

DASTURLASH TILLARINING RIVOJLANISHI VA TAHLILI

To‘xtayeva Robiya Ravshanbek qizi

Andijon davlat universiteti talabasi, tukhtayeva_robija@adu.uz

ARTICLE INFO.

Kalit so‘zlar: Dasturlash tillari, algoritmik tafakkur, kompilatsiya, interpretatsiya, JavaScript, Python, Rust, sun‘iy intellekt, kodlash avtomatizatsiyasi, tarixiy rivojlanish, dasturlash paradigmalari, xavfsizlik, platforma mosligi, O‘zbekiston, raqamli transformatsiya.

Anotatsiya

Ushbu maqolada dasturlash tillarining tarixiy rivojlanish bosqichlari, ularning asosiy paradigmalari va zamonaviy ko‘rinishlari chuqur tahlil qilinadi. Mashina kodlaridan boshlab, sun‘iy intellektga yo‘naltirilgan tillargacha bo‘lgan jarayonlar ketma-ketligi o‘rganiladi. JavaScript, Python, Go, Rust, Kotlin va Swift kabi zamonaviy tillarning afzalliklari, qo‘llanish sohalari va texnik xususiyatlari ilmiy nuqtai nazardan tahlil qilinadi. Maqolada dasturlash tillarini o‘rganishdagi muammolar, kompilatsiya/interpretatsiya jarayonlaridagi cheklovlar, platformaga bog‘liqlik va xavfsizlik masalalari tanqidiy yondashuv asosida bayon etiladi. Kelajakda paydo bo‘lishi kutilayotgan yangi tillar, AI asosida kodlash tizimlari va multidisiplinar dasturlash yondashuvlariga doir prognozlar beriladi. Xulosa qismida esa O‘zbekistonda va global miqyosda dasturlash tillarini o‘rganish va tatbiq etishga doir metodik va strategik tavsiyalar keltirilgan.

<http://www.gospodarkainnowacje.pl/> © 2025 LWAB.

Tadqiqotning maqsadi: Zamonaviy ta‘lim tizimida web dasturlash o‘rganish nafaqat dasturchilar uchun, balki keng tarqalgan boshqa sohalarda ham muhim ahamiyat kasb etmoqda. Internetning kuchayishi va texnologiyalarning tez rivojlanishi bilan birga, web dasturlashning roli ham ortdi. Bugungi kunda web dasturlash nafaqat IT sohasida, balki barcha sohalarda — iqtisodiyot, sog‘liqni saqlash, ta‘lim, marketing va hokazolarda keng qo‘llanilmoqda. Internetga bog‘liq faoliyatlarning har bir turida web dasturlashning amaliy ko‘nikmalari talab qilinadi. Shuning uchun, web dasturlashni o‘rganish nafaqat IT mutaxassislari, balki har qanday soha vakillari uchun zarurdir.

Talabalar uchun web dasturlashni o‘rganish, o‘qish jarayonini yanada zamonaviylashtirishga yordam beradi va ularning kelajakdagi ish faoliyatida ko‘plab imkoniyatlarni ochib beradi. [1] Dasturlash ko‘nikmalariga ega bo‘lgan shaxslar faqatgina web saytlar yaratish bilan cheklanmay, balki murakkab tizimlar, mobil ilovalar, e-tijorat platformalarini ishlab chiqish kabi keng qamrovli sohalarda ham muvaffaqiyatli ishlashlari mumkin. Bunday ko‘nikmalarga ega bo‘lgan mutaxassislar talab etilayotgan sohaga qarab, o‘z faoliyatini yanada kengaytirishlari va ijtimoiy rivojlanishning yangi yo‘nalishlarini yaratishlari mumkin.

Maqolaning maqsadi va vazifalari: Web dasturlashni o‘rgatishda zamonaviy pedagogik yondashuvlar va innovatsion metodlarning o‘rni

Maqolaning asosiy maqsadi, web dasturlashni o‘rgatish jarayonida zamonaviy pedagogik yondashuvlar va innovatsion metodlarning o‘rni va ahamiyatini o‘rganishdir. Bugungi ta‘lim tizimi va texnologiyalar dunyosidagi o‘zgarishlar talabalar o‘qish jarayonini yangilashni va o‘qituvchilarning pedagogik

usullarini rivojlantirishni taqozo etmoqda. O'qitishning samarali bo'lishi uchun an'anaviy metodlardan tashqari, interaktiv, innovatsion va talabalarga moslashtirilgan yondashuvlar qo'llanishi kerak.

