

DASTURLARNI BAJARILISHI**Umarov Bekzod Azizovich***Farg‘ona davlat universiteti*ubaumarov@gmail.com**Abdumo‘minova Sevinch Bekzod qizi***Farg‘ona davlat universiteti**2-kurs talabasi*sevinchinsider@gmail.com

Anotatsiya. Ushbu maqolada dasturlarni bajarilish jarayoni, kompilyatsiya va interpretatsiya tushunchalari, ularning afzalliklari hamda kamchiliklari yoritilgan. Shuningdek, zamonaviy dasturlash tillarida dastur bajarilishining o‘ziga xos xususiyatlariga ham to‘xtalib o‘tilgan.

Kalit so‘zlar. Dastur, kompilyatsiya, interpretatsiya, bajarilish, dasturlash tillari.

Аннотация. В данной статье рассмотрен процесс выполнения программ, а также понятия компиляции и интерпретации, их преимущества и недостатки. Кроме того, обсуждаются особенности выполнения программ на современных языках программирования.

Ключевые слова. Программа, компиляция, интерпретация, выполнение, языки программирования.

Annotation. This article discusses the process of program execution, as well as the concepts of compilation and interpretation, their advantages and disadvantages. It also highlights specific features of execution in modern programming languages.

Keywords. Program, compilation, interpretation, execution, programming languages.

Kirish

Zamonaviy axborot texnologiyalari hayotimizning deyarli barcha jabhalariga chuqur kirib bormoqda. Bu jarayonda dasturlar alohida o‘rin egallaydi, chunki ular turli masalalarni avtomatlashtirish, ma’lumotlarni qayta ishlash va boshqarish imkonini beradi. Har qanday dastur faqat yozilishi bilan emas, balki qanday bajarilishi bilan ham muhim hisoblanadi. Dastur bajarilishi deganda, uning kodidagi buyruqlarni kompyuter tomonidan tushunilishi va amalda bajarilishi tushuniladi. Dastur qanday bajarilishi — ya’ni kompilyatsiya yo‘li bilanmi yoki interpretatsiya asosidami — bu jarayon samaradorligiga, tezligiga va ishonchliligiga bevosita ta’sir ko‘rsatadi. Ushbu maqolada aynan dasturlarni bajarilish usullari, ularning o‘ziga xos xususiyatlari va zamonaviy dasturlash tillaridagi yondashuvlar

haqida so‘z yuritiladi. Dastur yozilgach, uning kompyuter tomonidan to‘g‘ri tushunilishi va bajarilishi muhim bosqich hisoblanadi. Bu jarayon asosan ikkita asosiy usul orqali amalga oshiriladi: kompilyatsiya va interpretatsiya.

Kompilyatsiya — bu dastur kodini birdaniga mashina tiliga tarjima qilish jarayonidir. Ya’ni, dasturchi tomonidan yozilgan yuqori darajadagi kod kompilyator yordamida tahlil qilinadi va bajarilishga tayyor binar faylga aylantiriladi. Bu usulda dastur avval to‘liq tarjima qilinadi, so‘ngra bajariladi. Shuning uchun kompilyatsiyalangan dasturlar odatda tez ishlaydi.

Kompilyatsiya afzalliklari:

Ishlash tezligi yuqori.

Xatoliklar dastur bajarilishidan oldin aniqlanadi.

Mustaqil ijro fayl hosil bo‘ladi.

Kamchiliklari:

Dasturdagi kichik o‘zgarishdan so‘ng, butun kodni qayta kompilyatsiya qilish kerak bo‘ladi.

Platformaga bog‘liqlik yuzaga kelishi mumkin.

Interpretatsiya — bu kodni real vaqt rejimida, qadam-baqadam bajarish usulidir. Dasturchi yozgan kodni interpretator dastur ketma-ket o‘qiydi va shu zahoti bajaradi. Python, JavaScript kabi tillar interpretatsiya usulida ishlaydi.

Interpretatsiya afzalliklari:

Dastur tez sinovdan o‘tkaziladi va tuzatishlar oson kiritiladi.

