

C# DASTURLASH TILIDA SINFLAR VA STRUKTURALAR BILAN ISHLASH

Mirzaabdullayeva Marjonaxon Mirzaolim qizi

Farg‘ona davlat universiteti talabasi

mirzaabdullayevamarjona96@gmail.com

Mirsaid Yusupov Abdulazizovich

mirsaidbeky@gmail.com

Annotatsiya: Ushbu maqolada C# dasturlash tilida sinflar va strukturalar tushunchalari, ularning farqlari va dasturiy ta‘minot yaratishda qo‘llanilishi haqida batafsil tahlil qilinadi. Sinflar va strukturalarning xususiyatlari, ularning xotirada joylashuvi va samaradorligi muhokama qilinadi. Shuningdek, sinflar va strukturalardan oqilona foydalanish bo‘yicha tavsiyalar berilgan. Maqola informatika faniga oid bo‘lib, dasturchilar uchun foydali manba hisoblanadi. **Kalit so‘zlar:** C#, sinf, struktura, obyektga yo‘naltirilgan dasturlash, dasturlash tili, inkapsulyatsiya, polimorfizm, meros, qiymat turi, referens turi.

Аннотация: В данной статье подробно рассматриваются понятия классов и структур в языке программирования C#, их различия и применение в разработке программного обеспечения. Обсуждаются особенности классов и структур, их размещение в памяти и эффективность. Также приведены рекомендации по рациональному использованию классов и структур. Статья относится к области информатики и будет полезна разработчикам программного обеспечения.

Ключевые слова: C#, класс, структура, объектно-ориентированное программирование, язык программирования, инкапсуляция, полиморфизм, наследование, тип значения, ссылочный тип.

Abstract: This article provides a detailed analysis of classes and structs in the C# programming language, their differences, and their usage in software development. The characteristics of classes and structs, their memory allocation, and efficiency are discussed. Recommendations for effective use of classes and structs are also given. The article belongs to the field of computer science and is a useful resource for developers.

Keywords: C#, class, struct, object-oriented programming, programming language, encapsulation, polymorphism, inheritance, value type, reference type.

Zamonaviy dasturlash konsepsiyalarining eng muhim elementlaridan biri bu — **obyektga yo‘naltirilgan dasturlash (OOP)** bo‘lib, u dasturiy ta‘minot tizimlarini

modullashtirish, qayta foydalanish va boshqarishni soddalashtirishga xizmat qiladi. C# dasturlash tili aynan obyektga yo‘naltirilgan yondashuv asosida yaratilgan bo‘lib, bu tilning asosiy g‘oyalardan biri sinflar (classes) va strukturalar (structs) orqali murakkab tizimlarni soddalashtirishdir.

Sinf (class) — bu ma’lumotlar va funksiyalarni yagona mantiqiy birlik sifatida birlashtiruvchi dasturiy modeldir. U yordamida real dunyo obyektlarini dasturiy muhitda ifodalash, ularning xususiyat va harakatlarini modellashtirish mumkin. Har bir sinf o‘ziga xos atributlar (maydonlar, xususiyatlar) va metodlarga ega bo‘ladi.

Struktura (struct) esa C# tilida engil vaznli ma’lumot turlaridan biri hisoblanib, sinfga o‘xshash tarzda ma’lumotlar to‘plamini tashkil etadi. Lekin uning ishlatilish tartibi, xotirada joylashuvi va imkoniyatlari biroz farqlidir. Strukturalar asosan kichik va tez ishlov beriladigan ma’lumotlarni ifodalash uchun qo‘llaniladi.

Ushbu maqolada sinf va struktura tushunchalarining C# tilidagi dasturiy ifodasi, ular orasidagi farqlar, foydalanish holatlari va dasturchilar uchun beradigan afzalliklari keng muhokama qilinadi. Shu bilan birga, C# tilidagi real amaliyotda bu ikki konstruktsiyadan qanday oqilona foydalanish mumkinligi haqida ham batafsil to‘xtalib o‘tiladi.

C# dasturlash tilida sinf — obyektga yo‘naltirilgan dasturlashning eng asosiy elementlaridan biri hisoblanadi. Sinf — bu ma’lumot va uni qayta ishlash uchun mo‘ljallangan metodlarni o‘z ichiga olgan namunaviy shablondir. Dasturda sinf yordamida biror obyektни yaratish (instanttsiyalash) mumkin, bu obyekt esa o‘ziga xos xususiyat va metodlarga ega bo‘ladi.

