

JAVA DASTURLASH TILINING ASOSIY TUSHUNCHALARI: TILING TASHKIL ETUVCHILARI, PRIMITIV TURLAR, OPERATORLAR VA MASSIVLAR

Umarov Bekzod Azizovich

*Farg'ona davlat universiteti Amaliy matematika va
Informatika kafedrasida katta o'qituvchisi p.f.b.d (PhD)*

ubaumarov@mail.ru

Hakimov Muhammadqodir Tolibjon o'g'li

Farg'ona davlat universiteti talabasi
muhammadqodirxakimov775@gmail.com

Annotatsiya: Mazkur maqolada Java dasturlash tilining asosiy tushunchalari ilmiy yondashuv asosida tahlil qilinadi. Dastlab tilning strukturaviy asoslari — sinflar, metodlar, ob'ektlar va paketlar ko'rib chiqildi. Shundan so'ng, Java tilining statik va kuchli tiplangan tip tizimi doirasida primitiv ma'lumot turlari va ularning samaradorligi tahlil qilindi. Shuningdek, operatorlarning algoritmik roli va ular yordamida mantiqiy qarorlar ishlab chiqish imkoniyatlari o'rganildi. Yakunda massivlarning tuzilmasi, ularning samarali foydalanilishi va hisoblash algoritmalaridagi o'rni chuqur yoritildi. Maqola davomida nazariy bilimlar amaliy misollar bilan mustahkamlandi, hamda dasturiy komponentlarning ilmiy va tizimli yondashuv asosida baholanishiga alohida e'tibor qaratildi.

Kalit so'zlar: Java, sinf, metod, ob'ekt, primitiv turlar, massiv, operator, statik tip tizimi, ob'ektga yo'naltirilgan dasturlash, algoritm, kapsulyatsiya, modullilik, indekslash, semantika, dasturiy arxitektura

Аннотация: В данной статье анализируются основные концепции языка программирования Java на основе научного подхода. Сначала были рассмотрены структурные основы языка — классы, методы, объекты и пакеты. Затем были проанализированы примитивные типы данных и их эффективность в рамках статической и строго типизированной системы типов языка Java. Также была изучена алгоритмическая роль операторов и возможности разработки логических решений с их помощью. Наконец, была подробно рассмотрена структура массивов, их эффективное использование и роль в вычислительных алгоритмах. На протяжении всей статьи теоретические знания подкреплялись практическими примерами, а особое внимание уделялось оценке программных компонентов на основе научного и системного подхода.

Ключевые слова: Java, класс, метод, объект, примитивные типы, массив, оператор, статическая система типов, объектно-ориентированное программирование, алгоритм, инкапсуляция, модульность, индексирование, семантика, архитектура программного обеспечения

Abstract: This article analyzes the basic concepts of the Java programming language based on a scientific approach. First, the structural foundations of the language - classes, methods, objects and packages - were considered. Then, primitive data types and their effectiveness within the static and strongly typed type system of the Java language were analyzed. Also, the algorithmic role of operators and the possibilities of developing logical decisions with their help were studied. Finally, the structure of arrays, their effective use and role in computing algorithms were thoroughly covered. Throughout the article, theoretical knowledge was reinforced with practical examples, and special attention was paid to the evaluation of software components based on a scientific and systematic approach.

Keywords: Java, class, method, object, primitive types, array, operator, static type system, object-oriented programming, algorithm, encapsulation, modularity, indexing, semantics, software architecture

Introduction (Kirish)

Dasturlash tillari axborot texnologiyalari taraqqiyotida asosiy poydevor hisoblanadi. Ushbu tillar orasida **Java** — oʻzining yuqori darajadagi platformaviy mustaqilligi, xavfsizligi, kuchli tip tizimi va obʻektga yoʻnaltirilgan yondashuvi bilan ajralib turadi. Java dasturlash tili ilk marotaba **1995-yilda Sun Microsystems** tomonidan ishlab chiqilgan va hozirda **Oracle Corporation** tomonidan qoʻllab-quvvatlanadi. Tilning keng koʻlamda qoʻllanishi uning soddaligi, portativligi, modulligi va kuchli kutubxonalar toʻplamiga ega ekanligidan dalolat beradi.

