндағы бұрыштық жақшалар синтаксис элементі болып табылады.
- *#include* директивасы файлдың құрамын қосады, оған жол директивасы бар тармақтың орнына компиляцияланған файлға берілген.
- Егер жол бұрыштық жақшаларға алынған болса, онда файлды іздеу стандартты директорияларда жүзеге асырылады.
- Егер жол тырнақшаларға алынған болса және толығымен берілген болса, онда файлды іздеу берілген директорияда жүзеге асырылады, ал егер жол берілмеген болса – ағымдағы директорияда жүзеге асырылады.
- Осы директиваның көмегімен Сіз бағдарламаның мәтініне стандартты да, өзінің файлдарын да қосуға болады.

# Бағдарламаның параметрлері

- *main* функциясының, басқа бір функция сияқты параметрлері бар. Осы параметрлер бағдарламаға бұйрықтық (командалық) тармақтан беріледі.
- **void main(int argc, char \*argv[])**
- { ... }
- // *argc* мен *argv* атаулары тілдің талабы болып табылмайды
- Бірінші параметр массивте – екінші параметрде элементтер санын білдіреді, ол тармақтарға бағыттауыш массиві болып табылады.
- Әрбір тармақ бағдарламаға берілген бір параметрді сақтайды, сондай-ақ бірінші параметр (индексі 0) орындалатын файлдың атауын қосады және әрқашан бар болады.



# Алфавит

C тілінің таңбалар жиынының құрамында бар:

- Латын алфавитінің бас әріптері;
- Латын алфавитінің кіші әріптері;
- Араб сандары;
- бөлгіштер: , . ; : ? ! ' " | / \ ~ \_ ^ ( ) { } [ ] < > # % & - = + \*

C++ тілі бас және кіші әріптерді айырады,  
*name* және *Name* – әртүрлі идентификаторлар.

# Литералдар

- C++ тіліндегі литералдар бүтін, заттық, символдық және тармақты болуы мүмкін.
- Бүтін:
  - ондық: 10, 132, -32179;
  - сегіздік («0» символы алдында қойылады): 010, 0204, -076663;
  - оналтылық («0x» символдар алдында қойылады): 0xA, 0x84, 0x7db3.
- Заттық : 15.75, 1.575e1, .75, -.125
- Символдық: 'a', 'e', '!', '?', '2'.
- Тармақты: «тармақ».

# C++ тілінің мәліметтер типі

Атауы	Өлшемі	Ұсынылатын мәндер	Диапазон
bool	1 байт	логикалық	false, true
(signed) char	1 байт	Бүтін сан символдары	-128-ден 127-ге дейін
wchar_t	2 байта	Unicode символдары	0-ден 65535-ке дейін
(signed) short int	2 байта	Бүтін сандар	-32768-ден 32767-ге дейін
(signed) int	зависит от реализации (в последних компиляторах обычно 4 байта)	Бүтін сандар	
(signed) long int	4 байта	Бүтін сандар	-2147483648-ден 2147483647-ге дейін
(signed) long long int (signed) __int64 (MS)	8 байт	Бүтін сандар	-9,223,372,036,854,775,808-ден 9,223,372,036,854,775,807-г дейін
unsigned char	1 байт	Бүтін сандар символдары	0-ден 255-ке дейін
unsigned short int	2 байта	Бүтін сандар	0-ден 65535-ке дейін
unsigned int	Жүзеге асырудан тәуелді (соңғы компиляторларда 4 байта)	Бүтін сандар	

# C++ тілінің мәліметтер типі

Атауы	Өлшемі	Ұсынылатын мәндер	Диапазон
unsigned long int	4 байта	Бүтін сандар	0-ден 4294967295-ке дейін
(unsigned) long long int (unsigned) _int64 (MS)	8 байт	Бүтін сандар	0-ден 18,446,744,073,709,551,615 -ке дейін
float	4 байта	Заттық сандар	1.175494351e-38-ден 3.402823466e+38-ге дейін
double	8 байт	Заттық сандар	2.2250738585072014e-308 -ден 1.7976931348623158e+308 -ге дейін
long double	Жүзеге асырудан тәуелді	Заттық сандар	

● C++ тілінде *enum* – атап өтілетін бар, ол бүтін типтің жиыншасы болып табылады және *void* – бос тип, ол арнайы мәнге ие. Ол ешбір мәнді қайтармайтын функцияларды бейнелеуде және *void* типтің мәніне **бағыттауыштарды** бейнелеу үшін қолданылады. Осындай бағыттауыштар басқа бір типке бағыттауыштарға түрлендірілуі мүмкін.

● C++ тілінде **құрылымдарды** және **қосылыстарды** бейнелеуге болады.

● C++ тілінде символдар массиві болып табылатын массивтер мен тармақтар үшін арнайы типтері жоқ .

