Дәріс 7. Массивтер және жолдар Бірөлшемді және көпөлшемді массивтер, C-style жолдарымен және std::string класымен жұмыс.

Массивтер және жолдар

1. Кіріспе

Бағдарламалау тілдерінде **массив (array)** — бір типтегі бірнеше деректерді бір атаумен сақтауға арналған құрылым.
Массивтер деректерді жүйелеуге, іздеуге, сұрыптауға және өңдеуге ыңғайлы.

Мысалы:
Сыныптағы 30 студенттің бағасын сақтау үшін 30 жеке айнымалы жазудың орнына, бір массив қолдануға болады:

```
int grades[30];
```

Ал **жол (string)** — бұл таңбалар тізбегі. C++ тілінде жолдарды екі тәсілмен көрсетуге болады:

1. **C-style жолдары** — символдар массиві (`char` типі);
2. **C++ std::string класы** — жолдармен ыңғайлы жұмыс істеуге арналған стандартты класс.

---

2. Бірөлшемді массивтер

Жариялау синтаксисі:

```
тип массив_атауы[элемент_саны];
```

Мысал:

```
int numbers[5]; // 5 элементтен тұратын массив
```

Мұнда `numbers` массиві 5 бүтін сан сақтай алады: `numbers[0]`, `numbers[1]`, ..., `numbers[4]`.

Назар аударыңыз: массив индекстері **0-ден басталады**.

Массивті инициализациялау:

```
int arr[5] = {1, 2, 3, 4, 5};
```

Немесе ішінара:

```
int arr[5] = {10, 20}; // қалған элементтер 0-ге тең
```

Кейде өлшемді көрсетпей-ақ жазуға болады:

```
int arr[] = {2, 4, 6, 8, 10};
```

Массив элементтеріне қол жеткізу:

```
arr[0] = 15;      // бірінші элементке мән беру
cout << arr[2];  // үшінші элементті шығару
```

Мысал: массив элементтерінің қосындысын табу

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

int main() {
    int numbers[5] = {1, 2, 3, 4, 5};
    int sum = 0;

    for (int i = 0; i < 5; i++) {
        sum += numbers[i];
    }

    cout << "Қосынды: " << sum;
}
```

---

3. Көпөлшемді массивтер

Көпөлшемді массив — массивтердің массиві.

Жариялау:

```
тип атау[жол_саны][баған_саны];
```

Мысал:

```
int matrix[3][3];
```

Инициализация:

```
int matrix[2][3] = {
    {1, 2, 3},
    {4, 5, 6}
};
```

Элементке қол жеткізу:

```
cout << matrix[1][2]; // 5-ке тең
```

Мысал: екі өлшемді массивті шығару

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

int main() {
    int a[2][3] = {{1,2,3}, {4,5,6}};

    for(int i=0; i<2; i++){
        for(int j=0; j<3; j++){
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
}
```

---

4. Массивтермен жұмыс істеу ерекшеліктері

- Массивтің **мөлшері** компиляция кезінде белгілі болуы тиіс;
- Массивтің аты (`arr`) — оның бірінші элементінің адресін көрсетеді (указатель);
- Массив шегінен шығу (`arr[5]` егер өлшем 5 болса) — **қате** (Undefined Behavior);
- Массивтерді функцияларға **сілтеме (reference)** немесе **указатель арқылы** жіберуге болады.

Функцияға массив жіберу мысалы:

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

void printArray(int arr[], int size) {
    for(int i=0; i<size; i++)
        cout << arr[i] << " ";
}

int main() {
    int nums[5] = {1,2,3,4,5};
    printArray(nums, 5);
}
```

---

5. C-style жолдары (char массивтері)

С тілінен қалған тәсіл — жолды символдар массиві ретінде сақтау.
Мұндай жолдың соңында міндетті түрде **'\0' (нөлдік символ)** болуы керек.

Мысал:

```
char str[6] = {'H','e','l','l','o','\0'};
```

немесе қысқаша:

```
char str[] = "Hello";
```

C-style жолды шығару:

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

int main() {
    char name[] = "Bauyrzhan";
    cout << name;
}
```

C-string кітапханалық функциялары (<cstring>):

Функция	Мақсаты
`strlen(str)`	Жол ұзындығын табу
`strcpy(dest, src)`	Бір жолды екіншісіне көшіру
`strcat(dest, src)`	Жолдарды біріктіру
`strcmp(str1, str2)`	Екі жолды салыстыру
`strchr(str, ch)`	Символды іздеу
`strstr(str, substr)`	Ішкі жолды іздеу

Мысал:

