

Дәріс 1. C++ тіліне кіріспе

1. Кіріспе

C++ — бұл жоғары деңгейлі, көп парадигмалы (объектілі-бағдарланған, процедуралық, және тіпті функционалды) бағдарламалау тілі. Ол тиімділігі мен икемділігі арқасында қазіргі таңда кеңінен қолданылатын әмбебап тілдердің бірі. C++ тілі жүйелік бағдарламалардан бастап, қолданбалы және ойын индустриясына дейінгі көптеген салаларда қолданылады.

2. C++ тілінің шығу тарихы

C++ тілінің тарихы 1979 жылы Дания инженері **Бьёрн Страуструп** (Bjarne Stroustrup) Bell Laboratories (AT&T) зертханасында жұмыс істеп жүрген кезінде басталды. Ол кезде негізгі тіл ретінде C кеңінен қолданылатын, бірақ объектіге бағытталған мүмкіндіктердің жоқтығы күрделі жобаларды дамытуда қиындық туғызды.

2.1. Алғашқы даму кезеңдері

- **1979–1983 жж.** — Страуструп "**C with Classes**" («Кластармен бірге C») деп аталатын жаңа тіл құра бастады. Ол C тіліне кластар, мұрагерлік, конструкторлар мен деструкторлар сияқты объектілі-бағдарланған элементтерді қосты.
- **1983 ж.** — Тілдің атауы C++ болып өзгертілді. Мұндағы “++” C тіліндегі инкремент операциясы (бірлікке арттыру) символы, яғни “*C тіліне қосылған жаңа мүмкіндіктер*” дегенді білдіреді.
- **1985 ж.** — Бірінші ресми компилятор **Cfront** және алғашқы кітап — “*The C++ Programming Language*” жарық көрді.
- **1990 ж.** — Тіл кеңінен тарала бастады және ISO стандарттау комитетінің назарын аударды.

2.2. C++ стандарттарының эволюциясы

Жыл	Стандарт атауы	Ерекшеліктері
1998	C++98	Бірінші ресми ISO стандарты. STL (Standard Template Library) енгізілді.
2003	C++03	C++98-дегі қателер мен кемшіліктер түзетілді.
2011	C++11	Үлкен жаңарту: auto, lambda-функциялар, smart pointers, move semantics.
2014	C++14	Кішкене жетілдірулер: generic lambdas, constexpr кеңейтілді.

Жыл	Стандарт атауы	Ерекшеліктері
2017	C++17	if constexpr, structured bindings, filesystem кітапханасы.
2020	C++20	Concepts, ranges, modules, coroutines енгізілді.
2023	C++23	UTF-8 қолдауы, жаңа стандарт кітапханалар және compile-time есептеулер жетілдірілді.

3. C++ тілінің негізгі ерекшеліктері мен мүмкіндіктері

3.1. Негізгі сипаттары

- **Жоғары өнімділік** — C++ төмен деңгейлі ресурстармен (жад, процессор) тікелей жұмыс істей алады.
- **Объектілі-бағдарланған тәсіл** — класстар, мұрагерлік, инкапсуляция және полиморфизм ұғымдарын қамтиды.
- **Жалпы типтермен жұмыс (Templates)** — кодты қайта пайдалануға және түрлі типтерге бейімдеуге мүмкіндік береді.
- **Төмен деңгейлі операциялар** — көрсеткіштер (pointers), жадыны тікелей басқару, биттік операциялар.
- **Жоғары тасымалданғыштық (portability)** — C++ коды әртүрлі операциялық жүйелерде аз өзгеріспен орындала алады.
- **Көп парадигмалы тіл** — C++ объектілі, процедуралық және функционалды тәсілдерді біріктіреді.

3.2. Негізгі артықшылықтары

- Тиімділік пен жылдамдық
- Жүйелік ресурстарды толық бақылау
- Үлкен жобаларға бейімделгіш
- Үлкен стандартты кітапхана (STL)
- Артқа үйлесімділік (C тілінің кодтарын қолдануға болады)

4. C++ тілінің қолдану салалары

Сала	Мысалдар
Жүйелік бағдарламалау	Операциялық жүйелер (Windows NT, macOS компоненттері, Linux ядросының кей бөліктері), драйверлер
Ойын индустриясы	Unreal Engine, Unity (C++ backend), CryEngine

Сала	Мысалдар
Ғылыми есептеулер	MATLAB ядросы, физикалық модельдеу, деректерді өңдеу
Қаржы жүйелері	Жоғары жылдамдықты сауда алгоритмдері, банктік серверлер
Embedded жүйелер	Микроконтроллерлер, IoT құрылғылары
Қолданбалы бағдарламалар	Adobe Photoshop, AutoCAD, Microsoft Office кей компоненттері
Машиналық оқыту және AI	TensorFlow және PyTorch кітапханаларының негізгі бөлігі C++ тілінде жазылған

5. C++ бағдарламасының құрылымы

C++ бағдарламасы бір немесе бірнеше **файлдардан** тұрады. Негізгі орындау нүктесі – `main()` функциясы.