Maqolada web dasturlashni o'rgatishda muhim bo'lgan yangi metod va usullar, o'quv jarayoniga o'quvchilarni faol jalb etish va amaliyotga asoslangan yondashuvlar alohida ta'kidlanadi. [2] Web dasturlashni o'rgatishda talabalar nafaqat nazariy bilimlarni o'zlashtirishlari, balki ularni amaliyotda qo'llashni ham o'rganishlari kerak. Shuning uchun, o'qituvchilarni zamonaviy pedagogik usullar bilan tanishtirish va ular yordamida talabalarning o'rganish samaradorligini oshirish juda muhimdir.

Bundan tashqari, maqolada web dasturlashni o'rgatishda innovatsion texnologiyalar va onlayn resurslar, shuningdek, o'quvchilarning o'z-o'zini boshqarish, kreativitlik va muammolarni hal qilish qobiliyatlarini rivojlantirishga qaratilgan metodlar tahlil qilinadi. Maqsadimiz — web dasturlashni o'rgatishdagi pedagogik metodlarni takomillashtirish, talabalarga yanada samarali o'quv tajribasi yaratishdir.

Shunday qilib, maqola web dasturlashni o'rgatishning zamonaviy metodlarini o'rganish va o'quvchilarni ilg'or pedagogik usullar yordamida bilim va ko'nikmalarni o'zlashtirishga undashga qaratilgan.

Dasturlash tillarining tarixiy rivojlanishi kompyuter texnologiyalarining evolyutsiyasi bilan chambarchas bog'liq bo'lib, ularning shakllanishi, soddalashuvi va imkoniyatlarining kengayishi bugungi kunda global axborot tizimlarining asosiy harakatlantiruvchi kuchiga aylangan. Dastlabki bosqichlarda kompyuterlar faqat mashina kodlari – nol va birlar ketma-ketligida buyruq oladigan qurilmalardan iborat edi. Bu holat dasturchidan kompyuter arxitekturasini chuqur bilishni talab qilgan va dasturlash jarayonini juda murakkab va ko'p mehnat talab qiluvchi vazifaga aylantirgan. Bu davr birinchi avlod dasturlash tillari – mashina tillari davri hisoblanadi.

Keyingi bosqichda inson tiliga yaqinlashtirilgan, ammo hali ham apparatga bog'liq bo'lgan assambler tillari yuzaga keldi. Bu ikkinchi avlod tillari deb yuritiladi. Assambler tillari dasturchiga mnemonik belgilar orqali buyruqlar berishga imkon yaratdi, biroq ular hali ham past darajali bo'lib, konkret kompyuter arxitekturasiga moslashtirilgan edi. Bu tillarning afzalligi – bajarilish tezligi va resurslar bilan bevosita ishlay olishidir, ammo yuqori darajadagi tushunarlilik darajasi mavjud emas edi.

Dasturlashning yangi bosqichi sifatida uchinchi avlod tillari – yuqori darajadagi dasturlash tillari rivojlandi. Ushbu bosqichda FORTRAN (ilmiy hisob-kitoblar uchun), COBOL (biznes tizimlari uchun), va Pascal (ta'limiy va umumiy dasturlash uchun) kabi tillar yaratilgan bo'lib, ular dasturchilar orasida keng qo'llanila boshlandi. Bu tillar orqali dastur yozish ancha qulay, tahrirlash esa osonlashgan edi. Sintaksislar inson mantiqiga yaqinlashtirilgani sababli, ular kompyuter dasturlashini keng ommaga yetkazish imkonini berdi.

