

DASTURLASH TILLARINING TASNIFI

Shodimurodov Ulug'bek Akmalovich

*Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti
Nurafshon filiali, talaba*

Ilg'orova Dilnoza Ilg'or qizi

*Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti
Nurafshon filiali, talaba*

Annotasiya: *So'nggi bir necha o'n yilliklarda axborot va kompyuter texnologiyalari sohasi oldinga katta qadam tashladi. Texnologiyalar bizning kundalik hayotimizga mustahkam o'rnashib oldi va zamonaviy odamni kompyuterda hech qanday ma'lumotni qayta ishlamasdan tasavvur qilish allaqachon juda qiyin. Dasturlash, albatta, texnik taraqqiyot bilan yonma-yon boradi va boshqa texnologiyalar kabi tez rivojlanadi. Eski dasturlash tillari takkomillashtirilmoqda va yangilari ishlab chiqilmoqda. Elektron kompyuterlar rivojlanishining butun tarixi davomida yuzlab dasturlash tillari paydo bo'ldi, ularning ko'pchiligi bugungi kunda ham qo'llanilmoqda. Dasturlash tillarini tasniflash zarur. Hozirgi vaqtda individual xususiyatlarga asoslangan bir nechta turli tasniflar mavjud va har bir muallif ularni o'ziga xos tarzda talqin qiladi, tillarni sinflarga o'ziga xos tarzda taqsimlaydi.*

Kalit so'zlar: *dasturlash, dasturlash tillari, tasniflash, dasturlash tillarining darajalari, dasturlash tillarining avlodlari, dasturlash paradigmalari.*

Avvalo, dasturlash tillarini darajalarga ajratish kerak. Turli mualliflar dasturlash tillarni turli darajalarga ajratadilar. Uch darajaga (mashina, mashinaga yo'naltirilgan va mashinadan mustaqil tillar), yoki ikki darajadagi (pastki darajali dasturlash tillari, yuqori darajali dasturlash tillari). Agar dasturlash tillari uchta darajaga bo'lingan bo'lsa, eng past daraja mashina kodi bo'ladi. Ma'lum bir protsessor tomonidan bajariladigan va u uchun maxsus ishlab chiqilgan ko'rsatmalar to'plami. Odatda o'n oltilik belgilar ketma-ketligidan iborat bo'ladi. "Salom, dunyo!" qatorini chop etish x86 protsessori uchun quyidagicha ko'rinadi: BB 11 01B9 0D 00 B4 0E 8A 07 43 CD 10 E2 F9 CD 20 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 21. O'rta darajada esa mashina tili bo'ladi, ya'ni, protsessor buyruqlarini to'g'ridan-to'g'ri boshqarish uchun mo'ljallangan, lekin inson idroki uchun qulayroq tillar. Masalan, assemblya tillari. Assemblya tili mohiyatan har bir mashina kodi buyrug'larini odamlar uchun tushunarli bo'lgan ramziy buyruqlar mnemonika yordamida ifodalaydi. Odatda, assembler tili ma'lum bir protsessor oilasining xususiyatlaridan foydalanadi. Eng yuqori daraja, mashinadan mustaqil tillar hisoblanadi. Ular tushunish, tushunish va ishlash oson bo'lishi uchun yaratilgan. Ushbu dasturlash tillarining o'ziga xos xususiyati abstraksiya, ya'ni ma'lumotlar va ular ustidagi operatsiyalarni qisqacha tavsiflovchi semantik konstruktsiyalarni kiritishdir, ularning tavsifi mashinaga yo'naltirilgan tillardan juda uzoq va tushunish qiyin emas. Ular, shuningdek,

algoritmlar mohiyatining platforma mustaqilligini ta'minlash uchun ishlab chiqilgan. Ularning paydo bo'lishi bilan platformaga bog'liqlik tarjimonlarga o'tadi, ular yuqori darajadagi tilda yozilgan matnni elementar mashina buyruqlariga "tarjima qiladilar". Yuqori darajadagi dasturlash tillariga misol qilib C++, C#, PHP, Perl, Java va boshqalarni keltirish mumkin.

Dasturlash tillarini daraja bo'yicha zamonaviy tasniflashda ikki darajaga bo'linish tobora ko'proq qo'llanilmoqda: yuqori va past darajali tillar. Past darajadagi tillar - bu mashina kodiga yaqin bo'lgan va diqqat markazida bo'lgan tillar, maxsus protsessor ko'rsatmalari, ya'ni mashinaga yo'naltirilgan tillar. Yuqori darajali tillar- bu zamonaviy tasnifdagi, foydalanish qulayligi uchun ishlab chiqilgan mashinadan mustaqil tillar hisoblanadi. Shunday qilib, hozirgi vaqtda mualliflar ko'pincha o'z tasniflarida mashina kodini hisobga olmaydilar. Buning sababi shundaki, hozirgi vaqtda mashina kodi tor sohadagi mutaxassislariga kerak bo'lishi mumkin, masalan, tizim dasturlash, axborot xavfsizligi yoki qurilmalarni dasturlash. Amaliy dasturchi o'z dasturi qanday saqlanishi va bajarilishini tushunish uchun mashina kodini tushunishi va ba'zi hollarda dasturlarini disk raskadrovka qilish va optimallashtirish uchun kerak bo'lishi mumkin.

