



C# DASTURLASH TILIDA REKURSIYA VA ITERATSIYA USULLARINING SAMARADORLIK TAHLILI

Ergashev Po'latbek Nuriddin o'g'li

Kazakova Shahnoz Janatbek qizi

O'zbekiston Milliy universiteti Jizzax filiali talabasi

Annotatsiya: *Ushbu maqolada C# dasturlash tilida rekursiya va iteratsiya usullarining samaradorligi tahlil qilinadi. Bir xil algoritmik masalalar ikkala usulda ham amalga oshirilib, ularning bajarilish tezligi, xotira sarfi va vaqt murakkabligi solishtirildi. Tadqiqot natijalari shuni ko'rsatadiki, iterativ usul ko'p hollarda tezroq va xotira jihatidan samaraliroq ishlaydi, rekursiv usul esa kod soddaligi va mantiqiy tuzilishi bilan ustunlik qiladi.*

Kalit so'zlar: rekursiya, iteratsiya, C#, algoritm, vaqt murakkabligi, stack, samaradorlik.

Kirish

Zamonaviy dasturlash jarayonida algoritmlarni samarali tashkil etish muhim ahamiyatga ega. Har bir masalani turli usullar yordamida yechish mumkin. Shular ichida rekursiya va iteratsiya keng qo'llaniladigan yondashuvlar hisoblanadi. Ularning har biri o'ziga xos afzallik va kamchiliklarga ega.

C# dasturlash tili yuqori darajadagi obyektga yo'naltirilgan til bo'lib, algoritmlarni yozishda ikkala usulni ham qo'llash imkonini beradi. Mazkur maqolada ushbu ikki yondashuvning samaradorligi tahlil qilinadi.

Asosiy qism

Rekursiya — bu funksiyaning o'zini o'zi chaqirishi orqali masalani kichik qismlarga ajratib yechish usulidir. Har bir rekursiv funksiya ikkita asosiy qismdan iborat bo'ladi: Bazaviy holat (to'xtash sharti), Rekursiv chaqiriq.

**Misol: Faktorial (rekursiv)**

```
static long FactorialRec(int n)
{
    if (n == 1)
        return 1;
    return n * FactorialRec(n - 1);
}
```

Rekursiv usul algoritmi qisqa va mantiqiy yozish imkonini beradi, biroq har bir chaqiriq stack xotirasidan foydalanadi.

Iteratsiya — bu sikl operatorlari (for, while, do-while) yordamida takrorlash asosida masalani yechish usulidir.

Misol: Faktorial (iterativ)

```
static long FactorialIter(int n)
{
    long result = 1;
    for (int i = 1; i <= n; i++)
        result *= i;
    return result;
}
```

Iterativ usulda qo'shimcha stack chaqiriqlari bo'lmaydi, shu sababli xotira sarfi kamroq bo'ladi.

Vaqt murakkabligi (Big O) tahlili

Faktorial misolida: Rekursiya: $O(n)$, Iteratsiya: $O(n)$.

Nazariy jihatdan vaqt murakkabligi bir xil bo'lsa ham, amaliy bajarilish tezligida farq kuzatiladi. Sababi — rekursiv chaqiriqlarda qo'shimcha stack operatsiyalari mavjud.

Amaliy tajriba



Tajriba davomida faktorial va Fibonacci algoritmlari ikkala usulda yozilib, bajarilish vaqti o'Ichandi

```
static int FibonacciRec(int n)
{
    if (n <= 1)
        return n;
    return FibonacciRec(n - 1) + FibonacciRec(n - 2);
}
```

Fibonacci (iterativ)

```
static int FibonacciIter(int n)
{
    if (n <= 1)
        return n;

    int a = 0, b = 1;
    for (int i = 2; i <= n; i++)
    {
        int temp = a + b;
        a = b;
        b = temp;
    }
    return b;
}
```

Natijalar shuni ko'rsatdiki:

- Fibonacci rekursiv usulda juda sekin ishlaydi (eksponensial murakkablik $O(2^n)$).
- Iterativ usul chiziqli vaqt murakkabligiga ega ($O(n)$).

Taqqoslash tahlili



Mezoni	Rekursiya	Iteratsiya
Kod uzunligi	Qisqa	Uzunroq
Tushunarliklik	Yuqori	O'rtacha
Vaqt samaradorligi	Ba'zan sekin	Tezroq
Xotira sarfi	Ko'proq	Kamroq
Stack Overflow xavfi	Bor	Yo'q

Xulosa

Tadqiqot natijalari shuni ko'rsatadiki, C# dasturlash tilida iterativ usul ko'p hollarda samaraliroq hisoblanadi. Ayniqsa katta hajmdagi hisoblashlarda iteratsiya tezlik va xotira jihatidan ustunlik qiladi. Rekursiya esa mantiqiy soddalik va algoritmnı tushunarli yozishda qulaylik yaratadi.

Shu sababli algoritm tanlashda masalaning xususiyati va bajarilish muhiti hisobga olinishi zarur.

FOYDALANILGAN ADABIYOTLAR

- 1.Troelsen A., Japikse P. *Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming*. – New York: Apress, 2022.
- 2.Albahari J., Albahari B. *C# 10 in a Nutshell: The Definitive Reference*. – Sebastopol: O'Reilly Media, 2022.
- 3.Skeet J. *C# in Depth*. – 4th ed. – Manning Publications, 2019.
- 4.CLR via C# / Richter J. – 4th ed. – Microsoft Press, 2012.
- 5.Cormen T., Leiserson C., Rivest C., Stein C. *Introduction to Algorithms*. – 3rd ed. – MIT Press, 2009.