MAPLEDA MA'LUMOT TIPLARI TIZIMI VA UNING OBYEKTGA YO'NALTIRILGAN DASTURLASH BILAN BOG'LIQLIGI

Sharofutdinov Iqboljon Usmonjon o'g'li
Farg'ona davlat universiteti
Katta o'qituvchisi p.f.f.d (PhD)
iqbol0766@gmail.com
Saidjonova Gulasal Ahmadjon qizi
Farg'ona davlat universiteti talabasi
fbffbf475@gmail.com

ANNOTATSIYA: *Ushbu maqolada Maple kompyuter algebrasi tizimida mavjud bo'lgan ma'lumot tiplari tizimi va uning obyektga yo'naltirilgan dasturlash (OYD) paradigmasi bilan bog'liqligi ilmiy jihatdan tahlil qilingan. Maqolada Maple tilining asosiy ma'lumot tiplari, ularning ierarxiyasi, dinamik tiplash mexanizmi, shuningdek, Maple modul tizimi orqali obyektga yo'naltirilgan yondashuvning amalga oshirilishi ko'rib chiqilgan. Tadqiqot natijasida Maple tizimining an'anaviy OYD tillaridan farqli xususiyatlari va o'xshash tomonlari aniqlangan, amaliy misollar orqali tushuntirilgan. Maqola matematika va informatika yo'nalishidagi oliy ta'lim muassasalari talabalari, magistrantlari va ilmiy-tadqiqot ishlarini olib borayotgan mutaxassislar uchun mo'ljallangan.*

Kalit so'zlar: *Maple, ma'lumot tiplari, obyektga yo'naltirilgan dasturlash, dinamik tiplash, modul, klass, inkapsulatsiya, polimorfizm, merosxo'rlik, kompyuter algebrasi.*

АННОТАЦИЯ: *В данной статье проводится научный анализ системы типов данных в системе компьютерной алгебры Maple и её связи с парадигмой объектно-ориентированного программирования (ООП). Рассмотрены основные типы данных языка Maple, их иерархия, механизм динамической типизации, а также реализация объектно-ориентированного подхода через систему модулей Maple. В результате исследования выявлены особенности и*

сходства системы Maple с традиционными объектно-ориентированными языками, подкреплённые практическими примерами. Статья предназначена для студентов, магистрантов и специалистов, занимающихся научными исследованиями в области математики и информатики.

Ключевые слова: *Maple, типы данных, объектно-ориентированное программирование, динамическая типизация, модуль, класс, инкапсуляция, полиморфизм, наследование, компьютерная алгебра.*

ANNOTATION: *This article presents a scientific analysis of the data type system in the Maple computer algebra system and its relationship with the object-oriented programming (OOP) paradigm. The main data types of the Maple language, their hierarchy, dynamic typing mechanism, as well as the implementation of object-oriented approach through the Maple module system are examined. As a result of the research, the distinctive features and similarities of the Maple system compared to traditional OOP languages are identified and illustrated through practical examples. The article is intended for undergraduate and graduate students, as well as researchers in the fields of mathematics and computer science.*

Keywords: *Maple, data types, object-oriented programming, dynamic typing, module, class, encapsulation, polymorphism, inheritance, computer algebra system.*

KIRISH

Zamonaviy ilm-fan va texnologiyaning jadal rivojlanishi matematikaviy hisoblashlar va dasturlash tizimlarining integratsiyasini yangi bosqichga olib chiqdi. Shunday tizimlar ichida Maple — Kanadadagi Waterloo universiteti tomonidan yaratilgan va hozirda Maplesoft kompaniyasi tomonidan qo'llab-quvvatlanadigan kompyuter algebrasi tizimi — alohida o'rin tutadi. Maple tizimi dastlab 1980-yillarning boshida analitik hisoblashlarni avtomatlashtirish maqsadida yaratilgan bo'lsa-da, keyinchalik to'laqonli dasturlash muhitiga aylangan [1].