Platformaga moslashuvchanlik yuqori.

Kamchiliklari:

Ishlash tezligi nisbatan past bo‘lishi mumkin.

Xatoliklar bajarilish jarayonida yuzaga chiqadi.

Ba’zi zamonaviy dasturlash tillari (masalan, Java) gibril yondashuvdan foydalanadi. Java dasturi avval bayt-kodga kompilyatsiya qilinadi, keyin esa Java Virtual Machine (JVM) yordamida interpretatsiya qilinadi yoki JIT (Just-In-Time) kompilyatsiyasi orqali bajariladi. Bu yondashuv tezlik va platformaviy moslashuvchanlikni uyg‘unlashtirishga yordam beradi.

Jadval: Kompilyatsiya va interpretatsiya usullarining taqqoslanishi

№	Belgilar / Xususiyatlar	Kompilyatsiya	Interpretatsiya
1	Bajarilish tartibi	Dastur avval to‘liq tarjima qilinadi	Dastur real vaqt rejimida qadam-baqadam

			bajariladi
2	Ishlash tezligi	Tez	Nisbatan sekin
3	Xatolik aniqlanishi	Kompilyatsiya vaqtida	Bajarilish vaqtida
4	Dastur tuzilgan fayl	Mustaqil binar fayl (masalan, .exe)	Maxsus fayl hosil bo‘lmaydi
5	Platformaga bog‘liqlik	Yuqori	Past
6	Misollar	C, C++, Pascal	Python, JavaScript, PHP
7	Afzalliklar	Tezlik, aniqlik	Moslashuvchanlik, qulay test
8	Kamchiliklar	Qayta kompilyatsiya kerak bo‘ladi	Sekin ishlashi, runtime xatolar

C# dasturi misoli

Quyidagi C# dasturi foydalanuvchi kiritgan sonning faktorialini hisoblaydi:

`using System;`

`class Program`

`{`

`static int Faktorial(int n)`

`{`

`if (n == 0 || n == 1)`

`return 1;`

`else`

`return n * Faktorial(n - 1); }`

`static void Main()`

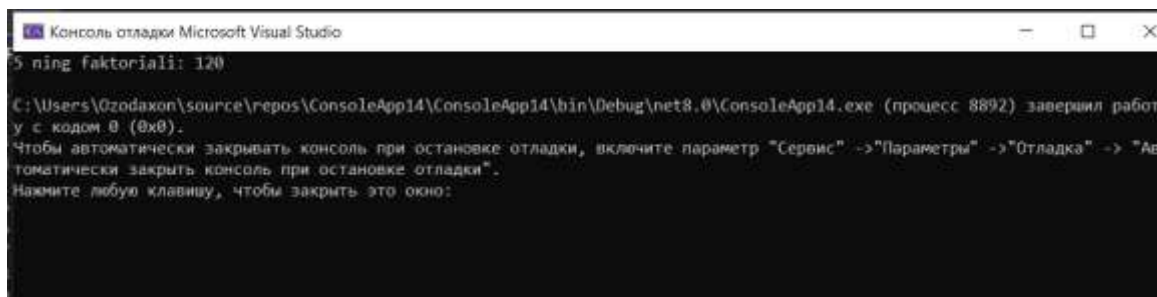
`{ int son = 5;`

`int natija = Faktorial(son);`

`Console.WriteLine($"{son} ning faktoriali: {natija}");`

`}}`

Natija:



```
Консоль отладки Microsoft Visual Studio
5 ning faktoriali: 120
C:\Users\Ozodaxon\source\repos\ConsoleApp14\ConsoleApp14\bin\Debug\net8.0\ConsoleApp14.exe (процесс 8892) завершил работу с кодом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
```

C# dasturida funksional yondashuv orqali faktorial hisoblandi. Ushbu dastur avval .NET kompilyatori yordamida kompilyatsiya qilinadi, so'ngra ishga tushiriladi. Kompilyatsiyalangan dastur mustaqil .exe fayl holida ijro etiladi. Bu yondashuv dastur tezligini oshiradi va bajarilish samaradorligini ta'minlaydi.