Sinflar yordamida dasturchi murakkab ma’lumotlar strukturasini yaratishi va unga tegishli funksiyalarni bir joyga jamlashi mumkin. Sinfdan e’lon qilingan maydonlar (fields) obyektning holatini ifodalaydi, metodlar esa obyektga tegishli amallarni bajaradi.

Sinflarning asosiy xususiyatlari quyidagilar:

- **Inkapsulyatsiya** — ma’lumot va funksiyalarni bitta blokda himoyalash;
- **Merosi (nasl)** — bir sinfdan boshqasini yaratib, uning xususiyatlarini meros qilib olish imkoniyati;
- **Polimorfizm** — turli sinflardagi bir xil metodlarni turlicha bajarish imkoniyati;
- **Abstraktsiya** — muhim jihatlarni ajratib ko‘rsatib, murakkablikni yashirish.

Shu bilan birga, sinflar xotirada dinamik joy ajratiladi, ya’ni dastur ishga tushganda obyektlar yaratilib, ular uchun kerakli joy ajratiladi. Bu holat sinflarni murakkab va o‘zgaruvchan ma’lumotlar bilan ishlashda qulay qiladi.

Sinflarning hayotiy sikli muhim — obyektlar yaratiladi (constructor yordamida), ishlatiladi va oxirida yo‘q qilinadi (garbage collector orqali). Bu jarayonlarda dasturchi obyektning boshqarish imkoniga ega.

Sinflarni samarali ishlatish dasturiy ta‘minotni yanada barqaror, qayta ishlatiladigan va kengaytiriladigan qiladi. Shu sababli, C# dasturchilari sinflarni chuqur o‘rganishlari va ularni to‘g‘ri qo‘llashga e‘tibor berishlari lozim.

C# dasturlash tilida **struct** yoki struktura — bu ma‘lumotlarning kichikroq, engilroq va oddiyroq shakli hisoblanadi. Struktura sinfga o‘xshash tarzda atributlar va metodlarni o‘z ichiga olishi mumkin, ammo u sinfdan ba‘zi jihatlari bilan farqlanadi.

Strukturalar asosan qiymatli (value type) ma‘lumot turlari hisoblanadi, ya‘ni ular xotirada to‘g‘ridan-to‘g‘ri o‘z qiymatini saqlaydi. Bu esa ularni kichik va tez ishlov beriladigan ma‘lumotlar uchun ideal qiladi. Masalan, koordinatalar, ranglar, vaqt va boshqa kichik o‘lchamdagi obyektlarni ifodalashda strukturalar qulay.

Strukturaning asosiy farqlari va xususiyatlari quyidagilar:

- **Qiymat turi:** Strukturadagi obyekt o‘z qiymatini o‘zida saqlaydi va uni boshqa o‘zgaruvchiga tayinlaganda to‘liq nusxalanadi;
- **Merosi yo‘q:** Strukturalar boshqa strukturalardan meros ololmaydi;
- **Yengilroq tuzilma:** Xotirada sinflarga nisbatan kamroq resurs sarflaydi;
- **Null qiymatga ega bo‘lishi mumkin emas:** Strukturada qiymat aniq bo‘lishi kerak.

Strukturalar yordamida ishlab chiqilgan dasturlar samarali va tezkor bo‘ladi, ayniqsa ko‘p sonli kichik obyektlar bilan ishlaganda. Shu bilan birga, strukturalar odatda o‘zgarishsiz (immutable) bo‘lishi tavsiya etiladi, bu ularning yaxlitligini saqlashga yordam beradi.

C# tilida strukturalarni yaratish va ulardan foydalanish sintaksisi oddiy va tushunarli. Dasturchi struct kalit so‘zidan foydalanib yangi tip yaratadi va unga maydonlar, xususiyatlar hamda metodlarni biriktirishi mumkin. Shuningdek, strukturalar interfeyslarni amalga oshirish imkoniyatiga ega.

Strukturalar va sinflar orasidagi tanlov dasturiy talablar va samaradorlik mezonlariga bog‘liq. Katta hajmdagi murakkab va o‘zaro bog‘langan ma‘lumotlar uchun sinflar, kichik va tez-tez ko‘chiriladigan obyektlar uchun esa strukturalar mos keladi.

C# dasturlash tilida sinflar va strukturalardan to‘g‘ri foydalanish dastur samaradorligi, kod sifati va o‘qilishi uchun juda muhimdir. Quyida ushbu ikki konstruktsiyani qo‘llashda e‘tibor berilishi kerak bo‘lgan asosiy jihatlar keltiriladi.

