Дәріс 5. Басқару конструкциялары if/else, switch шартты операторлары. for, while, do…while циклдері. break және continue операторлары.

Басқару конструкциялары (Control Structures)

---

1. Кіріспе

Бағдарламалау тілдерінде басқару конструкциялары (немесе басқару операторлары) — бұл бағдарламаның орындалу барысын, яғни **қатарлардың орындалу ретін** өзгертуге арналған құралдар.
C++ тілінде басқару конструкциялары үш негізгі түрге бөлінеді:

1. **Тізбектей орындалу (Sequential execution)** – нұсқаулар бірінен соң бірі орындалады.
2. **Тармақталу (Selection)** – шартқа байланысты әртүрлі блок орындалады (`if`, `if…else`, `switch`).
3. **Қайталану (Iteration / Loop)** – бір блок бірнеше рет қайталанып орындалады (`for`, `while`, `do…while`).

Сонымен қатар, **`break`**, **`continue`**, және **`return`** операторлары циклдер мен шарттардың орындалу тәртібін басқаруға мүмкіндік береді.

---

2. Тармақталу операторлары

2.1. if операторы

`if` операторы берілген шарттың ақиқат (true) не жалған (false) болуына байланысты белгілі бір блокты орындайды.

Жалпы жазылуы:

```
if (шарт) {
    // шарт ақиқат болса орындалатын блок
}
```

Мысал:

```
int a = 10;
if (a > 5) {
    cout << "a 5-тен үлкен";
}
```

2.2. if...else операторы

`if…else` конструкциясы екі нұсқаның бірін орындайды.

Жалпы жазылуы:

```
if (шарт) {
    // шарт ақиқат болса орындалады
} else {
    // шарт жалған болса орындалады
}
```

Мысал:

```
int x = -3;
if (x >= 0) {
    cout << "Оң сан";
} else {
    cout << "Теріс сан";
}
```

2.3. if...else if...else көп тармақты нұсқасы

Бірнеше шартты тізбектеп тексеру үшін қолданылады.

Мысал:

```
int grade = 85;
if (grade >= 90)
    cout << "A";
else if (grade >= 75)
    cout << "B";
else if (grade >= 60)
    cout << "C";
else
    cout << "F";
```

Ескерту: `else if` тармақтары ретімен тексеріледі, және бірінші ақиқат шарт табылғанда, қалғандары орындалмайды.

---

3. switch операторы

`switch` операторы көп тармақты таңдаулар үшін ыңғайлы. Ол **бүтін (int, char, enum)** типті айнымалы мәніне байланысты түрлі блокты орындайды.

Жалпы жазылуы:

```
switch (өрнек) {
    case мән1:
        // орындалатын код
        break;
    case мән2:
        // орындалатын код
        break;
    default:
        // ешбір шарт сәйкес келмесе орындалады
}
```

Мысал:

```
int day = 3;
switch (day) {
    case 1: cout << "Дүйсенбі"; break;
    case 2: cout << "Сейсенбі"; break;
    case 3: cout << "Сәрсенбі"; break;
    default: cout << "Белгісіз күн";
}
```

Маңызды:

- Әр `case` соңында `break` операторын қою қажет.
- Егер `break` жазылмаса, келесі `case` блоктары да орындала береді («**fall-through**» эффектісі).

---

4. Циклдер (Loops)

Циклдер белгілі бір блокты **бірнеше рет қайталап орындау** үшін қолданылады.
C++ тілінде үш негізгі цикл бар:

1. `for` – қайталау саны белгілі болғанда қолданылады.
2. `while` – қайталау шарты алдын ала белгілі болғанда қолданылады.
3. `do...while` – блок кем дегенде бір рет орындалуы қажет болғанда қолданылады.

4.1. for циклі

Жалпы жазылуы:

```
for (инициализация; шарт; өзгеріс) {
    // орындалатын блок
}
```

Мысал:

```
for (int i = 0; i < 5; i++) {
    cout << i << " ";
}
```

Нәтиже:
`0 1 2 3 4`

Түсіндірме:

- `int i = 0` — циклдің басында айнымалыны инициализациялау
- `i < 5` — жалғастыру шарты
- `i++` — әр итерациядан кейін орындалатын операция

4.2. while циклі

Жалпы жазылуы:

```
while (шарт) {
    // шарт ақиқат болғанша орындалады
}
```

Мысал:

```
int i = 0;
while (i < 5) {
    cout << i << " ";
    i++;
}
```

Нәтиже:
```
0 1 2 3 4
```

Егер шарт ешқашан жалған болмаса, цикл **шексіз** орындалып қалады.