Har qanday dasturlash tilining asosini ma'lumot tiplari tizimi tashkil etadi. Ma'lumot tipi — bu dasturda ishlatiladigan qiymatlar to'plami va ular ustida bajariladigan amallar majmui sifatida ta'riflanadi [2]. Ma'lumot tiplari tizimini

chuqur tushunish dasturchilar uchun nafaqat samarali kod yozishga, balki dasturning ichki mantiqini to'g'ri modellashtirshiga ham imkon beradi.

Obyektga yo'naltirilgan dasturlash (OYD) — hozirgi kunda eng keng tarqalgan dasturlash paradigmalaridan biri bo'lib, u real dunyodagi ob'ektlarni dasturiy modellashtirish g'oyasiga asoslanadi. Alan Kay, Bjarne Stroustrup, Grady Booch va boshqa olimlarning asarlarida OYD ning asosiy tamoyillari — inkapsulatsiya, merosxo'rlik va polimorfizm — batafsil yoritilgan [3]. Java, C++, Python kabi tillar ushbu paradigmani to'liq qo'llab-quvvatlaydi.

Maple esa an'anaviy ma'noda sof OYD tili emas. Biroq, Maple 9-versiyasidan boshlab kiritilgan modul tizimi va keyingi versiyalarda rivojlangan Object mexanizmi orqali OYD ning asosiy tamoyillari Maple muhitida ham qisman amalga oshirilgan [4]. Shu sababli, Maple ma'lumot tiplari tizimi va OYD o'rtasidagi munosabatni o'rganish — ham nazariy, ham amaliy jihatdan muhim ilmiy masala hisoblanadi.

Ushbu maqolaning maqsadi — Maple tizimidagi ma'lumot tiplarinining to'liq tasnifini berish, ularning o'zaro munosabatini ko'rsatish va OYD bilan bog'liqligini ilmiy asosda tahlil qilishdan iborat. Maqolada qo'yilgan asosiy tadqiqot savollari quyidagilardir:

1. Maple ma'lumot tiplari tizimi qanday tuzilgan va qanday ierarxiyaga ega?
2. Maple tizimida dinamik tiplash qanday ishlaydi?
3. Maple modullari OYD ning qaysi tamoyillarini amalga oshiradi?
4. Maple obyekt tizimi an'anaviy OYD tillaridan qanday farq qiladi?

ASOSIY QISM

Maple — ramziy va sonli hisoblashlarni birlashtirgan ko'p maqsadli kompyuter algebrasi tizimi (CAS — Computer Algebra System) hisoblanadi. Maple tizimida yozilgan dasturlar o'ziga xos Maple tilida ifodalanadi, bu til esa funktsional va protsedural dasturlash paradigmalarini uyg'unlashtiradi [1].

Maple tilining asosiy xususiyatlaridan biri — kuchli va moslashuvchan ma'lumot tiplari tizimi. Maple'da barcha matematik ob'ektlar — sonlar, ifodalar,

ro'yxatlar, to'plamlar, funksiyalar — ma'lumot tiplari sifatida ifodalanadi. Maple interpretatsiya qilinadigan til bo'lib, uning ma'lumot tiplari kompilyatsiya vaqtida emas, balki ijro vaqtida aniqlanadi — bu dinamik tiplashning mohiyatini tashkil etadi [5].

Maple tizimining arxitekturasi ikki asosiy qismdan iborat:

1. **Yadro (Kernel)** — C tilida yozilgan va tezkor hisoblashlarni ta'minlaydigan past darajali qism;
2. **Kutubxona (Library)** — Maple tilida yozilgan va yuqori darajali funksiyalarni o'z ichiga olgan qism [1].

Ushbu ikki qatlamli arxitektura Maple ma'lumot tiplari tizimining ham asosini tashkil etadi.

Maple Ma'lumot Tiplari Tizimining Umumiy Tasnifi

Maple ma'lumot tiplarini uch asosiy toifaga ajratish mumkin: asosiy (primitive) tiplar, tuzilmali (structured) tiplar va maxsus (special) tiplar [6].