Xulosa

Dasturlarni bajarilish usullari — bu dasturlash asoslarida muhim o‘rin egallagan mavzulardan biridir. Dastur qanday bajarilishi (kompilyatsiya yoki interpretatsiya orqali) dastur samaradorligi, ishlash tezligi, xatoliklarni aniqlash va platformaviy moslashuvchanlikka bevosita ta'sir ko'rsatadi.

Kompilyatsiya usuli dasturlarni avval to'liq tarjima qilib, tez ishlashiga erishsa, interpretatsiya usuli esa dasturga oson sinov o'tkazish va tez o'zgartirish kiritishga imkon beradi. C#, C++, va Java kabi tillar kompilyator asosida ishlaydi, Python va JavaScript esa interpretatorga asoslanadi. Gibridd yondashuv esa ikkala usulning afzalliklarini uyg'unlashtiradi. Amaliy misollar orqali bu yondashuvlar qanday ishlashi va foydalanuvchiga qanday natija berishi tushunarli tarzda ko'rsatib berildi. Xususan, C# dasturining kompilyatsiya asosida ishlashi misolida dastur samaradorligi, aniq natija va ishlash tartibi yoritildi.

Foydalanilgan adabiyotlar

1. B.Umarov., M.Umarova. THE PROBLEM OF APPROXIMATING SIGNALS BASED ON MODELING OF WAVELET - HAAR TRANSFORMATION. - 2020. - C. 502-506.

2. Azizovich UB. INNOVATION TECHNOLOGIYALAR ORQALI O'QITUVCHILAR KOMETANSIYATINI SHAKLLANTIRISH TASOSIYLARI. Finlyandiya xalqaro ta'lim ilmiy jurnali. Ijtimoiy va gumanitar fanlar. 2023;11(5):823-8.

3. Umarov, B., G'ulomjonova, S. (2024). BULUT TEXNOLOGIYASI VA ULARDAN FOYDALANISH. Zamonaviy ta'limda innovatsion tadqiqotlar, 2(7), 12-14. Qaytadan olindi
4. Karimov O. – *Dasturlash asoslari*, Toshkent: “Fan va texnologiya”, 2020.
5. To‘xtasinov D. – *Algoritmlarni tuzish va tahlil qilish*, Toshkent: “Iqtisod-Moliya”, 2019.
6. Raxmatov A. – *Kompyuter tizimlari va tarmoqlari*, Toshkent: TDYU nashriyoti, 2021.
7. Sattorov X. – *Axborot texnologiyalari va dasturlash*, Samarqand: 2019.
8. Juraev M. – *Kompyuter grafikasi va dasturlash asoslari*, Qarshi: QDU nashriyoti, 2020.
9. Turg'unboyeva S. – *Dasturlash tillari nazariyasi*, Nukus: 2017.
10. Abdukarimov A. – *Algoritmlar va ma'lumotlar tuzilmasi*, Buxoro: BDU, 2021.
11. Yo‘ldoshev A. – *Kompilyatorlar tuzilishi*, Toshkent: TATU, 2020.
12. Umarov B. – *Axborot texnologiyalari va tizimlari*, Toshkent: 2021.
13. O‘rinboyev I. – *Windows operatsion tizimi va dasturlash*, Toshkent: 2019.
14. Norqulov A. – *C# dasturlash asoslari*, Namangan: 2023.
15. Ziyodov T. – *Axborot tizimlari loyihalash asoslari*, Toshkent: 2020.
16. Maxmudov A. – *Algoritmlarni loyihalash va tahlil qilish*, Andijon: 2019.
17. G‘afforov D. – *Axborot xavfsizligi va dasturlash asoslari*, Farg‘ona: 2021.
18. Jo‘rayev N. – *Kompyuter ilmlari va dasturlash*, Qarshi: 2021.
19. Niyozov M. – *Ma'lumotlar bazasi va SQL*, Toshkent: 2020.