4.3. do...while циклі

Бұл циклде блок **кем дегенде бір рет орындалады**, себебі шарт соңында тексеріледі.

Жалпы жазылуы:

```
do {
    // орындалатын код
} while (шарт);
```

Мысал:

```
int x = 0;
do {
    cout << "x = " << x << endl;
    x++;
} while (x < 3);
```

Нәтиже:

```
x = 0
x = 1
x = 2
```

do…while циклі **меню түріндегі бағдарламаларда** жиі қолданылады (мысалы, пайдаланушы «шығу» нұсқасын таңдағанша қайталану).

---

5. Циклдерді басқару операторлары

5.1. break операторы

Циклдің орындалуын **толық тоқтатады** және циклден шығады.

Мысал:

```
for (int i = 1; i <= 10; i++) {
    if (i == 5) break;
    cout << i << " ";
}
```

Нәтиже:
```
1 2 3 4
```

5.2. continue операторы

Циклдің ағымдағы итерациясын өткізіп, келесіге өтеді.

Мысал:

```
for (int i = 1; i <= 5; i++) {
    if (i == 3) continue;
    cout << i << " ";
}
```

Нәтиже:
```
1 2 4 5
```

`i == 3` болғанда `continue` орындалады және `cout` жолы аттап кетіледі.

---

6. Ішкі (құрамдас) циклдер (Nested loops)

Циклдердің бірін екіншісінің ішінде қолдануға болады.
Мысалы, көбейту кестесін шығару:

```
for (int i = 1; i <= 3; i++) {
    for (int j = 1; j <= 3; j++) {
        cout << i * j << "\t";
    }
    cout << endl;
}
```

Нәтиже:

```
1       2       3
2       4       6
3       6       9
```

7. Шартты және циклдік операторларды қолдану мысалдары

Мысал 1. Ең үлкен санды табу

```
int a, b;
cout << "Екі сан енгізіңіз: ";
cin >> a >> b;

if (a > b)
    cout << "Ең үлкені: " << a;
else
    cout << "Ең үлкені: " << b;
```

Мысал 2. 1-ден 100-ге дейінгі жұп сандарды шығару

```
for (int i = 1; i <= 100; i++) {
    if (i % 2 != 0)
        continue; // тақ сандарды өткізіп кетеміз
    cout << i << " ";
}
```

Мысал 3. do...while арқылы мәзір

```
int choice;
do {
    cout << "\n1. Қосу\n2. Азайту\n3. Шығу\nТаңдаңыз: ";
    cin >> choice;

    switch (choice) {
        case 1: cout << "Қосу операциясы\n"; break;
        case 2: cout << "Азайту операциясы\n"; break;
        case 3: cout << "Бағдарлама аяқталды\n"; break;
        default: cout << "Қате таңдау!\n";
    }
} while (choice != 3);
```

---

8. Қателер мен сақтық шаралары

- Цикл ішінде айнымалы мәнін жаңартуды ұмытпау (әйтпесе шексіз цикл).
- `switch` операторында әр `case` соңына `break` қою.
- Логикалық шарттардың дұрыс жазылғанын тексеру.
- Бірдей атаулы айнымалыларды әртүрлі блоктарда абайлап қолдану (scope ережелері).

---

9. Қорытынды

Басқару конструкциялары — бағдарламаның логикасын құратын негізгі элементтер. Олардың көмегімен:

- шартқа байланысты әрекеттер таңдалады (`if`, `switch`),
- әрекеттер бірнеше рет орындалады (`for`, `while`, `do…while`),
- циклдер мен шарттардың орындалу реті басқарылады (`break`, `continue`).

10. Бақылау сұрақтары:

1. Басқару конструкциялары дегеніміз не?
2. `if` және `switch` операторларының айырмашылығы неде?
3. `for`, `while`, және `do...while` циклдерінің айырмашылықтарын атаңыз.
4. `break` және `continue` операторларының қызметі қандай?
5. Циклдің шексіз орындалуының себебі неде болуы мүмкін?
6. Құрамдас (nested) цикл деген не және ол не үшін қолданылады?
7. do...while циклі қай жағдайда қолайлы?