Asosiy Ma'lumot Tiplari

Sonlar (Numbers). Maple bir necha turdagi sonlarni qo'llab-quvvatlaydi:

1. *Butun sonlar (integer)* — Maple'da butun sonlar ixtiyoriy o'lchamda bo'lishi mumkin, ya'ni tizim katta sonlar arifmetikasini (arbitrary precision arithmetic) to'liq qo'llab-quvvatlaydi. Masalan, 2^1000 ifodasi Maple'da aniq hisob-kitob qilinadi [6].
2. *Ratsional sonlar (fraction)* — Maple ratsional sonlarni aniq (exact) shaklda saqlaydi: 3/7 ifodasi o'nli kasrga aylantirilmaydi, balki ratsional son sifatida saqlanadi.
3. *O'nli kasr (float)* — suzuvchi nuqtali sonlar; Maple'da ularning aniqligi Digits global o'zgaruvchisi orqali boshqariladi (standart qiymat 10 ta raqam).
4. *Kompleks sonlar (complex)* — Maple kompleks sonlarni a + b*I shaklida ifodalaydi, bu yerda I mavhum birlikni (√(−1)) bildiradi [7].

Mantiqiy qiymatlar (Boolean). Maple'da true, false va FAIL — uchlamchi mantiq (three-valued logic) qo'llaniladi. FAIL qiymati noaniqlikni ifodalaydi, bu Maple'ning matematik tabiati bilan bog'liq [6].

Satrlar (String). Maple'da satrlar qo'shtirnoq ichida yoziladi: "Salom, dunyo!". Satrlarga nisbatan standart amallar (length, cat, substring va boshqalar) qo'llanilishi mumkin [6].

Nomlar va identifikatorlar **.** Maple'da nomlar — bu hali qiymat berilmagan o'zgaruvchilar yoki matematik ramzlar. Masalan, x, y, alpha — bular Maple'da ramziy o'zgaruvchilar sifatida qaraladi va ular ustida algebraik amallar bajarish mumkin. Bu xususiyat Maple'ni oddiy dasturlash tillaridan tubdan farqlantiradi [5].

Tuzilmali Ma'lumot Tiplari

Ro'yxat (List). Maple ro'yxatlari kvadrat qavs bilan ifodalanadi: [1, 2, 3, 4]. Ro'yxatlar tartibli va o'zgarmas (immutable) bo'lib, indeks orqali elementlarga murojaat qilish mumkin. Maple'da ro'yxatlar heterojen bo'lishi mumkin, ya'ni turli tiplardagi elementlarni o'z ichiga olishi mumkin [6].

To'plam (Set). To'plamlar figurali qavs bilan ifodalanadi: {1, 2, 3}. Matematikadagi to'plamlar kabi, Maple to'plamlarida tartib yo'q va elementlar takrorlanmaydi. To'plamlar ustida birlashtirish (union), kesishish (intersect), ayirma (minus) amallari bajarilishi mumkin [7].

Massiv va jadval (Array, Table). Maple'da Array — indeksli, o'zgaruvchan (mutable) tuzilma; Table esa assotsiativ massiv (kalit-qiymat juftliklari) hisoblanadi. Jadvallar Maple'da lug'at (dictionary) vazifasini bajaradi [6].

Ketma-ketlik (Sequence). Ketma-ketliklar Maple'ning o'ziga xos ma'lumot turi bo'lib, vergul bilan ajratilgan elementlar majmuini ifodalaydi. Masalan: a, b, c — uch elementli ketma-ketlik. Ketma-ketliklar ko'pincha funksiyalar argumenti sifatida ishlatiladi [5].

Matritsa va vektor (Matrix, Vector). Maple LinearAlgebra paketi orqali to'laqonli matritsa va vektor tiplarini taqdim etadi. Bu tiplar ham yadro darajasida optimallashtirilgan maxsus ma'lumot tuzilmalari hisoblanadi [7].

Maxsus Ma'lumot Tiplari

NULL. Maple'da NULL bo'sh ketma-ketlikni ifodalaydi. U Python'dagi None yoki Java'dagi null dan farqli ravishda, NULL hech narsa emas — u hatto ro'yxatga qo'shilganda ham iz qoldirmaydi [6].

Infinity. Maple infinity, -infinity va undefined qiymatlarini maxsus matematik ob'ektlar sifatida qo'llab-quvvatlaydi. Bu matematik analiz va cheksizlik bilan ishlashda juda muhimdir [7].