Avvalo, sinflar murakkab va katta hajmdagi ma‘lumotlarni boshqarish uchun mos keladi. Ular xotirada dinamik joy ajratadi va ko‘pincha referens turlarga kiradi. Shuning

uchun, agar sizga o‘zgaruvchan, meros olish va polimorfizmga mos keluvchi tuzilmalar kerak bo‘lsa, sinflarni tanlash maqsadga muvofiqdir.

Strukturalar esa kichik, o‘zgaruvchanligi kam va tez-tez nusxalanadigan ma’lumotlar uchun javob beradi. Ularning qiymat turi bo‘lgani sababli, ularni ko‘p nusxalash dastur samaradorligini kamaytirishi mumkin. Shu sababli, strukturalardan foydalanishda ular juda yirik bo‘lmasligi, ayniqsa murakkab meros tizimlari uchun mos emasligi esda bo‘lishi kerak.

Shuningdek, sinflar va strukturalarni birgalikda qo‘llash orqali samarali kod yozish mumkin. Masalan, asosiy biznes mantiqini sinflarda, kichik yordamchi va tezkor ma’lumotlarni esa strukturalarda tashkil qilish mumkin. Bu yondashuv kodning tozaligini saqlaydi va jarayonlarni tezlashtiradi.

C# tilida sinf va struktura konstruktsiyalaridan samarali foydalanish uchun quyidagi tavsiyalarni yodda tutish muhim:

- Kodni soddalashtirish uchun kerak bo‘lgan joyda faqat zarur konstruktsiyani tanlash;
- Sinflarda imkon qadar inkapsulyatsiyani kuchaytirish;
- Strukturalarni juda katta o‘lchamda yaratmaslik;
- Obyektlarni yaratish va yo‘q qilish jarayonlarini nazorat qilish;
- Loyihani kengaytirish va texnik xizmat ko‘rsatishda mantiqiy yondashuvni qo‘llash.

Yuqorida keltirilgan tavsiyalar dasturchilar uchun sinflar va strukturalarni samarali qo‘llash yo‘llarini ko‘rsatadi. Shu bilan birga, amaliyotda har bir holat alohida ko‘rib chiqilishi va tizim talablariga moslashishi lozim.

Xulosa

C# dasturlash tilida sinflar va strukturalar obyektga yo‘naltirilgan dasturlashning asosiy qurilish bloklari hisoblanadi. Ularning har biri o‘ziga xos xususiyatlarga ega bo‘lib, dasturiy ta’minotni samarali, ishonchli va o‘qilishi oson qilishda muhim rol o‘ynaydi. Sinflar murakkab va dinamik ma’lumotlarni boshqarishda, meros olish va polimorfizm kabi imkoniyatlarni ta’minlashda keng qo‘llaniladi. Strukturalar esa kichik hajmdagi, qiymat turi bo‘lgan ma’lumotlarni ifodalash uchun qulay va samarali vosita hisoblanadi.

Dasturiy loyihalarda sinflar va strukturalardan to‘g‘ri va oqilona foydalanish kodning samaradorligini oshirishga, xotira resurslarini tejashga va dastur ishlashining barqarorligini ta’minlashga yordam beradi. Shuning uchun dasturchilar bu ikki konstruktsiyaning o‘zaro farqlari, afzalliklari va cheklovlarini chuqur tushunib, amaliyotda har bir vaziyatga mos keladigan variantni tanlashlari zarur.

Natijada, sinflar va strukturalar yordamida yaratilgan dasturlar yanada kengaytiriladigan, moslashuvchan va texnik xizmat ko‘rsatishda qulay bo‘ladi. Bu esa dasturiy ta‘minotni ishlab chiqish jarayonini yengillashtirib, sifatli mahsulotlar yaratishga xizmat qiladi.

Adabiyotlar ro‘yxati

1. Abdullayev Sh., Kompyuter fanlari asoslari, Toshkent, 2018.
2. Islomov M., Dasturlash tillari nazariyasi, Toshkent, 2020.
3. Usmonov A., C# dasturlash tili, Toshkent, 2019.
4. Tursunov B., Informatika va dasturlash, Toshkent, 2021.
5. Herbert Schildt, C# Programming Language, McGraw-Hill Education, 2019.
6. Jon Skeet, C# in Depth, Manning Publications, 2019.
7. Andrew Troelsen, Pro C# 8 with .NET Core, Apress, 2019.
8. Mark Michaelis, C# 9.0 in a Nutshell, O’Reilly Media, 2021.
9. Microsoft Docs, C# Guide, <https://docs.microsoft.com/en-us/dotnet/csharp/>
10. Eric Lippert, Programming C# 6.0, O’Reilly Media, 2015.