To'rtinchi avlod dasturlash tillari esa amaliy ma'lumotlar ustida ishlashga yo'naltirilgan bo'lib, ular orasida SQL (strukturaviy so'rovlar tili) va MATLAB (matematik modellash uchun) alohida o'rin tutadi. Ushbu tillar yordamida foydalanuvchilar dastur tuzmasdan turib ham muayyan vazifalarni bajarishlari mumkin bo'ldi, ya'ni bu tillar foydalanuvchi bilan dastur o'rtasidagi masofani yanada qisqartirdi.

So'nggi, beshinchi avlod dasturlash tillari sun'iy intellekt, mashinali o'rganish, statistik tahlil va katta ma'lumotlar bilan ishlash kabi murakkab sohalarda uchun yaratilgan. [3] Ushbu avlod vakillari sifatida Prolog, Python va R tillari ko'rsatiladi. Prolog sun'iy intellektda, Python esa o'zining universal imkoniyatlari, oson sintaksisi va kuchli kutubxonalari tufayli eng ommabop tillardan biriga aylangan. R tili esa statistik tahlil va ma'lumotlarni vizualizatsiya qilishda yetakchi o'rin tutadi.

Umuman olganda, dasturlash tillari taraqqiyoti inson-kompyuter o'zaro aloqasini soddalashtirish, dasturlash samaradorligini oshirish va yangi sohalarda qo'llanish doirasini kengaytirish sari yo'naltirilgan. Har bir avlodning paydo bo'lishi – mavjud ehtiyojlar, texnologik imkoniyatlar va inson

tafakkurining texnik ifodasi o'rtasidagi uyg'unlik mahsuli hisoblanadi.

Zamonaviy dasturlash tillari texnologik muhitning talablariga mos ravishda shakllanib, har birining o'ziga xos sintaksis, ishlash prinsipi va amaliy qo'llanilish sohasi mavjud. Bu tillar dasturlash samaradorligini oshirish, tizimlararo moslikni ta'minlash, xavfsizlik va barqarorlikni kuchaytirish, shuningdek, foydalanuvchi uchun intuitiv interfeyslar yaratish kabi bir qator ustunliklarga ega. Hozirgi davrda keng qo'llanilayotgan dasturlash tillari orasida JavaScript, Python, Go, Rust, Kotlin va Swift tillari alohida o'rin tutadi.

JavaScript bugungi kunda veb dasturlashning asosiy tili sifatida tanilgan bo'lib, u foydalanuvchi interfeyslarini interaktiv qilishda, brauzer muhitida ishlovchi ilovalar yaratishda va front-end hamda back-end (Node.js orqali) dasturlashda keng qo'llaniladi. Bu tilning asosiy afzalliklari – sintaksisining soddaligi, keng kutubxonalar to'plami, tez ishlovchanligi va kross-platformaligi hisoblanadi. JavaScript doimiy rivojlanishda bo'lib, React, Vue, va Angular kabi zamonaviy ramkalar orqali ishlab chiqish jarayonlarini sezilarli darajada yengillashtiradi.

Python esa o'zining o'qilishi oson sintaksisi, kuchli kutubxonalar bazasi va ko'p maqsadli qo'llanishi bilan ajralib turadi. U sun'iy intellekt, mashinali o'rganish, ma'lumotlar tahlili, veb dasturlash va avtomatlashtirishda yetakchi tilga aylangan. NumPy, Pandas, TensorFlow va Django kabi kutubxonalar Python dasturlovchilariga yuqori samaradorlik bilan murakkab masalalarni hal qilish imkonini beradi. Pythonning interpretatsiyalanuvchi til bo'lishi dastur kodining platformalardan mustaqil ishlashini ta'minlaydi.

Go (yoki Golang) tili Google tomonidan ishlab chiqilgan va ko'proq tizimli dasturlash, bulutli xizmatlar, mikroxizmatlar arxitekturasini va tarmoq dasturlari yaratishda qo'llaniladi. U kompilyatsiya tezligi, parallellikni qo'llab-quvvatlashi va kam xotira sarfi bilan ajralib turadi. Go tili sintaksis jihatidan soddalik va qat'iylikka asoslangan bo'lib, keng ko'lamli dasturlarni barqaror va tez ishlab chiqishga moslashtirilgan.