Algebraik ifodalar (Algebraic Expression). Maple'dagi eng universal tip — bu algebraik ifodalar tipidir. sin(x) + x^2 - 3*y kabi ifodalar Maple'da maxsus daraxt (expression tree) tuzilmasida saqlanadi. Bu tuzilma ramziy hisoblashlarning asosini tashkil etadi [5].

Maple'da Dinamik Tiplash Mexanizmi

Dinamik tiplash (dynamic typing) — dastur ijro vaqtida o'zgaruvchi tipining aniqlanishi demakdir. Maple — sof dinamik tiplangan til bo'lib, bu uning ramziy hisoblashlar uchun moslashuvchanligini ta'minlaydi [8].

Maple'da whattype() funksiyasi orqali istalgan ob'ektning tipi aniqlash mumkin:

```
whattype(42);       # integer
whattype(3.14);     # float
whattype([1,2,3]);  # list
whattype(x^2 + 1);  # `+`
```

E'tiborli jihati shundaki, x^2 + 1 ifodasi + (qo'shish operatori) tipiga ega, chunki Maple ifodalarni daraxt tuzilmasida saqlaydi va daraxtning ildiz operatorini tip sifatida ko'rsatadi [5].

Maple'da tiplash mexanizmi quyidagi asosiy tamoyillarga asoslanadi:

1. **Kechiktirilgan baholash (Lazy Evaluation)** — Maple ifodalarni darhol hisoblash o'rniga, kerak bo'lganda baholaydi. Bu ramziy hisoblashlar uchun juda muhim xususiyat [8].

2. **Avtomatik soddalashtirish (Automatic Simplification)** — Maple ba'zi ifodalarni avtomatik ravishda soddalashtiradi. Masalan, 0 * x har doim 0 ga aylantiriladi [5].

3. **Tip tekshiruvi (Type Checking)** — type() funksiyasi orqali ob'ekt berilgan tipga mosligini tekshirish mumkin: type(42, integer) → true.

Maple'da Foydalanuvchi Tomonidan Aniqlanadigan Tiplar

Maple tizimi foydalanuvchilarga o'z tiplarini aniqlash imkonini beradi. Buning ikkita asosiy usuli mavjud:

Birinchi usul: TypeTools paketi. Ushbu paket orqali yangi tiplar ro'yxatdan o'tkaziladi:

```
TypeTools:-AddType('PositiveEven',
   x -> type(x, posint) and (x mod 2 = 0));
type(4, PositiveEven);   # true
type(3, PositiveEven);   # false
```

Ikkinchi usul: Record tuzilmasi. Maple'da Record — ma'lumotlarni guruhlash uchun ishlatiladigan tuzilma bo'lib, u OYD'dagi struct yoki sodda klassga o'xshaydi:

```
talaba := Record('ism' = "Alisher", 'yosh' = 20, 'baho' = 4.5);
talaba:-ism;    # "Alisher"
talaba:-yosh;   # 20
```

Bu yondashuv ma'lumotlarni mantiqan guruhlash imkonini beradi, ammo to'laqonli OYD klassiga nisbatan cheklangan funksionallikka ega [9].

Maple Modullari va Obyektga Yo'naltirilgan Dasturlash

Maple'dagi eng muhim OYD mexanizmi — **modul tizimi** hisoblanadi. Modul tizimi Maple 6-versiyasida kiritilgan va u Modula-2 hamda SML tillaridagi modul tushunchasiga asoslanadi [4].

Modullarning Tuzilishi

Maple moduli quyidagi umumiy ko'rinishga ega:

```
ModulNomi := module()
   description "Modul tavsifi";
   local xususiyPrivat;
   export ommaviyFunksiya;
```

```
    xususiyPrivat := 10;
        ommaviyFunksiya := proc(x)
        return x + xususiyPrivat;
    end proc;
    end module;
```

Bu tuzilmada local kalit so'zi bilan e'lon qilingan o'zgaruvchilar va funksiyalar moduldan tashqarida ko'rinmaydi — bu **inkapsulatsiya** tamoyilining amalga oshirilishidir [4].

