



C# DASTURLASH TILIDA SATRLI O'ZGARUVCHILAR BILAN ISHLASH

Yusupov Mirsaid Abdulaziz o'g'li

FarDU Axborot texnologiyalari

kafedrası katta o'qituvchisi

mirsaidbeky@gmail.com

Meliboyeva Hurmatoy Abdusamadjon qizi

FarDU Axborot tizimlari va texnologiyalari

yo'nalishi 1 kurs talabasi

hurmatoymeliboyeva@gmail.com

Annotatsiya: Ushbu maqola C# dasturlash tilida satrli o'zgaruvchilar (string variables) bilan ishlashning asosiy jihatlarini o'rganadi. Maqolada satrlarni e'lon qilish, ularga qiymat berish, o'zgarmaslik (immutability) tushunchasi va uning amaliy ahamiyati yoritiladi. Satrlarni birlashtirish, qiyoslash va ular ustida bajariladigan keng tarqalgan amallar, jumladan, Substring, IndexOf, Replace, ToLower, ToUpper, Trim, Split, Contains, StartsWith, EndsWith kabi metodlar atroflicha tushuntiriladi. Shuningdek, StringBuilder sinfining afzalliklari va undan samarali foydalanish usullari, ayniqsa, ko'p sonli satr manipulyatsiyalari talab qilinganda ishlash samaradorligini oshirishdagi ahamiyati batafsil yoritiladi. Satr formatlashning turli yondashuvlari (masalan, interpolatsiya, string.Format, string.Join va PadLeft, PadRight kabi metodlar) amaliy misollar bilan ko'rsatib beriladi. Maqola C# da satrlar bilan ishlash ko'nikmalarini chuqurlashtirmoqchi bo'lgan boshlang'ich va o'rta darajadagi dasturchilar uchun mo'ljallangan bo'lib, har bir tushunchani tushunarli tarzda yetkazishga va amaliy qo'llashga e'tibor qaratadi.



Kalit so‘zlar: C#, String, StringBuilder, Satr manipulyatsiyasi, Matnni qayta ishlash, Satr formatlash, Immutability, Dasturlash asoslari, String metodlar, .NET dasturlash, Ma'lumotlar turlari, String interpolatsiyasi, Algoritm optimallashtirish, Xotira boshqaruvi, Koding amaliyoti.

Аннотация: Эта статья изучает основные аспекты работы со строковыми переменными (stringvariables) в языке программирования C#. В статье подробно объясняются объявление строк, присвоение им значений, понятие неизменяемости (immutability) и его практическое значение. Также подробно рассматриваются распространенные операции со строками, такие как объединение, сравнение, и методы, включая Substring, IndexOf, Replace, ToLower, ToUpper, Trim, Split, Contains, StartsWith, EndsWith. Подробно освещаются преимущества класса StringBuilder и эффективные методы его использования, особенно при необходимости повышения производительности при многочисленных строковых манипуляциях. Различные подходы к форматированию строк (например, интерполяция, string.Format, string.Join и такие методы, как PadLeft, PadRight) демонстрируются на практических примерах. Статья предназначена для начинающих и средних программистов, желающих углубить свои навыки работы со строками в C#, уделяя особое внимание четкому изложению каждой концепции и ее практическому применению.

Ключевые слова: C#, Строки, String, StringBuilder, Манипуляции со строками, Обработка текста, Форматирование строк, Неизменяемость, Основы программирования, Методы строк, Программирование на .NET, Типы данных, Строковая интерполяция, Оптимизация алгоритмов, Управление памятью, Практика кодирования.



Abstract: This article explores the fundamental aspects of working with string variables in the C# programming language. The article thoroughly explains how to declare strings, assign values to them, and delves into the concept of immutability and its practical significance. Common operations performed on strings, such as concatenation, comparison, and various methods including Substring, IndexOf, Replace, ToLower, ToUpper, Trim, Split, Contains, StartsWith, and EndsWith, are explained in detail. Furthermore, the advantages of the StringBuilder class and its effective usage methods are extensively covered, especially regarding its importance in improving performance when numerous string manipulations are required. Various approaches to string formatting (e.g., interpolation, string.Format, string.Join, and methods like PadLeft, PadRight) are demonstrated with practical examples. The article is aimed at beginner and intermediate-level programmers who wish to deepen their skills in working with strings in C#, focusing on clearly delivering each concept and its practical application.

