# ДӘРІС 3. Деректер типтері және айнымалылар (Data Types and Variables)

---

## 1. Кіріспе

C++ бағдарламасында деректер — ақпаратты сақтау мен өңдеудің негізгі бірлігі. Барлық бағдарламалар деректерді **сақтайды**, **есептейді**, **салыстырады** және **шығарады**.
Осы деректермен жұмыс істеу үшін тілде **деректер типтері (data types)** және оларды сақтайтын **айнымалылар (variables)** қолданылады.

---

## 2. Айнымалы (Variable) ұғымы

### *2.1. Анықтама*

**Айнымалы (variable)** — бұл жадыдағы белгілі бір ұяшыққа берілген атау, ол белгілі бір **мәнді (value)** сақтайды және **типімен (type)** сипатталады.

C++ тілінде әр айнымалы **типпен** міндетті түрде жарияланады. Бұл компиляторға жады көлемін және орындалу кезінде қолданылатын операцияларды анықтауға мүмкіндік береді.

### *2.2. Айнымалыны жариялау (declaration)*

Жалпы синтаксис:

```
<деректер_типі> <айнымалы_атауы>;
```

**Мысалдар:**

```
int age;          // бүтін сан типті айнымалы
double salary;    // нақты сан типті айнымалы
char grade;       // бір символ
```

### *2.3. Инициализация (мән беру)*

Айнымалыға бастапқы мән бірден берілуі мүмкін:

```
int age = 20;
double pi = 3.14159;
char letter = 'A';
```

### *2.4. Бірнеше айнымалыны бір жолда жариялау*

```
int x = 1, y = 2, z = 3;
```

---

## 3. C++ тіліндегі деректер типтері

C++ тілінде деректер бірнеше **негізгі категорияларға** бөлінеді:

| Категория | Түрлер |
|---|---|
| **Біріншілік (primitive)** | `int`, `char`, `float`, `double`, `bool` |
| **Құрама (compound)** | массивтер (arrays), көрсеткіштер (pointers), сілтемелер (references), құрылымдар (struct), класстар |
| **Туынды (derived)** | `enum`, `typedef`, `auto`, шаблондық типтер (templates) |
| **Біріккен (user-defined)** | қолданушы анықтайтын кластар мен типтер |

---

## 4. Негізгі деректер типтері және олардың өлшемдері

| Тип | Мән диапазоны (32-bit жүйе) | Өлшемі (байт) | Сипаттамасы |
|---|---|---|---|
| `bool` | `true` немесе `false` | 1 | Логикалық мән |
| `char` | -128 … +127 | 1 | Бір символ |
| `unsigned char` | 0 … 255 | 1 | Теріс емес символ |
| `short int` | -32,768 … +32,767 | 2 | Қысқа бүтін сан |
| `unsigned short int` | 0 … 65,535 | 2 | Теріс емес қысқа сан |
| `int` | -2,147,483,648 … +2,147,483,647 | 4 | Негізгі бүтін сан |
| `unsigned int` | 0 … 4,294,967,295 | 4 | Теріс емес бүтін сан |
| `long long` | шамамен $\pm 9 \times 10^{18}$ | 8 | Өте үлкен бүтін сан |
| `float` | $\pm 3.4 \times 10^{38}$ (6–7 ондық дәлдік) | 4 | Нақты сан (бір дәлдік) |
| `double` | $\pm 1.7 \times 10^{308}$ (15 ондық дәлдік) | 8 | Нақты сан (екі дәлдік) |

| Тип | Мән диапазоны (32-bit жүйе) | Өлшемі (байт) | Сипаттамасы |
|---|---|---|---|
| `long double` | одан да жоғары дәлдік | 10/16 | Нақты сан (кеңейтілген) |

⚙ Деректердің нақты өлшемдері компилятор мен платформаға байланысты өзгеруі мүмкін.

---

## 5. signed және unsigned модификаторлары

C++ тілінде бүтін типтерді екі түрлі белгілеуге болады:

- **signed** — теріс және оң мәндер сақталады;
- **unsigned** — тек оң мәндер сақталады, диапазон екі есе артады.

**Мысал:**

```
signed int a = -10;
unsigned int b = 10; // теріс мән бере алмаймыз
```

**Айырмашылық:**

| Модификатор | Диапазон |
|---|---|
| `signed int` | -2,147,483,648 … +2,147,483,647 |
| `unsigned int` | 0 … 4,294,967,295 |

---

## 6. Символдық тип (char)

`char` — бір символды (әріпті, санды немесе таңбаны) сақтауға арналған.
Символдар **қос тырнақшада** емес, **жалғыз тырнақшада** жазылады.

```
char c = 'A';
char digit = '5';
```

C++ тілінде әрбір символ **ASCII коды** түрінде сақталады.

**Мысал:**

```
char letter = 'A';
cout << int(letter); // нәтижесі: 65
```

---

## 7. Нақты типтер (float, double)

Нақты типтер бөлшек сандарды сақтау үшін қолданылады.

```
float height = 1.75f;
double pi = 3.141592653589793;
```

Нақты типтерде шамалы қателік болуы мүмкін, себебі олар **жүзбелі нүктемен (floating point)** сақталады.