Inkapsulatsiya

Inkapsulatsiya — ma'lumot va uni qayta ishlovchi metodlarni birlashtirish, shuningdek, ichki holatni tashqaridan yashirish tamoyili [3]. Maple modullarida bu tamoyil local va export kalit so'zlari orqali amalga oshiriladi:

- local — faqat modul ichida ko'rinadigan (private) elementlar;
- export — moduldan tashqarida foydalanish mumkin bo'lgan (public) elementlar;
- global — global muhitdagi o'zgaruvchilar.

Masalan, bank hisobi modelini ko'raylik:

```
BankHisobi := module()
    local balans;
    export pul_qush, pul_ol, balans_korsatish;
        balans := 0;
        pul_qush := proc(miqdor::positive)
        balans := balans + miqdor;
        printf("Hisobga %g so'm qo'shildi. Joriy balans: %g\n", miqdor, balans);
    end proc;
        pul_ol := proc(miqdor::positive)
        if miqdor > balans then
            error "Hisobda yetarli mablag' yo'q";
        end if;
        balans := balans - miqdor;
```

```
    printf("Hisobdan %g so'm olindi. Joriy balans: %g\n", miqdor, balans);
end proc;
    balans_korsatish := proc()
    return balans;
end proc;
end module;
```

Ushbu misolda balans o'zgaruvchisi local deb e'lon qilingan, shuning uchun tashqaridan bevosita o'zgartirib bo'lmaydi. Bu klassik inkapsulatsiyaning namunasidir [9].

XULOSA

Ushbu maqolada Maple kompyuter algebrasi tizimidagi ma'lumot tiplari tizimi va uning obyektga yo'naltirilgan dasturlash bilan bog'liqligi chuqur ilmiy tahlil qilindi. Olib borilgan tadqiqot asosida quyidagi xulosalar chiqarildi:

Maple ma'lumot tiplari tizimi o'ta boy va moslashuvchan bo'lib, oddiy sonlardan tortib murakkab algebraik ifodalargacha bo'lgan barcha matematik ob'ektlarni qamrab oladi. Maple'ning eng muhim xususiyatlaridan biri — ixtiyoriy aniqlikdagi hisoblashlar va ramziy ifodalar bilan ishlash imkoni, bu an'anaviy dasturlash tillarida bevosita mavjud emas. Maple dinamik tiplash tizimidan foydalanadi, bu esa unga matematikaviy hisoblashlarda katta moslashuvchanlik beradi. whattype(), type(), TypeTools kabi vositalar orqali tiplash mexanizmi nazorat qilinadi.

FOYDALANILGAN ADABIYOTLAR

1. Char, B. W., Geddes, K. O., Gonnet, G. H., Leong, B. L., Monagan, M. B., & Watt, S. M. (1991). *Maple V Language Reference Manual*. Springer-Verlag, New York. 267 b.
2. Pierce, B. C. (2002). *Types and Programming Languages*. MIT Press, Cambridge, Massachusetts. 645 b.
3. Booch, G. (2007). *Object-Oriented Analysis and Design with Applications*. 3-nashri. Addison-Wesley Professional, Boston. 720 b.

4. Monagan, M., & Geddes, K. (2009). *Maple Advanced Programming Guide*. Maplesoft, Waterloo, Ontario. 389 b.

5. Heck, A. (2003). *Introduction to Maple*. 3-nashri. Springer, New York. 828 b.

6. Redfern, D. (1996). *The Maple Handbook: Maple V Release 4*. Springer, New York. 497 b.

7. Corless, R. M. (2004). *Essential Maple 7: An Introduction for Scientific Programmers*. Springer, New York. 374 b.

8. Gander, W., & Hrebicek, J. (2004). *Solving Problems in Scientific Computing Using Maple and MATLAB*. 4-nashri. Springer, Berlin Heidelberg. 423 b.

9. Abell, M. L., & Braselton, J. P. (2005). *Maple by Example*. 3-nashri. Elsevier Academic Press, Burlington. 588 b.

10. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, Massachusetts. 395 b.