Rust tili xavfsizlik va samaradorlikni birinchi o'ringa qo'yuvchi tizimli dasturlash tili bo'lib, ayniqsa resurslar bilan bevosita ishlovchi past darajadagi dasturlarda qo'llaniladi. U C va C++ tillariga muqobil sifatida paydo bo'lgan bo'lib, xotira xavfsizligini kompilyatsiya vaqtida ta'minlaydi. Rust bugungi kunda brauzer dvijoklari (masalan, Mozilla tomonidan yaratilgan Servo), o'yinlar va tizimlar dasturlashida faol foydalaniladi. Uning zamonaviy sintaksisi va qat'iy xatoliklar nazorati dastur ishonchliligini oshiradi.

Kotlin — Android dasturlashda Java tiliga zamonaviy alternativ sifatida keng tarqalgan bo'lib, JetBrains kompaniyasi tomonidan ishlab chiqilgan. U Java virtual mashinasida ishlaydi, sintaksisi qisqa va tushunarli, shuningdek, orqaga moslikni saqlaydi. Kotlin null xavfsizligini avtomatik tarzda nazorat qilishi, zamonaviy dasturlash konseptlarini qo'llab-quvvatlashi bilan Android ilovalarini ishlab chiqishda samarali vositadir. Bugungi kunda u Google tomonidan Android dasturlash uchun rasmiy tavsiya etilgan tildir.

Swift esa Apple ekotizimi uchun yaratilgan dasturlash tili bo'lib, iOS, macOS, watchOS va tvOS ilovalarini ishlab chiqishda qo'llaniladi. U xavfsizlik, yuqori ishlash tezligi, sintaktik qulaylik va kuchli statik tip tizimi bilan ajralib turadi. Swift yordamida yaratilgan ilovalar barqarorligi, tezligi va foydalanuvchi tajribasi bilan yuqori baholanadi. Shuningdek, u Xcode muhitida ishlab chiqish uchun to'liq integratsiyalangan.

Shu tarzda, har bir zamonaviy dasturlash tili muayyan sohada samarali bo'lishi, foydalanish qulayligi va texnik imkoniyatlari bilan tanlanadi. Tillar orasidagi bu xilma-xillik texnologik yechimlarni optimallashtirish, dasturchilarga tanlov erkinligini taqdim etish va dasturlash sohasida innovatsion yondashuvlarni amalga oshirish imkonini beradi.

Dasturlash tillarining rivojlanishi axborot texnologiyalari taraqqiyotiga sezilarli ta'sir ko'rsatgan bo'lsa-

da, ularning qo'llanilishi jarayonida yuzaga keladigan bir qator muammolar va cheklovlar mavjud bo'lib, bu jihatlar dasturiy ta'minot ishlab chiqish samaradorligiga bevosita ta'sir ko'rsatadi. Tanqidiy tahlillar shuni ko'rsatadiki, dasturlash tillarining ko'pchiligi o'zining o'ziga xos murakkab sintaksisi, resurslar bilan ishlashdagi cheklovlar va xavfsizlik jihatlaridagi zaifliklar tufayli dasturchilar oldiga turli murakkabliklarni keltirib chiqaradi.

Eng avvalo, o'rganish murakkabligi muhim muammo sifatida ajralib turadi. Dasturlashni endi boshlayotgan foydalanuvchilar uchun ayrim tillarning sintaktik qoidalari, kompilyatsiya muhiti va dasturlash paradigmalarini tushunarsiz bo'lishi mumkin. Ayniqsa, C++, Java, Rust kabi tillar chuqur texnik bilim va qat'iy sintaktik intizom talab qiladi. Bu holat esa o'quvchilarning motivatsiyasini pasaytiradi va dasturlashga bo'lgan qiziqishni sustlashtiradi. Hattoki yuqori darajadagi tillarda ham, masalan Python yoki JavaScriptda, murakkab kutubxonalar va asinxron jarayonlar bilan ishlash hamisha ham oson emas.