# Логикалық типтер

- С тілінде логикалық типтер болмаған.
- Логикалық мәндер бүтін типтің мәліметтері ретінде білдіріледі, сондай-ақ 0 мәні *жалған* логикалық мәнге мәйкес болды, ал **барлық басқа бүтін мәндер** *ақиқат* логикалық мәнге сәйкес болды. Анықтама бойынша *true* бүтін типке түрлендіру кезінде 1 деген мәнге ие, ал *false* – 0 деген мәнге ие.
- Сондай-ақ, керісінше бүтіндерді айқын емес түрде логикалық мәндерге түрлендіруге болады: нөлдік емес бүтін *true* түрлендіреді, а нөл – *false* түрлендіреді. Логикалық мәндер қажет ететін әрбір жерде бүтінсанды өрнек тұруы мүмкін.
- Арифметикалық және логикалық өрнектерде логикалық мәндер бүтінге түрленеді, операциялар түрлендірілген шамалардың үстінен орындалады.

## C++-тегі ең қарапайым бағдарлама мынадай болады:

```
#include <iostream>  
int main(void)  
{ std::cout << "Hello world";  
return 0; }
```

- Бағдарламаның бірінші тармағында **#include** директиваның көмегімен **iostream** басындағы (заголовочный) файлдың қосылуы орындалады.
- Басындағы (Заголовочные) файлдардың құрамында осы файл қосылғаннан кейін өзінің бағдарламаларында пайдалануға болатын функциялар мен басқа дайын элементтердің сипаттамалары бар. **iostream** C++-тің стандартты библиотекасына кіреді, бірақ C++, басындағы (Заголовочные) файлдарды өздігінен құруға болады да, оған жиі қолданылатын функцияларды, шаблондар мен басқа дайындамаларды орналастыруға болады.

● **iostream** басындағы (Заголовочные) файл ағындық енгізу мен шығару үшін дайын функциялар жиынтығын құрайды. Біз консольға шығару функциясын пайдаланамыз.

● Бағдарламаның екінші тармағында **main** функциясын бейнелейміз. Функцияның атауының алдында – оның қайтарылатын мәндерінің типі, бұл жағдайда – ол **int** (бүтін сандар). Атаудан кейін дөңгелек жақшамен функциялар параметрдің жиынтығы көрсетіледі, бұл жағдайда **void** сөзі параметрлердің жоқ болуын білдіреді. **Void** орнына дөңгелек жақшаларды бос қалтыруға болады.

● Функцияларды сипаттау, яғни оларда орындалатын бұйрықтар тізімі фигуралық жақшалармен белгіленеді (олар блок операторы деп аталады).



# Бағдарламаның құрылымы

- C++ тіліндегі бағдарлама препроцессордың директивасынан, компиляторға нұсқаулардан, ауыспалы және/немесе константаларды бейнелеуден және функцияларды анықтаудан тұрады.
- Айнымалыны бейнелеу айнымалының атауы мен атрибуттарын береді. Айнымалының атрибуттары ретінде **тип, элементтер саны (массивтер үшін), жады классының спецификациясы** және **инициализатор** бола алады .
- **Инициализатор** –айнымалыға, оның құрылуы кезінде мәнді беретін сәйкес константа.

# Константалар

- C++ тілінде мәнді тікелей өзгертуге болмайтынын көрсету үшін пайдаланушымен анықталатын константалар концепциясы енгізілген.
- Объектті константа деп жариялау үшін бейнелеуге *const* маңызды сөзді қосу керек. Константаға мәндерді қосуға рұқсаты болмағандықтан, ол инициализациялануы қажет.

# Функцияны бейнелеу және анықтау

- Функцияны бейнелеу функцияның атауын, қайтарылатын мәндер типін және параметрлердің саны мен типтерін береді, олар функцияларды шағырғанда болуы шарт.
- *void* сөзін қайтарымды мән ретінде белгілеу функцияның мәндерді қайтармайтынын білдіреді.
- Функцияның анықтауышы ретінде функцияны бейнелеу болып табылады, онда функцияның денесі бар. Функцияны анықтаудың мынадай синтаксисі бар:
  - *<типi> <атауы> (< формальді параметрлер тізімі>)*
  - {
  - *[<бейнелеу>]*
  - *[<операторлар>]*
  - }
  -

# Функцияларды анықтау

## мысалдары

- Функцияның шақырылуы кезінде оның формальді параметрлері үшін жады бөлінеді және әрбір формальді параметрге сәйкес нақты параметрдің мәні беріледі.
- Параметрлерді беру семантикасы инициализациялау семантикасына тең. Формальды және нақты параметрлердің типтерінің сәйкестігі тексеріледі және қажет болғанда стандартты немесе пайдаланушы анықтаған типтердің түрлендірулері орындалады.
- *Массивтерді параметрлер ретінде және параметрлерді беру құралы ретінде және үнсіз келісім бойынша параметрлерді берудің* арнайы ережелері бар, олардың сәйкестігі тексерілмейді.