5.1. Ең қарапайым бағдарлама мысалы

```
#include <iostream>      // енгізу-шығару кітапханасы
using namespace std;     // стандартты атау кеңістігін қолдану

int main() {
    cout << "Сәлем, әлем!" << endl; // Экранға шығару
    return 0; // Бағдарламаның сәтті аяқталуын білдіреді
}
```

5.2. Бағдарламаның негізгі бөліктері

- Препроцессор директивасы (`#include`)**
— Компиляция алдында қажетті кітапханаларды қосады.
Мысалы, `<iostream>` — енгізу және шығару ағындарын қолдануға арналған.
- Атау кеңістігі (`namespace`)**
— Кодты логикалық түрде бөлуге және атау қайшылығын болдырмауға мүмкіндік береді.
Мысалы, `std` — стандартты кітапхана кеңістігі.
- Негізгі функция (`main`)**
— Бағдарламаның орындалуы осы жерден басталады.
`return 0;` — бағдарламаның сәтті аяқталғанын білдіреді.
- Операторлар мен нұсқаулар**
— Негізгі әрекеттерді орындайтын кодтар.

5.3. Енгізу және шығару

C++ тілінде деректермен жұмыс істеу үшін:

- **cout** — экранға шығару үшін;
- **cin** — пернетақтадан енгізу үшін пайдаланылады.

Мысал:

```
#include <iostream>
using namespace std;

int main() {
    int age;
    cout << "Жасыңызды енгізіңіз: ";
    cin >> age;
    cout << "Сіздің жасыңыз: " << age << endl;
    return 0;
}
```

6. C++ бағдарламасын компиляциялау және орындау

6.1. Компиляция кезеңдері

1. **Препроцессорлық өңдеу** — `#include`, `#define` және т.б. өңделеді.
2. **Компиляция** — код машиналық тілге жақын *объектілік кодқа* айналады.
3. **Линковка (linking)** — барлық модульдер біріктіріліп, орындалатын `.exe` немесе `.out` файл жасалады.

6.2. Даму орталарын пайдалану

C++ тілінде код жазу үшін келесі IDE-лер жиі қолданылады:

- **Visual Studio** (Windows)
 - **CLion** (кроссплатформалы)
 - **Code::Blocks**, **Dev-C++**
 - **Qt Creator** (GUI жобалар үшін)
 - **VS Code** (жеңіл нұсқа, плагиндер арқылы)
-

7. Қорытынды

C++ — уақыт сынынан өткен, қуатты және икемді бағдарламалау тілі. Оның басты ерекшелігі — **жоғары өнімділік пен объектілі тәсілді біріктіру**. Тілдің даму тарихы мен стандарттарының жетілдірілуі оны әлі күнге дейін өзекті етіп ұстап тұр. Келесі дәрістерде C++ тілінің синтаксисі, айнымалылар, типтер, шартты операторлар және циклдер егжей-тегжейлі қарастырылады.

Негізгі ұғымдар мен терминдер

Термин	Түсініктеме
Компилятор	Бағдарламаны машиналық кодқа айналдыратын құрал
Класс (Class)	Объектіні сипаттайтын үлгі (шаблон)
Объект	Кластың нақты данасы
STL	Standard Template Library — стандартты шаблон кітапханасы
Namespace	Код кеңістігін логикалық бөлу әдісі
Header файл	Функциялар мен кластардың жарияланымдары сақталған файл

Бақылау сұрақтары

1. C++ тілін кім және қашан ойлап тапқан?
2. “C with Classes” дегеніміз не?
3. C++ тілінің қандай стандарттары бар және олардың ерекшелігі неде?
4. C++ тілінің басты артықшылықтары қандай?
5. C++ тілінде `#include` және `namespace` не үшін қажет?
6. C++ тілінің қолдану салаларына мысал келтіріңіз.
7. `main()` функциясының рөлі қандай?
8. C++ бағдарламасын компиляциялау кезеңдерін атаңыз.