**Мысал:**

```
cout << (0.1 + 0.2); // Нәтиже: 0.30000000000000004
```

---

## 8. Логикалық тип (bool)

`bool` типі тек екі мән қабылдайды:

- `true` (ақиқат)
- `false` (жалған)

**Мысал:**

```
bool isOpen = true;
bool isFinished = false;
```

Шығару кезінде:
`true` → 1
`false` → 0

---

## 9. Арнайы типтер

### *9.1. void типі*

- Мән қайтармайтын функциялар үшін қолданылады.
- Мысалы:

```
void sayHello() {
    cout << "Сәлем!" << endl;
}
```

### *9.2. auto типі*

- C++11 стандартынан бастап айнымалы типін автоматты түрде анықтайды.

```
auto x = 10;        // int
auto y = 3.14;      // double
auto z = "text";    // const char*
```

---

## 10. Константалар (тұрақтылар)

**Константа (constant)** — мәні өзгермейтін айнымалы.
C++ тілінде тұрақтыларды екі жолмен жариялауға болады:

### *1. `const` кілт сөзімен*

```
const double PI = 3.14159;
const int MAX = 100;
```

### *2. Препроцессор арқылы*

```
#define PI 3.14159
#define MAX 100
```

`const` — типтік қауіпсіз нұсқа, себебі ол компилятор деңгейінде бақыланады.

---

## 11. Айнымалылардың атаулары мен ережелері

Айнымалы атауы бағдарламаның кез келген бөлігіне түсінікті болуы тиіс және келесі ережелерге бағынады:

### *11.1. Атау ережелері*

✅ Рұқсат етіледі:

- Латын әріптері, цифрлар және _ (астын сызу);
- Алғашқы символ — әріп немесе _ болуы керек.

❌ Рұқсат етілмейді:

- Санмен бастау (`2count` ❌);
- Бос орын (`user name` ❌);
- Арнайы символдар (`%`, `$`, `#` ❌);
- Кілт сөздерді пайдалану (`int`, `while`, `class` ❌).

**Мысал:**

```
int age;          // дұрыс
int 2age;         // қате
int my-age;       // қате
```

### *11.2. Атау жазу стильдері*

| Стиль | Мысал | Қолдану саласы |
|---|---|---|
| **camelCase** | userAge, totalSum | айнымалылар мен функция атаулары |
| **PascalCase** | UserName, StudentData | класс атаулары |
| **snake_case** | user_age, max_value | жиі Linux жобаларында |

## 12. Айнымалының өмір сүру аймағы (Scope)

Айнымалының **өмір сүру аймағы (scope)** — оның код ішінде көрінетін және қолданылатын бөлігі.

### *12.1. Жергілікті айнымалылар (local)*

Функция ішінде жарияланады және тек сол функцияда ғана қолданылады.

```
void func() {
    int x = 10; // тек осы жерде қолданылады
}
```

### *12.2. Глобалды айнымалылар (global)*

Функциялардың сыртында жарияланады және барлық жерден қол жетімді.

```
int count = 0;

void func() {
    count++; // глобалды айнымалыға қатынау
}
```

### *12.3. Көріну деңгейлері*

| Область | Анықтама |
|---|---|
| **Local scope** | Функция ішіндегі айнымалы |
| **Global scope** | Барлық бағдарламаға ортақ |
| **Block scope** | `{ }` ішінде жарияланған айнымалы |
| **Namespace scope** | Белгілі бір атау кеңістігіне тиесілі айнымалы |

---

## 13. Айнымалылардың жадыда орналасуы

Әр айнымалы **жадтың белгілі бір аймағында** орналасады.
Оны тексеру үшін **& (address-of)** операторын қолданамыз:

```
int a = 10;
cout << &a; // айнымалының жадыдағы мекенжайы
```

**Шығыс мысалы:**

```
0x7ffeea1234
```

---

## 14. Мысалдар

### *Мысал 1. Айнымалы түрлері*

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

int main() {
    int age = 25;
    double salary = 1250.50;
    char grade = 'A';
    bool isStudent = true;

    cout << "Жасы: " << age << endl;
    cout << "Жалақысы: " << salary << endl;
    cout << "Бағасы: " << grade << endl;
    cout << "Студент пе? " << isStudent << endl;
    return 0;
}
```

### *Мысал 2. Константалар*

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

int main() {
    const double PI = 3.14159;
    double r = 5.0;
    double area = PI * r * r;
    cout << "Шеңбер ауданы: " << area << endl;
}
```

---

## 15. Қорытынды

C++ тілінде айнымалылар — бағдарламаның деректерді сақтау мен өңдеу өзегі. Оларды дұрыс типте жариялау — **жадты үнемдеу**, **жылдамдықты арттыру** және **қателіктерді азайту** тәсілі.
Деректер типтерін, олардың диапазондарын және айнымалылардың көріну аймағын білу — кәсіби C++ бағдарламалаушы болудың алғашқы қадамы.

---

## 16. Бақылау сұрақтары

1. Айнымалы дегеніміз не және ол не үшін қажет?
2. C++ тілінде қандай негізгі деректер типтері бар?
3. `signed` және `unsigned` айырмашылығы неде?
4. `float` пен `double` типтерінің айырмашылығы қандай?
5. Константа қалай жарияланады?
6. Айнымалының “scope” ұғымын түсіндіріңіз.
7. `auto` кілт сөзі не үшін қолданылады?
8. Айнымалының жадтағы мекенжайын қалай алуға болады?