Bundan tashqari, dasturlash tillarining kompilyatsiya yoki interpretatsiya tezligi ham muhim cheklovlardan biri hisoblanadi. Kompilyatsiya bosqichida vaqt sarfi ko'p bo'lgan tillar (masalan, C++ yoki Swift) tezkor prototiplash jarayonlari uchun mos kelmasligi mumkin. Aksincha, interpretatsiya asosida ishlaydigan tillar (masalan, Python yoki JavaScript) o'zining qulayligi bilan ajralib tursa-da, bajarilish tezligi va samaradorligi nuqtai nazaridan resurs talabchan muhitlarda sust ishlashi mumkin. Bu holat ayniqsa real vaqtda ishlovchi tizimlar, o'yinlar yoki og'ir hisob-kitoblar talab qilinadigan ilovalarda sezilarli farq qiladi.

Shuningdek, platformaga bog'liqlik va xavfsizlik masalalari ham dolzarb muammolardan biridir. Ba'zi dasturlash tillari faqat muayyan operatsion tizimlar yoki apparat arxitekturasida to'liq ishlashi mumkin. Masalan, Swift faqat Apple platformalarida to'liq qo'llab-quvvatlanadi, Android uchun yozilgan Kotlin esa boshqa muhitlarda cheklovlarga duch keladi. Bu esa ishlab chiquvchilardan ko'proq platformaga moslashtirish yoki kross-platforma yechimlarini qidirishni talab qiladi. Xavfsizlik jihatidan esa xotira boshqaruvi, ma'lumotlar uzatishdagi zaifliklar va noto'g'ri yozilgan kod tufayli yuzaga keluvchi ekspluatatsiya xatarlari muammolarni kuchaytiradi. Ayniqsa, til darajasida xatoliklarni aniqlash imkoniyati bo'lmagan tillarda bu masalalar jiddiy tus oladi.

Shu sababli, zamonaviy dasturlash tillarini tanlashda nafaqat ularning imkoniyatlari, balki ularning cheklovlari, o'rganishga ketadigan vaqt, xavfsizlik siyosati va ishlash samaradorligi ham chuqur tahlil qilinishi zarur. Ushbu omillarni hisobga olish, dasturiy mahsulotning sifatli, xavfsiz va barqaror ishlashini ta'minlashda muhim ahamiyatga ega.

Dasturlash tillarining evolyutsiyasi nafaqat mavjud ehtiyojlarga javob berish, balki kelajakdagi texnologik tendensiyalarni shakllantirish yo'lida davom etmoqda. Zamonaviy ilm-fan va sanoatning tezkor rivojlanishi natijasida yangi avlod dasturlash tillariga ehtiyoj ortib bormoqda. Bugungi dasturchilar oldida nafaqat samarali, balki ko'p sohalarida integratsiyalashgan, xavfsiz va o'rganishga yengil tillarni yaratish masalasi turibdi. Bu jarayon zamonaviy texnologiyalar, xususan sun'iy intellekt (AI) va mashinali o'rganish (ML) bilan uyg'unlashib, dasturlash tillarining yangi bosqichiga zamin yaratmoqda.

So'nggi yillarda paydo bo'lgan Zig, Elixir, Crystal kabi yangi dasturlash tillari mavjud tillarning zaif tomonlarini bartaraf etish maqsadida yaratilmoqda. Zig tili, masalan, C tiliga muqobil sifatida taklif etilgan bo'lib, yuqori darajadagi nazorat va aniqlikni saqlagan holda, sintaktik aniqlik, xotira xavfsizligi va modulli tuzilmani ta'minlaydi. [4] Elixir esa Erlang virtual mashinasi asosida yaratilgan bo'lib, ko'p yadroli tizimlar va real vaqtda ishlovchi dasturlar uchun yuqori samaradorlikni ta'minlaydi. Bu kabi tillar asosan tizim darajasidagi ishonchli va kengaytiriluvchi ilovalarni ishlab chiqishda foydalidir.

