



ALGORITMIK TILLARNING RIVOJLANISH BOSQICHLARI VA ULARNING ZAMONAVIY DASTURLASH TILLARIGA TA'SIRI

Amonqilicheva Iroda

Jo'raboyeva Marjona

Termiz davlat pedagogika instituti 2kurs talabalari

Ilmiy raxbar: To'rayev Ro'ziboy Norovich

Termiz davlat pedagogika instituti p.f.f.d. PhD, dots. v.b.

Annotatsiya: Ushbu maqolada algoritmik tillarning shakllanishi, tarixiy rivojlanish bosqichlari hamda ularning zamonaviy dasturlash tillariga ko'rsatgan ta'siri ilmiy-nazariy jihatdan tahlil qilinadi. Tadqiqotda dastlabki algoritmik tushunchalardan boshlab yuqori darajali, obyektga yo'naltirilgan, funksional va parallel dasturlash tillarigacha bo'lgan evolyutsiya jarayoni yoritiladi. Maqola algoritmik tillarning rivojlanish jarayonini chuqur tahlil qilib, ularning zamonaviy dasturlash tillariga ta'siri va amaliy ahamiyatini ilmiy asosda yoritadi.

Kalit so'zlar: algoritmik tillar, dasturlash tillari, algoritm, tarixiy rivojlanish, zamonaviy dasturlash, informatika.

Kirish

Algoritm tushunchasi inson tafakkurining eng muhim mantiqiy mahsullaridan biri hisoblanadi. U muammoni yechishga qaratilgan ketma-ket, aniq va chekli ko'rsatmalar majmuasini ifodalaydi. Informatika fanining asosini tashkil etuvchi algoritmik yondashuv kompyuter texnologiyalarining rivoji bilan yanada muhim ahamiyat kasb etdi. Dasturlash tillari aynan algoritmlarni mashina tushunadigan va bajaradigan shaklga keltirish vositasi sifatida shakllandi. Algoritmik tillarning rivojlanishi bevosita fan-texnika taraqqiyoti, matematik mantiq, hisoblash texnikasi va jamiyat ehtiyojlari bilan bog'liq. Dastlabki davrlarda algoritmlar og'zaki yoki yozma matematik qoidalar sifatida ifodalangan bo'lsa, keyinchalik elektron hisoblash mashinalarining yaratilishi algoritmik tillarning formal, qat'iy sintaksisga



ega bo'lgan ko'rinishlarini talab etdi. Bugungi kunda dasturlash tillari nafaqat hisoblash masalalarini yechish, balki sun'iy intellekt, katta ma'lumotlar (Big Data), veb-texnologiyalar, mobil ilovalar va kiberxavfsizlik sohalarida ham keng qo'llanilmoqda. Python, Java, C++, JavaScript kabi tillar turli algoritmik paradigmalarni o'zida mujassam etgan bo'lib, ularning ildizi tarixiy algoritmik tillarga borib taqaladi. Mazkur maqolaning asosiy maqsadi algoritmik tillarning tarixiy rivojlanish bosqichlarini chuqur tahlil qilish, har bir bosqichning o'ziga xos xususiyatlarini ochib berish hamda ularning zamonaviy dasturlash tillariga ko'rsatgan ta'sirini ilmiy asosda yoritishdan iborat. Tadqiqot natijalari informatika va dasturlash fanlarini o'rganayotgan talabalar hamda tadqiqotchilar uchun nazariy va amaliy ahamiyatga ega bo'lishi kutiladi.

Algoritmik tillarning rivojlanishini tushunish uchun avvalo algoritm va dasturlash tili tushunchalari o'rtasidagi bog'liqlikni aniqlash zarur. Donald E. Knuth algoritmni “chekli qadamlar orqali muammoni yechishga olib boruvchi aniq ko'rsatmalar majmuasi” sifatida ta'riflaydi. Ushbu ta'rif algoritmik tillarning asosiy vazifasi — algoritmni formal va bajariladigan shaklga keltirish ekanligini ko'rsatadi. Dasturlash tillari aynan shu ehtiyojdan kelib chiqib shakllangan. Algoritmik tillarning dastlabki rivojlanish bosqichi matematik mantiq va hisoblash nazariyasi bilan chambarchas bog'liq bo'lgan.[1] Niklaus Wirth ta'kidlaganidek, “dastur algoritmilar va ma'lumotlar tuzilmalari birligidan iborat”. Bu fikr algoritmik tillarning faqat sintaksis emas, balki muammoni yechishning mantiqiy modelini ham ifodalashini anglatadi. Dastlabki algoritmik tillar aynan algoritmni aniq, izchil va tushunarli tarzda yozish muammosini hal etishga qaratilgan edi. 1950–1960-yillarda yuqori darajali algoritmik tillarning paydo bo'lishi informatika tarixida tub burilish yasadi. FORTRAN tilining yaratilishi ilmiy va muhandislik hisob-kitoblarini avtomatlashtirish imkonini berdi.[2] ALGOL tili esa strukturalangan dasturlash g'oyalarining shakllanishiga asos bo'ldi. Edsger Dijkstra bu borada shunday yozadi: “Struktura — bu dastur tushunarligining asosiy sharti”. Ushbu yondashuv algoritmik tillarda shartli operatorlar, sikllar va blokli tuzilmalarning keng



qo'llanilishiga olib keldi. Keyingi bosqichda algoritmik tillar dasturiy ta'minot murakkablashuviga moslashdi. Obyektga yo'naltirilgan dasturlash paradigmasining shakllanishi bunga yaqqol misoldir. Alan Kay Smalltalk tili haqida so'z yuritir ekan, obyektlarni "dasturiy tizimni tashkil etuvchi mustaqil va o'zaro muloqot qiluvchi birliklar" sifatida ta'riflaydi.[3] Bu yondashuv algoritmik tillarda abstraksiya, inkapsulyatsiya va merosxo'rlik kabi tushunchalarning paydo bo'lishiga olib keldi. Natijada algoritmlar endilikda faqat ketma-ket buyruqlar emas, balki obyektlar o'rtasidagi o'zaro ta'sir orqali ifodalanadigan bo'ldi.[4]