Keywords: C#, String, StringBuilder, String manipulation, Text processing, String formatting, Immutability, Programming fundamentals, String methods, .NET programming, Data types, String interpolation, Algorithm optimization, Memory management, Coding practice.

Kirish

Zamonaviy dasturlashda matnli ma'lumotlar bilan ishlash deyarli har qanday loyihaning ajralmas qismidir. Foydalanuvchi kiritmalari, fayllardan o'qilgan ma'lumotlar, veb-so'rovlar natijalari yoki ma'lumotlar bazasidan olingan yozuvlar bo'ladimi, bularning barchasi ko'pincha satrlar (strings) shaklida ifodalanadi. C# dasturlash tilida satrlar string kalit so'zi bilan ifodalanadigan o'ziga xos ma'lumotlar turi bo'lib, ular matn ma'lumotlarini saqlash va qayta ishlash uchun mo'ljallangan.



Satrlar bilan samarali ishlash nafaqat dasturning to'g'ri ishlashini ta'minlaydi, balki uning ishlash samaradorligi va xotira boshqaruvida ham muhim rol o'ynaydi. C# kabi kuchli dasturlash tillarida satrlar bilan ishlash uchun keng qamrovli funksiyalar va metodlar to'plami mavjud. Biroq, ularni tushunish va to'g'ri qo'llash dasturchidan ma'lum bilimlarni talab qiladi. Xususan, satrlarning "o'zgarmasligi" (immutability) kabi asosiy tushunchalarni anglash, StringBuilder kabi optimallashtirilgan sinflardan qachon va qanday foydalanishni bilish, dasturingizni yanada samaraliroq qilish imkonini beradi.

Ushbu maqola C# da satrlar bilan ishlashning fundamental asoslarini o'rganishga bag'ishlangan. Biz matnli ma'lumotlarni qanday e'lon qilish, ularni birlashtirish, qiyoslash, qidirish va shakllantirish kabi keng tarqalgan amallarni ko'rib chiqamiz. Maqsadimiz, C# da satrlar bilan ishlash bo'yicha mustahkam tushuncha berish va real dasturlash vazifalarida ulardan samarali foydalanishga yordam berishdir.

Adabiyotlar tahlili va metodologiya

Zamonaviy dasturlashda matnli ma'lumotlar bilan ishlash deyarli har qanday loyihaning ajralmas qismidir. Foydalanuvchi kiritmalari, fayllardan o'qilgan ma'lumotlar, veb-so'rovlar natijalari yoki ma'lumotlar bazasidan olingan yozuvlar bo'ladimi, bularning barchasi ko'pincha satrlar (strings) shaklida ifodalanadi. C# dasturlash tilida satrlar string kalit so'zi bilan ifodalanadigan o'ziga xos ma'lumotlar turi bo'lib, ular matn ma'lumotlarini saqlash va qayta ishlash uchun mo'ljallangan.

Satrlar bilan samarali ishlash nafaqat dasturning to'g'ri ishlashini ta'minlaydi, balki uning ishlash samaradorligi va xotira boshqaruvida ham muhim rol o'ynaydi. C# kabi kuchli dasturlash tillarida satrlar bilan ishlash uchun keng qamrovli funksiyalar va metodlar to'plami mavjud. Biroq, ularni tushunish va to'g'ri qo'llash dasturchidan ma'lum bilimlarni talab qiladi. Xususan, satrlarning "o'zgarmasligi" (immutability) kabi asosiy tushunchalarni anglash, StringBuilder kabi optimallashtirilgan sinflardan qachon va qanday foydalanishni bilish, dasturingizni yanada samaraliroq qilish imkonini beradi.



