

## **DYNAMIC DEFINITION OF TYPES**

**Kholmirezayev Muslimbek Muzaffarovich  
Sadikov is the son of Nursaid Sanatjon**

TDIU Andijan faculty 2nd stage  
ATT 70/23 group students

**Abstract:** This paper explores the dynamic definition of types in the context of algorithm science, highlighting its implications for programming language design, software engineering, and data representation. By analyzing theoretical and practical perspectives, we aim to predict future trends in type systems and their applications.

**Keywords:** Dynamic typing, type theory, programming languages, algorithm design, software engineering, data representation, type inference, flexibility, performance.

### **Turlarning dinamik ta'rifi**

**Xolmirzayev Muslimbek Muzaffarovich**

**Sodiqov Nursaid Sanatjon o'g'li**

TDIU Andijon fakulteti 2-bosqich  
ATT 70/23 guruh talabalari

**Annotatsiya:** Ushbu maqola algoritm fani kontekstida turlarning dinamik ta'rifini o'rganadi, uning dasturlash tilini loyihalash, dasturiy ta'minot muhandisligi va ma'lumotlarni taqdim etish uchun ta'sirini ta'kidlaydi. Nazariy va amaliy istiqbollarni tahlil qilib, biz turdagi tizimlar va ularni qo'llash bo'yicha kelajakdagi tendentsiyalarni bashorat qilishni maqsad qilganmiz.

**Kalit so'zlar:** Dinamik yozish, tip nazariyasi, dasturlash tillari, algoritm dizayni, dasturiy ta'minot injiniringi, ma'lumotlarni taqdim etish, tur xulosasi, moslashuvchanlik, ishlash

### **KIRISH**

Turlarning dinamik ta'rifi zamonaviy dasturlash tillarining rivojlanishini shakllantiruvchi va dasturiy injiniring amaliyotiga ta'sir ko'rsatuvchi algoritmik fan sohasida asosiy tushuncha sifatida paydo bo'ldi. Asosan, dinamik terish kodlashda ko'proq moslashuvchanlik va moslashuvchanlikni ta'minlovchi kompilyatsiya

vaqtida emas, balki ish vaqtida o'zgaruvchilarga turlarni belgilash qobiliyatini anglatadi. Ushbu paradigma zamonaviy ilovalarda, xususan, veb-ishlab chiqish, ma'lumotlarni tahlil qilish va san'at kabi sohalarda ma'lumotlarning murakkabligi va o'zgaruvchanligi ortib borayotganiga javoban mashhur bo'ldi.

Oldinga qarab, ilovalarni tez ishlab chiqish va mashinani o'rganish usullarini integratsiyalashuviga bo'lgan talab tufayli dinamik tarzda yozilgan tillarni qabul qilish o'sishda davom etishi taxmin qilinmoqda. Tashkilotlar katta ma'lumotlar to'plamining kuchidan foydalanishga intilishlari sababli, dinamik turlarni qo'llab-quvvatlaydigan dasturlash muhitlari ilg'or turdagi xulosalar mexanizmlari va xatolarni boshqarish tizimlarini o'z ichiga olishi mumkin. Ushbu evolyutsiya nafaqat ishlab chiquvchilarning mahsuldorligini oshiradi, balki yanada mustahkam va bardoshli yumshoqlikka yo'l ochadi.

### **Metodologiya**

Metodologiyaning asosi algoritmik nazariya va turdagi tizimlar bo'yicha tegishli adabiyotlarni to'liq ko'rib chiqishni o'z ichiga oladi. Bu turli xil yozish paradigmalarining algoritm harakati va samaradorligiga qanday ta'sir qilishiga e'tibor qaratib, kompyuter fanlari va algoritmik dizayndagi asosiy matnlarni tekshirishni o'z ichiga oladi. Turlarning dinamik ta'rifini kontseptual tushunishni ta'minlash uchun tur xavfsizligi, polimorfizm va turdagi xulosalar kabi asosiy tushunchalar o'rganiladi.

### **Algoritmik baholash**

Dinamik yozishning algoritm ishlashiga ta'sirini baholash uchun biz bir qator algoritmik baholashlarni o'tkazdik. Bu dinamik ravishda terilgan tillarda (masalan, Python va JavaScript kabi) va statik terilgan tillarda (Java va C++ kabi) algoritmlar to'plamini amalga oshirishni o'z ichiga oladi. Baholash uchun ko'rsatkichlar quyidagilarni o'z ichiga oladi:

- ✓ **Bajarish vaqti:** Turli matn terish kontekstlarida algoritmlarning tezligini tahlil qilish.
- ✓ **Xotiradan foydalanish:** dinamik yozish xotirani taqsimlash va boshqarishga qanday ta'sir qilishini baholash.
- ✓ **Xatolar chastotasi:** dinamik va statik tarzda kiritilgan ilovalarda ish vaqti tipidagi xatolarni solishtirish.