Java tilining oʻziga xosligi – u dasturchilarni dastlabdan intizomli va strukturaviy yondashuvga oʻrgatadi. Shu sababli, ushbu maqolada Java dasturlash tilining eng muhim komponentlari: tilning asosiy tashkil etuvchilari, primitiv berilganlar turlari, operatorlar va massivlar **ilmiy-nazariy tahlil asosida** chuqur yoritiladi.

Methods (Uslublar)

Ushbu maqola **nazariy-analitik tadqiqot** uslubiga asoslanadi. Tahlil qilishda quyidagi manbalar asosiy vosita sifatida tanlandi:

- ✓ Java SE 17 rasmiy hujjatlari (Oracle)
- ✓ Akademik adabiyotlar: Horstmann & Cornell (Core Java), Schildt (Java: The Complete Reference)
- ✓ Amaliy mashgʻulotlar, laboratoriya ishlari, va ochiq kodli loyihalardan olingan real kod namunalari

Shuningdek, **induktiv yondashuv** orqali har bir tushuncha — avval umumiy nazariy koʻrinishda, keyin esa amaliy misollar bilan yoritildi. Bu metod bilimlarni sistemalashtirishga xizmat qiladi.

Results (Natijalar)

Tadqiqot natijasida Java dasturlash tilining to'rtta asosiy komponenti chuqur o'rganildi:

1. Tiling asosiy tashkil etuvchilari

Java tilining umumiy tuzilmasi quyidagilardan iborat:

- **Class (sinf):** Java dasturining asosiy qurilish bloki. Har bir dastur sinfdan boshlanadi.

1. public class HelloWorld {
2. public static void main(String[] args) {
3. System.out.println("Salom, dunyo!");
4. }
5. }

- ✓ **Object (ob'ekt):** Har bir sinfdan ko'plab ob'ektlar yaratish mumkin. Bu OOP (Object-Oriented Programming) asosidir.

- ✓ **Method (metod):** Belgilangan amallarni bajaruvchi kod blokidir. main() metodi – Java dasturining boshlang'ich nuqtasi.

- ✓ **Package (paket):** Sinflarni lozim darajada tartibga soluvchi mexanizm. Masalan: java.util, java.io.

Bu struktura dasturchiga **modullilik, kodni qayta ishlatish, kapsulyatsiya va polimorfizm** kabi prinsiplarni joriy etishga imkon beradi.

2. Primitiv berilganlar turlari

Java'da 8 ta primitiv berilganlar turi mavjud bo'lib, ularning har biri **xotira hajmi va qiymatlar diapazoni** bilan farqlanadi:

Qiymatlar oraligi		
		-2^{31} dan $2^{31}-1$ gacha

Bu turlar **javada kuchli tip nazoratini** ta'minlaydi, ya'ni har bir o'zgaruvchi aniq turga ega bo'ladi va avtomatik ravishda boshqa turga o'ta olmaydi (faqat aniq kasting orqali).

Java dasturlash tilining asosiy komponentlari ustida olib borilgan nazariy va amaliy tahlillar natijasida quyidagi muhim xulosalar olingan. Har bir komponent dasturlash jarayonining strukturaviy, semantik va funksional qatlamlarini tashkil qiladi.

1. Tiling asosiy tashkil etuvchilari: Sinflar, metodlar, ob'ektlar va paketlar

Java tilining arxitekturaviy tuzilmasi **ob'ektga yo'naltirilgan dasturlash (OOP)** paradigmasiga asoslanadi. Har qanday funksional imkoniyatlar sinflar orqali ifodalanadi, bu esa Java'ning **modullilik**, **kapsulyatsiya** va **kodni qayta ishlatish** (reusability) tamoyillarini mustahkamlaydi.