Kelajakda sun'iy intellekt va mashinali o'rganish sohasining kengayib borishi bilan, kod yozish jarayonining o'zi ham avtomatlashtirilmoqda. AI asosida ishlaydigan kod generatorlar, natural tilni kodga aylantiruvchi tizimlar (masalan, GitHub Copilot, OpenAI Codex) dasturchilarning ishini

soddalashtirib, tezkor prototiplash va kod takrorlashlaridan qochishga yordam bermoqda. Ushbu tendensiya yangi turdagi dasturlash interfeyslarini shakllantiradi: bunda dasturchi an'anaviy kod yozish o'rniga, natural tilda topshiriq beradi va tizim u uchun kod yozadi. Kelajakda bu yo'nalish kuchayib, AI yordamida o'z-o'zini optimallashtiruvchi, kodda xatolikni aniqlab tuzatuvchi va muhitga mos ravishda kod taklif qiluvchi tillar yuzaga kelishi kutilmoqda.

Dasturlash tillarining tarixiy va funksional rivoji axborot texnologiyalari sohasining tub o'zgarishlarga uchrashi, yangi innovatsion yechimlarning yaratilishi va inson faoliyatining deyarli barcha jabhalariga raqamli vositalar kirib kelishiga sabab bo'ldi. Har bir avlod dasturlash tillari o'z davridagi texnologik muammolarga javob sifatida paydo bo'lgan bo'lsa, zamonaviy dasturlash tillari esa ko'p platformalilik, xavfsizlik, ko'p foydalanuvchilik va avtomatlashtirish kabi kompleks talablarni qondirishga qaratilgan. Hozirgi sharoitda dasturlash tillari faqat texnik vosita emas, balki ilm-fan, ta'lim, iqtisodiyot, sog'liqni saqlash va boshqa ko'plab sohalarning raqamli asosini tashkil qilmoqda.

Shu nuqtai nazardan, O'zbekiston sharoitida dasturlash tillarini o'rganish va ularni amaliyotga tadbiq etish masalasi dolzarb bo'lib qolmoqda. Maktab va oliy ta'lim tizimida algoritmik tafakkurni shakllantirishga alohida e'tibor qaratilishi, dasturlashga oid darsliklar va metodik qo'llanmalar zamonaviy tillarga mos tarzda qayta ishlab chiqilishi zarur. Shuningdek, amaliy loyihalar, hackathonlar, ochiq kodli dasturlar ustida ishlash orqali yoshlarning real muammolarni yechishga yo'naltirilgan bilim va ko'nikmalarini rivojlantirish mumkin. Shu bilan birga, xorijiy tajriba asosida raqamli platformalar, masofaviy dasturlash kurslari va xalqaro sertifikatlashga tayyorlov dasturlarini joriy etish muhim ahamiyatga ega.

Dunyo miqyosida esa dasturlash tillarining kelajagi sun'iy intellekt, avtomatlashtirilgan kodlash, ko'p tillilik va soha integratsiyasi sari yo'nalmog'da. Shuning uchun, O'zbekistonda ham dasturlash sohasini global tendensiyalar bilan uyg'unlashtirish, zamonaviy texnologiyalarni o'zlashtirish va xalqaro hamkorlikda ishlashga tayyor mutaxassislar yetishtirish strategik zaruratdir. Bu esa o'z navbatida mamlakat raqamli transformatsiyasining asosiy tayanchi bo'lib xizmat qiladi.

Foydalanilgan adabiyotlar:

1. Xu, P. (2018). *Analysis of Programming Language and Software Development by Computer*. Proceedings of the 8th International Conference on Social Network, Communication and Education (SNCE 2018), 1196–1199. Atlantis Press.
2. *A Comparative Analysis of Programming Language Preferences Among Computer Science and Non-Computer Science Students*. ResearchGate.
3. Basyuk, T., & Vasyliuk, A. (2024). Development of the Programming Language Learning Skills Using Gamification Elements. In *Proceedings of the 3rd International Workshop on Social Communication and Information Activity in Digital Humanities (SCIA 2024)*, Lviv, Ukraine, October 31, 2024. CEUR Workshop Proceedings, Vol. 3851.
4. Laxminarayana, G. S., Satyanarana, G. S., Shaikh, S. A., & Ansari, A. M. S. (2024). The Impact of Programming Language on Development Productivity: An Empirical Study. *International Research Journal of Modernization in Engineering, Technology and Science*, 6(7).