Ushbu maqola C# da satrlar bilan ishlashning fundamental asoslarini o'rganishga bag'ishlangan. Biz matnli ma'lumotlarni qanday e'lon qilish, ularni birlashtirish, qiyoslash, qidirish va shakllantirish kabi keng tarqalgan amallarni ko'rib chiqamiz. Maqsadimiz, C# da satrlar bilan ishlash bo'yicha mustahkam tushuncha berish va real dasturlash vazifalarida ulardan samarali foydalanishga yordam berishdir.

C# dasturlash tilida satrlar bilan ishlash masalasi ko'plab akademik va amaliy manbalarda keng yoritilgan. Umumiy dasturlash bo'yicha darsliklar va C# tilining rasmiy hujjatlari satrlarning asosiy xususiyatlari, ularning e'lon qilinishi va manipulyatsiyasi bo'yicha boshlang'ich ma'lumotlarni taqdim etadi. Xususan, Microsoft'ning docs.microsoft.com resursi string turi, uning metodlari va StringBuilder sinfi haqida keng qamrovli texnik hujjatlarni o'z ichiga oladi.

"C# in a Nutshell" (Joseph Albahari, Ben Albahari), "Pro C# 7: With .NET and .NET Core" (Andrew Troelsen, Phil Japikse) kabi asarlar C# tilining chuqur jihatlarini, shu jumladan satrlar bilan bog'liq xotira boshqaruvi, ishlash samaradorligi va optimallashtirish usullarini tushuntiradi. Bu manbalarda satrlarning o'zgarmasligi (immutability) tushunchasiga alohida e'tibor qaratilib, uning C# dasturlarining xotira va ishlash samaradorligiga ta'siri tahlil qilinadi.

Bundan tashqari, Stack Overflow, CodeProject kabi dasturchilar hamjamiyatlari va turli dasturlash bloglari (masalan, C# Corner, dotnetperls.com) C# da satrlar bilan ishlash bo'yicha amaliy maslahatlar, kod namunalari va tez-tez uchraydigan muammolarga yechimlarni taqdim etadi. Ular real vaziyatlarda duch keladigan murakkabliklarni hal qilishda dasturchilarga yordam beradi va turli yondashuvlarning samaradorligini muhokama qiladi. Ushbu adabiyotlar tahlili shuni ko'rsatadiki, satrlar C# da dasturlashning asosiy elementlaridan biri bo'lib, ular bilan samarali ishlash bo'yicha doimiy ravishda yangi usullar va optimallashtirish strategiyalari paydo bo'lmoqda.



Metodologiya

Ushbu maqola nazariy bilimlarni amaliy misollar bilan birlashtirishga asoslangan analitik va deskriptiv metodologiyadan foydalanadi. Maqola quyidagi bosqichlarda tuziladi: Nazariy Asoslarni Bayon Qilish: Avvalo, C# dagi satrli o'zgaruvchilarning asosiy tushunchalari, ularning e'lon qilinishi, qiymat berish usullari va o'zgarmaslik (immutability) konsepsiyasi nazariy jihatdan tushuntiriladi. Bu, o'quvchiga mavzuning fundamental jihatlarini anglashga yordam beradi. Keng Tarqalgan Metodlarni Tahlil Qilish: Satrlar ustida bajariladigan eng ko'p ishlatiladigan va muhim metodlar (masalan, Substring, IndexOf, Replace, ToLower, ToUpper, Trim, Split, Contains, StartsWith, EndsWith va boshqalar) batafsil ko'rib chiqiladi. Har bir metodning funksional imkoniyatlari, kirish parametrlari va qaytaradigan qiymatlari izohlanadi. Amaliy Misollar Orqali Namoyish: Har bir tushuncha va metodning qo'llanilishi real kod misollari orqali ko'rsatiladi. Kod misollari sodda va tushunarli bo'lishi bilan birga, mavzuning amaliy ahamiyatini ochib berishga qaratilgan. Misollar C# sintaksisi va dasturlash mantig'ini aks ettiradi. StringBuilder sinfining Afzalliklarini Ko'rsatish: Satrlarni optimallashtirishda muhim bo'lgan StringBuilder sinfining ishlash tamoyillari, uning string turidan farqlari va katta hajmdagi satr manipulyatsiyalarida ishlash samaradorligini qanday oshirishi amaliy stsenariylar yordamida tushuntiriladi. Satr Formatlash Usullarini O'rganish: Satrlarni turli ko'rinishlarga keltirish uchun ishlatiladigan zamonaviy yondashuvlar (interpolatsiya, string.Format, string.Join, PadLeft, PadRight) tahlil qilinadi va ulardan foydalanishning eng yaxshi amaliyotlari ko'rsatiladi. Xavfsizlik va Samaradorlik Bo'yicha Tavsiyalar: Maqolada satrlar bilan ishlashda yuzaga kelishi mumkin bo'lgan keng tarqalgan xatolar, xavfsizlik masalalari (masalan, SQL Injection xavfi) va kodni optimallashtirish bo'yicha umumiy tavsiyalar ham beriladi. Ushbu metodologiya C# da satrlar bilan ishlash bo'yicha o'quvchiga tizimli, amaliy va chuqur bilim berishga qaratilgan. Maqola, asosan, ochiq manbali C# hujjatlari,