✓ **Sinf** – OOP'da **abstraktsiya** vositasi hisoblanadi. Sinf yordamida murakkab obyektlarning xatti-harakatlari va atributlari bir joyga jamlanadi.

✓ **Metodlar** – harakatni ifodalovchi funksiyalar bo'lib, ular sinflar ichida aniqlanadi. Ular yordamida **algoritmik jarayonlar** tashkil qilinadi.

✓ **Ob'ektlar** – sinflarning real namunalari bo'lib, dasturda **holat (state)** va **xatti-harakat (behavior)** tushunchalarini ifodalaydi.

✓ **Paketlar** – kodni modul shaklida tashkil qilish imkonini beradi. Bu esa **ko'p modulli dasturiy arxitekturani** yaratish uchun zarur.

Ilmiy jihatdan qaralganda, bu komponentlar **sinflararo bog'liqlik (coupling)** va **sinflarning ichki murakkabligi (cohesion)** tushunchalarini tadbqiq qilishda muhim rol o'ynaydi. Yaxshi loyihalangan sinflar yuqori bog'liqlikdan (tight coupling) saqlanishi va maksimal ichki yaxlitlikka (high cohesion) ega bo'lishi kerak.

2. Primitiv turlar va tip tizimi

Java'dagi **statik va kuchli tiplangan** tip tizimi dasturchining xatoliklarni erta bosqichda aniqlashiga imkon beradi. Har bir primitiv tur **ma'lumotlarning aniqligi, xotira tejamkorligi va ishlash tezligi** uchun muhim hisoblanadi.

O'lchami			
			Fan, moliya, ilmiy hisob-kitoblar

Primitiv turlar ustida olib borilgan tahlillar shuni ko'rsatdiki, ular **yengil arifmetik** va **ko'p sonli takrorlanuvchi hisob-kitoblar** uchun eng maqbul tanlovdir. Masalan, int turidagi massivlar yordamida **milliy statistika**, **axborot oqimlarini filtrlash**, **real vaqtda ishlov berish** tizimlari ishlab chiqiladi.

Shuningdek, **boxing/unboxing** mexanizmlari (masalan: $\text{int} \rightarrow \text{Integer}$) orqali primitiv turlarni ob'ekt sifatida ishlatish imkoniyati mavjud. Bu esa ularni kolleksiya strukturalarida (List, Map) qo'llashga yo'l ochadi.

3. Operatorlar: Sintaksis va semantik tizimlar

Operatorlar Java tilining **mantiqiy, hisoblash va qaror qabul qilish** mexanizmlarini tashkil qiladi. Ularning to'g'ri tanlanishi va optimallashtirilgan ishlatilishi — kodning samaradorligi, o'qilishi va bajarilish tezligiga bevosita ta'sir ko'rsatadi.

Java'da operatorlar quyidagi semantik rollarni bajaradi:

- ✓ **Aritmetik operatorlar:** matematik modellarni tuzishda asosiy vosita ($y = mx + b$ kabi).
- ✓ **Mantiqiy operatorlar:** if, while, for bloklarida qaror daraxtlari tashkil qiladi.
- ✓ **Bitli operatorlar:** kriptografiya, grafik, va signal ishlov berish algoritmlarida qo'llaniladi.
- ✓ **Tayinlash operatorlari:** qiymatlar oqimini boshqaradi va **mutable stateni** tashkil qiladi.

Masalan, quyidagi kodda ifoda murakkab mantiqiy qaror asosida shakllantirilgan:

```
if ((a > b && c < d) || flag == true) {  
    // muayyan shart bajarilganda  
}
```

Ilmiy yondashuvda bu mantiqiy ifoda **Boolean algebraga** asoslanadi. Bu esa dastur tuzilishini matematik ifodalar bilan tekshirishga imkon yaratadi.