Ko'rib chiqilishi kerak bo'lgan keyingi tasnif bu dasturlash tillarini avlodlar bo'yicha tasniflanishidir. Tillarni odatda besh avlodga bo'lishadi, ammo turli mualliflar yana avlodlarni o'zlariga ko'ra ajratadilar. Ba'zi tasniflash variantlari darajalar bo'yicha tasniflash bilan umumiy narsaga ega, ba'zilari beshinchi avlod dasturlash tillari hali mavjud emasligini ko'rsatadi.

Mualliflarning ko'pchiligi birinchi avlod tillari - mashina kodlari, ya'ni past darajadagi tillar, ikkinchi avlod tillari - biz allaqachon muhokama qilgan asambleya tillari va shunga mos ravishda uchinchi avlod tillari - yuqori darajadagi tillar. Evolyutsion asosda bu tasnif juda mantiqiy: past darajadagi tillar dasturlashning boshida paydo bo'lgan, o'rta va yuqori darajali tillar esa kompyuter texnologiyalarining o'sib borayotgan imkoniyatlari bilan bog'liq holda ishlab chiqilgan va takomillashtirilgan. Ammo to'rtinchi va beshinchi avlod tillarini tasniflash unchalik oson emas. To'rtinchi avlod dasturlash tillari - ko'pincha ob'ektga yo'naltirilgan, deklarativ (protsessual bo'lmagan) va vizual tillarni o'z ichiga oladi. Ko'pincha bu so'rov tillarini o'z ichiga oladi, masalan, SQL, o'z mohiyatiga ko'ra umuman dasturlash tili emas. Umuman olganda, to'rtinchi avlod dasturlash tillari ma'lum bir dasturiy ta'minot qobig'iga o'rnatilgan va yuqori ixtisoslashgan vazifalar uchun, masalan, ma'lumotlar bazalarini yaratish va boshqarish uchun ishlatiladigan tillar ekanligini tez-tez o'qishingiz mumkin (ma'lumotlar bazasini boshqarish tizimlarining o'rnatilgan tillari). Ma'lum bo'lishicha, to'rtinchi avlod dasturlash tillari uchinchi avlod tillarining mantiqiy rivojlanishi, bu tillar asosan o'zlarining dasturiy ta'minot qobig'iga o'rnatilganligi sababli dasturchilarning ishini soddalashtiradi.. Shuningdek, to'rtinchi avlod tillari ko'p protsessorli arxitektura uchun dasturiy ta'minot yaratishga qaratilgan parallel dasturlash tillarini o'z ichiga olishi kerak. Beshinchi avlod tillari - nazariy jihatdan, dasturchilar ishini yanada soddalashtirishi va tezlashtirishi kerak: bu tillarning maqsadi ishning muhim qismini odamlardan kompyuterlarga o'tkazishdir.

Ko'pincha bu ekspert tizimlari va bilim bazalarining tillarini o'z ichiga oladi. Hozirgi vaqtda haqiqatan ham faqat topshiriqni olgan holda to'liq mahsulot ishlab chiqaradigan yechimlar mavjud emas. Ba'zi mualliflar MathCAD tizimi haqida gapirishadi: u haqiqatan ham foydalanuvchilarning ishini ancha osonlashtiradi va hisoblash ishlarining katta qismini oladi. Biroq, hozircha beshinchi avlod tillarini qanday mezonlar bo'yicha aniqlashning aniq ta'rifi yo'q. Bir narsa aniq: ulardan foydalanib, kompyuter foydalanuvchi algoritmi qo'lda amalga oshirmasdan muammoni hal qilishi kerak bo'ladi.

Keyinchalik, biz dasturlash tillarining funktsional xususiyatlari yoki dasturlash paradigmasi bo'yicha tasnifini ko'rib chiqamiz. Paradigma - bu dasturlashga yondashuvni belgilaydigan g'oyalar va tushunchalar to'plami. Ko'pchilik mualliflar dasturlash tillarini protsessual (imperativ) va protsessual bo'lmagan (deklarativ) ga aniq ajratadilar. Protessual bo'lmagan tillar sinfi ikkita kichik sinfni o'z ichiga oladi: funktsional va mantiqiy tillar. Ko'pincha ob'ektga asoslangan alohida sinf ajratiladi.