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

int main() {
    char s1[20] = "Hello";
    char s2[] = "World";

    strcat(s1, s2); // s1 = "HelloWorld"
    cout << s1 << endl;
    cout << "Ұзындығы: " << strlen(s1);
}
```

---

6. std::string класы

C++ тілінде жолдармен ыңғайлы жұмыс істеу үшін `std::string` класы енгізілген (`<string>` кітапханасында).

Жариялау және инициализация:

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

int main() {
    string name = "Aidar";
    cout << name;
}
```

Негізгі операциялар:

Операция	Мысал	Түсіндірме
Қосу	`s1 + s2`	Екі жолды біріктіреді
Көшіру	`s2 = s1;`	Бір жолды екіншісіне көшіру
Ұзындығы	`s.size()` немесе `s.length()`	Жол ұзындығын алу
Элементке қол жеткізу	`s[i]`	і-ші символды алу
Іздеу	`s.find("word")`	Ішкі жолды іздеу
Жою	`s.erase(pos, len)`	Белгілі бөлікті жою
Қосу	`s.insert(pos, "text")`	Текст қосу
Салыстыру	`s1 == s2`, `s1 != s2`	Салыстыру

Мысал:

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

int main() {
    string str1 = "Hello";
    string str2 = "World";
    string str3 = str1 + " " + str2;

    cout << str3 << endl;
    cout << "Ұзындығы: " << str3.size() << endl;
    cout << "2-ші символ: " << str3[1] << endl;
}
```

Жолды енгізу (input):

```
string name;
cout << "Атыңызды енгізіңіз: ";
getline(cin, name); // бос орынмен қоса енгізу
```

```
cout << "Сәлем, " << name << "!";
```

C-style және std::string арасында түрлендіру:

```
string s = "Hello";
const char* cstr = s.c_str(); // string → C-string

char c[] = "World";
string s2 = c; // C-string → string
```

---

7. Массивтер мен жолдарды салыстыру

Ерекшелік	C-style	std::string
Мән типі	`char[]`	`std::string`
Кітапхана	`<cstring>`	`<string>`
Ұзындығы	`strlen(str)`	`str.size()`
Біріктіру	`strcat(s1,s2)`	`s1 + s2`
Салыстыру	`strcmp(s1,s2)`	`s1 == s2`
Қауіпсіздік	Төмен	Жоғары
Икемділік	Аз	Көп

Қорытынды: Қазіргі C++ тілінде **`std::string`** класы C-style жолдарынан әлдеқайда қауіпсіз және ыңғайлы.

---

8. Массивтер мен жолдар бойынша мысалдар

Мысал 1: Массивтің ең үлкен элементін табу

```
int main() {
    int arr[5] = {10, 4, 7, 25, 3};
    int max = arr[0];
    for (int i = 1; i < 5; i++)
        if (arr[i] > max) max = arr[i];
    cout << "Максимум: " << max;
}
```

Мысал 2: Студенттердің есімдерін шығару

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

```
int main() {
    string students[3] = {"Aidar", "Bota", "Diana"};

    for (int i = 0; i < 3; i++)
        cout << i+1 << ". " << students[i] << endl;
}
```

Мысал 3: Жолдағы дауысты әріптер санын есептеу

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

int main() {
    string text;
    cout << "Мәтінді енгізіңіз: ";
    getline(cin, text);

    int count = 0;
    for (char c : text)
        if (c=='a'||c=='e'||c=='i'||c=='o'||c=='u'||
            c=='A'||c=='E'||c=='I'||c=='O'||c=='U')
            count++;

    cout << "Дауысты әріптер саны: " << count;
}
```

---

9. Қорытынды

- Массивтер — бір типтегі деректердің реттелген жиыны.
- Көпөлшемді массивтер кестелер мен матрицаларды сақтауға ыңғайлы.
- C-style жолдары ескі тәсіл, бірақ әлі қолданылады.
- `std::string` класы жолдармен заманауи және қауіпсіз жұмыс тәсілін ұсынады.

---

10. Өзіндік бақылау сұрақтары

1. Массив дегеніміз не және ол не үшін қолданылады?
2. Бірөлшемді және көпөлшемді массивтің айырмашылығы неде?
3. Массив элементтеріне қалай қол жеткіземіз?
4. Массивті функцияға қалай жіберуге болады?
5. C-style жолдары қалай сақталады?
6. C-string және std::string арасындағы айырмашылық қандай?
7. `strlen()` және `size()` функцияларының айырмашылығы неде?
8. `getline()` не үшін қолданылады?
9. Массив шегінен шығудың қауіптілігі неде?
10. std::string класындағы негізгі әдістерді атаңыз.