4. Massivlar: Tizimli ma'lumotlar tuzilmasi

Massivlar Java'da **ma'lumotlar strukturasi**ning eng fundamental shakli bo'lib, ularni analiz qilishda quyidagi mezonlar e'tiborga olinadi:

- ✓ **Turg'unlik (statik o'lcham):** Java'dagi massivlar bir marta yaratilgach, o'lchami o'zgarmaydi. Bu kompilyatsiya vaqtida **xotira ajratishni optimallashtiradi**.
- ✓ **Indekslashgan kirish:** har bir elementga tezkor kirish imkonini beradi ($O(1)$).
- ✓ **Turlarning bir xil bo'lishi:** massivda faqat bitta turdagi ma'lumotlar saqlanadi, bu esa **ma'lumot yaxlitligini (data consistency)** ta'minlaydi.
- ✓ **Multidimensional massivlar:** ko'p o'lchovli (matritsa, tensor) tuzilmalarni tashkil etishda foydalidir.

```
int[][] matritsa = new int[3][3];
```

```
matritsa[0][0] = 1;
```

Massivlar ko'plab **tadqiqot yo'nalishlarida:** chiziqli algebra, rasm tahlili, ma'lumotlar saralash va indekslash algoritmlarida faol qo'llaniladi. Ilmiy dasturlashda (masalan, differensial tenglamalarni raqamli yechishda) massivlar asosiy hisoblash vositasi hisoblanadi.

Bundan tashqari, Java'dagi Arrays klassi yordamida **saralash (sort)**, **qidiruv (search)** va **nusxalash (copy)** kabi amallarni samarali bajarish mumkin. Bu amallar algoritmik jihatdan $O(n \log n)$ va $O(\log n)$ darajadagi optimallashtirilgan metodlarni o'z ichiga oladi.

3. Operatorlar

Java tilida turli matematik va mantiqiy amallarni bajarish uchun operatorlar mavjud:

1. **Aritmetik:** +, -, *, /, % – odatiy hisoblashlar
2. **Solishtirish:** ==, !=, >, <, >=, <= – shartlarni tekshirish
3. **Mantiqiy:** &&, ||, ! – shartlar kombinatsiyasi

4.**Bitli:** &, |, ^, ~ – raqamlar ustida bit darajasidagi amallar

5.**Tayinlash:** =, +=, -=, *=, /= – qiymat biriktirish va o‘zgartirish

Masalan:

```
int a = 5, b = 10;
```

```
boolean natija = (a < b) && (b > 0); // true
```

4. Massivlar

Java’da massivlar **bir xil turdagi elementlar to‘plami** bo‘lib, ular ob’ekt sifatida qaraladi:

```
int[] sonlar = new int[5]; // bo‘sh massiv
```

```
String[] ismlar = {"Ali", "Vali"}; // to‘ldirilgan massiv
```

Massivlar yordamida katta hajmdagi ma’lumotlar ustida **tejamkor va tizimli ishlov berish** mumkin. Ko‘p o‘lchovli massivlar ham mavjud:

```
int[][] matritsa = {  
    {1, 2},  
    {3, 4}  
};
```

Discussion (Muhokama)

Yuqorida ko‘rib chiqilgan barcha komponentlar — sinflar, primitiv turlar, operatorlar va massivlar — dasturlashda **asosiy konstruktor birliklar** hisoblanadi. Java dasturlash tilining bu tarkibiy qismlari nafaqat tilning sintaksisiga, balki **algoritmik fikrlash, murakkab tizimlar yaratish va ularni optimallashtirish**ga ham xizmat qiladi.

Masalan, primitiv turlar yordamida dastur samaradorligini oshirish, operatorlar orqali murakkab shartli strukturani yaratish, massivlar orqali ma’lumotlar ustida yuqori samarali algoritmlar tuzish mumkin. Ayniqsa, massivlar va operatorlarning birgalikda ishlatilishi **qidiruv, saralash, algoritmik tahlil** kabi yo‘nalishlarda keng qo‘llaniladi.