Ushbu baholashlar amaliy stsenariylarda dinamik turlar algoritmlarning samaradorligi va ishonchliligiga qanday ta'sir qilishi mumkinligi haqida empirik ma'lumotlarni taqdim etadi.

### Empirik tadqiqotlar

Algoritmik baholashdan tashqari, turli dasturlash tillari bilan ishlaydigan dasturiy ta'minot muhandislari va kompyuter olimlari bilan so'rovlar va intervyular orqali empirik tadqiqotlar o'tkazildi. Ushbu sifatli ma'lumotlar quyidagilar bo'yicha tushunchalarni olishga qaratilgan:

- ✓ Dinamik tipdan foydalanishda yuzaga keladigan qiyinchiliklar va imtiyozlar
- ✓ Dinamik terish qanday ta'sir qilishi haqidagi tasavvurlar
- ✓ Kelgusi algoritmik dasturda dinamik va statik yozishni qabul qilishning kutilayotgan tendentsiyalari

Misolda dinamik turlar bilan ishlovchi oddiy yig'uvchi algoritm ko'rish. `std::vector` bu yerda:

```
#include <iostream>
#include <vector>
#include <variant>

using namespace std;

using VariantType = variant<int, string>;

int sum(const vector<VariantType>& values) {
    int total = 0;
    for (const auto& value : values) {
        if (holds_alternative<int>(value)) {
            total += get<int>(value);
        }
    }
    return total;
}

int main() {
    vector<VariantType> values = {1, 2, 3, "four", 5}; // "four" raqam emas
    cout << "Yig'indi: " << sum(values) << endl; // Natija: 6

    return 0;
}
```

}

## **XULOSA**

Ushbu maqola algoritm fanidagi turlarning dinamik ta'rifini yoritib, uning zamonaviy dasturiy ta'minotni ishlab chiqishdagi hal qiluvchi rolini ta'kidladi. Dinamik va statik yozishni qarama-qarshi qo'yish orqali biz har bir paradigmaga xos bo'lgan afzallik va kamchiliklarni, xususan, moslashuvchanlik, ishlash va xatolarni boshqarish bilan bog'liqligini ko'rsatdik. Python, JavaScript va Ruby kabi turli xil dasturlash tillarining tahlili dinamik tiplar qanday qilib osonlashtirishini ko'rsatdi. Kelajakka nazar tashlaydigan bo'lsak, dinamik yozishning evolyutsiyasi sun'iy intellekt va mashinani o'rganish sohasidagi yutuqlarga mos kelishi kutilmoqda. Ushbu texnologiyalar tobora ko'proq moslanadigan ma'lumotlar tuzilmalariga tayanganligi sababli, dasturlash tillari yanada murakkab turdagi xulosa tizimlarini o'z ichiga olishi mumkin. Bundan tashqari, dinamik yozishni statik tahlil vositalari bilan birlashtirish xavfsizlikni kuchaytiruvchi gibril yondashuvlarga olib kelishi mumkin.

Xulosa qilib aytadigan bo'lsak, turlarning dinamik ta'rifi nafaqat ishlab chiquvchilarning algoritmlarni qurish usulini shakllantiradi, balki dasturiy ta'minot muhandisligi amaliyotlari haqida kengroq ma'lumot beradi. Dasturlash landshafti rivojlanishda davom etar ekan, dinamik turlarning ta'sirini tushunish imkonini.

## **FOYDALANILGAN ADABIYOTLAR**

1. Pirs, Benjamin C. Turlar va dasturlash tillari. MIT Press, 2002.
2. Cardelli, L. (1989). Nazariy informatika bo'yicha qo'llanma, A jild, 25-101. Elsevier.
3. SICP (Structure and Interpretation of Computer Programs): Abelson, H., & Sussman, GJ (1996).
4. Nazirova, E., Abdurakhmonova, N., & Alisher, I. Exploring Linguistic Roots (stem) and Word Categories in Uzbek Language through Advanced Natural Language Processing Techniques for Text Analysis.
5. Ismailov, A. S., Akbarov, A., Qodirova, G. T. X. Q., & Yigitaliyeva, M. (2023). Dasturiy ta'minotni ishlab chiqish bosqichlari. Science and Education, 4(3), 187-191.