Yo'naltirilgan dasturlash tillari, garchi ular protsessual dasturlash elementlarini o'z ichiga oladi. Keling, har bir sinfni batafsil ko'rib chiqaylik.

Protessual tillarda dastur bajarilishi kerak bo'lgan amallarni aniq tasvirlab beradi, ya'ni bu kompyuter bajarishi kerak bo'lgan aniq buyruqlar ketma-ketligidir. Aytishimiz mumkinki, bu tillarda dasturlash orqali dasturchi kompyuterga nima qilishni emas, balki qanday qilishni aytadi. Bu sinfga ko'pgina dasturlash tillari kiradi: Pascal, Basic, C va boshqalar.

Protessual bo'lmagan tillarni osonlikcha protsessual tillarga qarama-qarshi deb atash mumkin. Deklarativ dasturlash - bu kompyuter nima qilish kerakligini tavsiflovchi dasturlash paradigmasi. Yaxshi misol, HTML belgilash tili, keling, tegini olaylik: siz shunchaki undagi rasmga havolani qo'shishingiz kerak va uning sahifada qanday ko'rsatilishi kompyuterining vazifasidir.

Funktsional dasturlashda hisoblash jarayonini matematik ma'noda funktsiyalar qiymatlarini hisoblash deb ta'riflash mumkin. Amalda, matematik funktsiyaning imperativ dasturlashdagi "funksiya" tushunchasidan farqi shundaki, imperativ funktsiyalar nafaqat argumentlarga, balki funktsiyadan tashqaridagi o'zgaruvchilarning holatiga ham tayanishi mumkin va shu tariqa turli joylarda olinadi. Funktsional tilda funktsiyani bir xil argumentlar bilan chaqirganda, biz doimo bir xil natijaga

erishamiz- chiqish faqat kirishga bog'liq. Bu funktsional dasturlash muhitlariga funktsiyalarni bajarish natijalarini keshlash va ularni kerakli vaqtda chaqirish imkonini beradi. Masalan, LISP - birinchi funktsional til, Haskell va boshqalar.

Mantiqiy dasturlash tillarida yozilgan dasturlar harakatlarni umuman tasvirlamaydi. Dasturchi ma'lumotlar va ular orasidagi munosabatlarni o'rnatadi, shundan so'ng dasturga savol berilishi mumkin, kompyuter berilgan ma'lumotlarni saralaydi va javobni topadi. Mantiqiy dasturlash tilining klassik namunasi Prologdir.

Ob'ektga yo'naltirilgan dasturlash tillarida o'zgaruvchilar va funktsiyalar sinflarga guruhlanadi va shu bilan dastur tuzilmasining yuqori darajasiga erishiladi. Sinflardan olingan ob'ektlar bir-birining usullarini (funktsiyalarini) chaqiradi va shu

bilan xususiyatlar (o'zgaruvchilar) holatini o'zgartiradi. Rasmiy tomondan, ob'ektga yo'naltirilgan dasturlash protsessual modelga asoslanadi, ammo mazmunli

tomoni - u funktsiyaga emas, balki integral tizim sifatida ob'ektga asoslanadi.

Maqolada dasturlash tillarining uchta asosiy va eng mashhur tasnifi muhokama qilinadi. Hozirgi vaqtda dasturlash tillarining barcha xususiyatlarini o'z ichiga olgan va turli tasniflarning bir-biri bilan "o'zaro bog'liqligi" haqida gapiradigan aniq yoki aniq tasnifi yo'q. Tabiiyki, ushbu mavzuni ushbu maqolaga qaraganda ancha kengroq o'rganish mumkin.

FOYDALANILGAN ADABIYOTLAR:

1. Dasturlash tillarining tasnifi [Elektron [http://b ourab ai.ru/alg/classification.htm](http://b.ourab.ai.ru/alg/classification.htm)]
2. Dasturlash tillarining besh avlodi [Elektron resurs]. URL: [http://life-prog.ru/view zam2.php?id=194&cat=5&page=11](http://life-prog.ru/view_zam2.php?id=194&cat=5&page=11)
3. Dasturlash tillarining evolyutsiyasi [Elektron resurs]. URL: http://www.urtt.ru/bib/dataindex/oaip/lecture/html/lect_05.htm
4. Dasturlash tillari va ularning tasnifi [Elektron resurs]. URL: <http://www.maksakov-sa.ru/ProgrProd/YazProgr/index.html>
5. Dasturlash tili. Dasturlash tillarining evolyutsiyasi [Elektron resurs]. URL: http://wiki.mvtom.ru/index.php/Programming_language