Bundan tashqari, Java tilining qat’iy tiplanganligi dasturda yuzaga kelishi mumkin bo‘lgan xatoliklarni kompilyatsiya vaqtida aniqlab berish imkonini yaratadi. Bu esa **xavfsiz va barqaror dasturiy mahsulotlar** ishlab chiqishga xizmat qiladi.

Conclusion (Xulosa)

Java dasturlash tilining asosiy tushunchalari — sinf, ob’ekt, metod, primitiv turlar, operatorlar va massivlar — tilni o‘rganayotgan har bir dasturchi uchun zarur bilimlar asosini tashkil etadi. Ushbu komponentlar dasturchini strukturaviy va modulli fikrlashga o‘rgatadi, real muammolarni tizimli tarzda hal qilishga yo‘naltiradi.

Tadqiqot jarayonida aniqlanishicha, bu tushunchalarni chuqur o‘zlashtirish Java tilining ob’ektga yo‘naltirilgan imkoniyatlaridan to‘liq foydalanish, zamonaviy dasturiy tizimlar yaratish va ularni samarali boshqarishga asos bo‘ladi. Kelgusidagi ilmiy

yoʻnalishlar ushbu komponentlarning murakkab tizimlar – masalan, GUI, multithreading, va tarmoq dasturlari – bilan integratsiyasini chuqurroq oʻrganishga qaratilishi mumkin.

Foydalanilgan adabiyotlar:

1. Umarov B. RAQAMLI TEXNOLOGIYALAR VOSITASIDA PEDAGOGLARNING PROFESSIONAL KOMPETENTLIGINI RIVOJLANTIRISH MAZMUNI //Евразийский журнал математической теории и компьютерных наук. – 2023. – Т. 3. – №. 5. – С. 87-93.
2. Azizovich U. B. PRINCIPLES OF FORMING TEACHER COMPETENCE THROUGH INNOVATIVE TECHNOLOGIES. Finland International Scientific Journal of Education //Social Science & Humanities. – 2023. – Т. 11. – №. 5. – С. 823-828.
3. Azizovich U. B. PEDAGOGICAL-PSYCHOLOGICAL PRINCIPLES OF THE FORMATION OF PROFESSIONAL COMPETENCE //Confrencea. – 2023. – Т. 6. – №. 6. – С. 204-212.
4. Azizovich U. B., Zarifjon oʻgʻli X. N. BULUT TEXNOLOGIYALARINING AFZALLIKLARI VA KAMCHILIKLARI //TAʼLIM, TARBIYA VA INNOVATSIYALAR JURNALI. – 2024. – Т. 1. – №. 1. – С. 46-54.
5. Azizovich U. B., Rustamjon oʻgʻli R. Z. MAʼLUMOTLARNI SHIRFLASH TENALOGIYALARI VA XAVFSIZLIK STANDARTLARI //TAʼLIM, TARBIYA VA INNOVATSIYALAR JURNALI. – 2024. – Т. 1. – №. 1. – С. 105-108.
6. Azizovich U. B. et al. OLAP TIZIMLARINING ASOSIY PRINSIPLARI //TAʼLIM, TARBIYA VA INNOVATSIYALAR JURNALI. – 2024. – Т. 1. – №. 1. – С. 81-86.
7. Azizovich U. B. THE DEVELOPMENT OF PROFESSIONAL COMPETENCY OF TEACHERS IN EDUCATIONAL TECHNOLOGY BASED ON DIGITAL TECHNOLOGIES //Eurasian Journal of Mathematical Theory and Computer Sciences. – 2024. – Т. 4. – №. 7. – С. 11-14.
8. Azizovich U. B. et al. MASHINALI OʻQITISHDA REGRESSIYA ENG KICHIK KVADRATLAR USULINI QOʻLLASH //INNOVATION IN THE MODERN EDUCATION SYSTEM. – 2024. – Т. 5. – №. 46. – С. 266-270.