Zamonaviy dasturlash tillarida funksional dasturlash g'oyalarining kuchayishi ham algoritmik tillar evolyutsiyasining mantiqiy davomidir. John Backus o'zining mashhur ma'ruzasida an'anaviy dasturlashni "von Neyman uslubiga haddan tashqari bog'langan" deb tanqid qilgan va funksional yondashuvni taklif etgan.[5] Funksional tillarda algoritmlar matematik funksiyalar orqali ifodalanadi, bu esa aniqlik va ishonchlilikni oshiradi. Shu bilan birga, parallel va taqsimlangan hisoblashga mo'ljallangan algoritmik tillar zamonaviy apparat vositalarining rivoji bilan bog'liq holda shakllandi. Ko'p yadroli protsessorlar va bulutli texnologiyalar algoritmlarni parallel bajarish zaruratini yuzaga keltirdi. Bu holat algoritmik tillarning semantik va sintaktik jihatdan yanada boyishiga sabab bo'ldi.

Umuman olganda, algoritmik tillarning rivojlanishi ilmiy fikr, texnologik imkoniyatlar va amaliy ehtiyojlar o'zaro ta'sirining natijasidir.[6] Zamonaviy dasturlash tillarida uchraydigan ko'p paradigmali yondashuvlar — strukturalangan, obyektgga yo'naltirilgan va funksional dasturlash elementlarining birlashuvi — tarixiy algoritmik tillar tajribasiga tayangan holda shakllangan.

Ushbu tadqiqot ilmiy-nazariy xarakterga ega bo'lib, unda asosan sifatli (qualitative) tadqiqot metodlaridan foydalanildi. Tadqiqot jarayonida algoritmik tillar va dasturlash tillari tarixiga oid klassik va zamonaviy ilmiy adabiyotlar chuqur tahlil qilindi.[7] Birinchi navbatda tarixiy-xronologik metod qo'llanilib, algoritmik tillarning shakllanishi va rivojlanishi bosqichma-bosqich o'rganildi. Har bir bosqich muayyan texnologik yutuqlar va ilmiy ehtiyojlar bilan bog'liq holda tahlil qilindi.



Ikkinchi metod sifatida taqqoshlama (comparative) tahlil ishlatildi. Bu usul yordamida past darajali va yuqori darajali algoritmik tillar, shuningdek, tarixiy tillar bilan zamonaviy dasturlash tillari o'rtasidagi umumiy va farqli jihatlar aniqlandi. Sintaksis, semantika va dasturlash paradigmalari asosiy taqqoslash mezonlari sifatida tanlandi. Shuningdek, umumlashtirish va mantiqiy tahlil metodlari yordamida olingan ma'lumotlar tizimlashtirildi hamda ilmiy xulosalar chiqarildi.[8] Tadqiqot natijalari nazariy asoslangan bo'lib, ular amaliy misollar va mavjud ilmiy qarashlar bilan mustahkamlandi.

Algoritmik tillarning rivojlanish bosqichlarini tahlil qilish shuni ko'rsatadiki, ushbu jarayon informatika fanining shakllanishi va taraqqiyoti bilan bevosita bog'liqdir. Algoritmik tillar dastlab oddiy matematik va mantiqiy qoidalarni ifodalash vositasi sifatida paydo bo'lgan bo'lsa, vaqt o'tishi bilan murakkab dasturiy tizimlarni yaratishga xizmat qiluvchi universal vositaga aylandi. Bu evolyutsiya jarayoni fan-texnika yutuqlari, hisoblash texnikasi imkoniyatlarining kengayishi hamda jamiyatning axborotga bo'lgan ehtiyojlari bilan izohlanadi. Tahlillar shuni ko'rsatadiki, algoritmik tillarning har bir tarixiy bosqichi keyingi avlod dasturlash tillari uchun nazariy va amaliy asos bo'lib xizmat qilgan. Mashina va assembler tillari kompyuter arxitekturasini chuqur tushunishga imkon bergan bo'lsa, yuqori darajali algoritmik tillar dasturchi mehnatini yengillashtirib, algoritmik fikrlashni keng ommalashtirdi. Strukturalangan dasturlash g'oyalari dasturiy ta'minotning aniqligi va ishonchliligini oshirdi, obyektga yo'naltirilgan paradigma esa murakkab tizimlarni modellashtirish va boshqarishni osonlashtirdi.

FOYDALANILGAN ADABIYOTLAR

1. Knuth, D. *The Art of Computer Programming*. Addison-Wesley, 1997.
2. Wirth, N. *Algorithms + Data Structures = Programs*. Prentice Hall, 1976.
3. Sebesta, R. W. *Concepts of Programming Languages*. Pearson, 2019.
4. Dijkstra, E. W. *Notes on Structured Programming*. Academic Press, 1972.



5. Backus, J. Can programming be liberated from the von Neumann style? *Communications of the ACM*, 21(8), 613–641, 1978.
6. Kay, A. The early history of Smalltalk. *ACM SIGPLAN Notices*, 28(3), 69–95, 1993.
7. Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D, *Compilers: Principles, Techniques, and Tools*. Pearson, 2006.
8. Sammet, J. E. *Programming Languages: History and Fundamentals*. Prentice Hall, 1969.