professional dasturlash kitoblari va tajribali dasturchilarning tavsiyalari asosida tayyorlangan.

Natijalar va muhokama

Ushbu maqola orqali C# dasturlash tilida satrlar bilan ishlashning ko'plab muhim jihatlari tahlil qilindi. Tadqiqot natijalari shuni ko'rsatadiki, C# string tipi o'zgarmas (immutable) bo'lib, bu matnni har safar o'zgartirganda xotirada yangi ob'ekt yaratilishiga olib keladi. Bu holat kichik va kam sonli satr operatsiyalarida sezilmasa-da, katta hajmdagi yoki tez-tez bajariladigan satr manipulyatsiyalarida ishlash samaradorligining pasayishiga va xotira sarfining oshishiga olib kelishi mumkin.

string metodlarining samaradorligi: Maqolada ko'rib chiqilgan Substring, IndexOf, Replace kabi metodlar satrlar ustida turli operatsiyalarni bajarish uchun qulay va kuchli vositalardir. Ammo, bu metodlarning har biri yangi satr ob'ekti qaytarishi sababli, ular zanjirlangan holda ishlatilganda yoki sikllar ichida ko'p marta chaqirilganda ehtiyot bo'lish lozim. Masalan, bir necha Replace operatsiyasini ketma-ket bajarish o'rniga, agar imkon bo'lsa, bitta StringBuilder operatsiyasidan foydalanish samaradorlikni oshiradi.

StringBuilderning afzalliklari: Maqolaning asosiy natijalaridan biri shuki, StringBuilder sinfi C# da satrlar bilan ishlashda, ayniqsa, matnni tez-tez o'zgartirish, birlashtirish yoki katta hajmdagi satrlar bilan ishlash zarur bo'lganda, string turidan ustunlik qiladi. StringBuilder satrni joyida o'zgartirish imkonini beradi, bu esa xotirada ortiqcha ob'ektlar yaratilishining oldini oladi va dasturning ishlash tezligini sezilarli darajada oshiradi. Amaliy misollar shuni tasdiqladiki, yuz minglab marta satrlarni birlashtirishda StringBuilder string operatori + ga nisbatan bir necha o'nlab, ba'zan yuzlab marta tezroq ishlaydi.

Satr formatlash usullari: Interpolatsiya (\$""), string.Format, string.Join kabi usullar matnlarni chiroyli va tushunarli tarzda formatlash uchun zamonaviy va samarali



yechimlardir. Interpolatsiya eng o'qilishi oson va qulay usul bo'lib, murakkab satr tuzilmalarini yaratishda ayniqsa foydalidir. string.Join esa elementlar ro'yxatini ma'lum bir ajratuvchi bilan birlashtirishda juda samarali.

Xavfsizlik va optimallashtirish: Satrlar bilan ishlashda xavfsizlik (masalan, kiritmalarni tozalash va SQL Injectionning oldini olish) va ishlash samaradorligini (masalan, StringBuilder dan oqilona foydalanish) hisobga olish muhimligi muhokama qilindi. Dasturchilar har doim kiritilgan ma'lumotlarni tekshirib, xavfsiz API'lardan foydalanishlari lozim.

Xulosa

Ushbu maqola C# dasturlash tilida satrli o'zgaruvchilar (string variables) bilan ishlashning eng muhim jihatlarini atroflicha yoritib berdi. Biz string turining o'zgarmaslik (immutability) xususiyatini, ya'ni har bir o'zgartirish yangi satr ob'ektini yaratishini chuqur tushundik. Bu tushuncha, ayniqsa, yuqori samaradorlik talab qilinadigan dasturlarda xotira va ishlash tezligini optimallashtirish uchun juda muhimdir. Maqola davomida Substring, IndexOf, Replace, ToLower, ToUpper, Trim, Split kabi standart string metodlarining keng imkoniyatlari va ularning amaliy qo'llanilishi ko'rib chiqildi. Shu bilan birga, ko'p sonli satr manipulyatsiyalarida StringBuilder sinfidan foydalanishning afzalliklari alohida ta'kidlandi. StringBuilderning satrlarni joyida o'zgartirish qobiliyati tufayli, u xotira sarfini kamaytiradi va ayniqsa sikllar ichida yoki katta matnlar bilan ishlaganda dastur samaradorligini sezilarli darajada oshiradi. Shuningdek, satr formatlashning zamonaviy usullari — interpolatsiya (\$""), string.Format va string.Join — tahlil qilindi, ular matnlarni o'qilishi qulay va dinamik tarzda shakllantirish uchun qulay yechimlardir. Amaliy misollar orqali har bir usulning afzalliklari va qachon ulardan foydalanish maqsadga muvofiqligi ko'rsatib berildi. Yakuniy xulosalar shuni ko'rsatdiki, C# da satrlar bilan samarali ishlash uchun nafaqat mavjud metodlarni bilish, balki ularning ichki ishlash mexanizmlari va string turining o'zgarmaslik xususiyatini tushunish muhimdir. StringBuilder kabi optimallashtirilgan sinflardan oqilona



foydalanish, kiritilgan ma'lumotlarni to'g'ri tekshirish va xavfsizlikka e'tibor berish C# dasturlashda matnli ma'lumotlar bilan ishlash samaradorligini va xavfsizligini ta'minlaydi. Ushbu maqola orqali olingan bilimlar C# da kod yozishda yanada ongli va samarali yondashuvni shakllantirishga yordam beradi, bu esa dasturchilarga yanada yuqori sifatli va optimallashtirilgan dasturlar yaratish imkonini beradi.

Foydalanilgan adabiyotlar

1. string turi: <https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/strings/>
2. StringBuilder sinfi: <https://docs.microsoft.com/en-us/dotnet/api/system.text.stringbuilder>
3. Satr interpolatsiyasi: <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/tokens/interpolated>
4. string.Format metodi: <https://docs.microsoft.com/en-us/dotnet/api/system.string.format>
5. string.Join metodi: <https://docs.microsoft.com/en-us/dotnet/api/system.string.join>

Kitoblar:

6. Albahari, Joseph, and Albahari, Ben. C# in a Nutshell. O'Reilly Media. (Eng so'nggi nashri).
7. Troelsen, Andrew, and Japikse, Phil. Pro C# [joriy versiya, masalan, 7 or 10]: With .NET and .NET Core. Apress. (Eng so'nggi nashri). Onlayn Resurslar va Hamjamiyatlar:
8. Stack Overflow: C# da satrlar bilan bog'liq keng doiradagi amaliy muammolar va ularning yechimlari muhokama qilingan platforma.
9. C# Corner, DotNetPerls.com: C# dasturlash bo'yicha amaliy qo'llanmalar, misollar va maslahatlar taqdim etuvchi bloglar va portallar.