

## HASH MA'LUMOTLAR TUZILMASI VA UNING METODLARI BILAN ISHLASH.

### HASH DATA STRUCTURE AND ITS METHODS ХЕШ-СТРУКТУРА ДАННЫХ И ЕЁ МЕТОДЫ

**Abdullayev Shaxboz Solijon o'g'li**

*FarDU Axborot texnologiyalari kafedrasida katta o'qituvchisi*

[shaxbozfardu2023@gmail.com](mailto:shaxbozfardu2023@gmail.com)

ORCID ID 0000-0001-9382-732X

**Qosimjonova Sadoqat Elmurod qizi**

*Farg'ona davlat universiteti Axborot tizimlari va texnologiyalari yo'nalishi  
1-kurs talabasi*

[sadoqatqosimjonova4@gmail.com](mailto:sadoqatqosimjonova4@gmail.com)

**Annotatsiya.** Hash ma'lumotlar tuzilmasi — bu ma'lumotlarni samarali saqlash va tezkor qidirish uchun ishlatiladigan muhim ma'lumotlar strukturasi. Hash funksiyasi yordamida ma'lumotlar indekslanadi, bu esa ularni tezkor topish imkonini beradi. Hash ma'lumotlar tuzilmasi asosan "key-value" (kalit-qiyamat) juftliklari ko'rinishida saqlanadi. Bu tuzilma Python, Java va boshqa dasturlash tillarida keng qo'llaniladi. Hash tuzilmasining metodlari, masalan, `put()`, `get()`, `remove()`, `containsKey()` kabi metodlar orqali ma'lumotlarni qo'shish, olish, o'chirish va tekshirish amallari bajariladi. Hash ma'lumotlar tuzilmasi samaradorligi va tezligi bilan dasturlashda juda muhim ahamiyatga ega.

**Kalit so'zlar:** Hash ma'lumotlar tuzilmasi, hash funksiyasi, key-value juftligi, metodlar, ma'lumotlarni qidirish, samaradorlik.

**Annotation.** The hash data structure is a fundamental structure used to store and efficiently search data. Through the use of a hash function, data is indexed, allowing for quick retrieval. A hash data structure is typically represented in the form of key-value pairs. It is widely used in programming languages such as Python, Java, and others. Methods of hash data structures, such as `put()`, `get()`, `remove()`, `containsKey()`, allow for operations like adding, retrieving, deleting, and checking data. The efficiency and speed of hash structures are crucial in programming.

**Keywords:** Hash data structure, hash function, key-value pairs, methods, data retrieval, efficiency.

**Аннотация.** Хеш-структура данных является важной структурой, используемой для эффективного хранения и быстрого поиска данных. С помощью хеш-функции данные индексируются, что позволяет быстро их находить. Хеш-структура данных обычно представляется в виде пар "ключ-значение". Эта структура широко используется в языках программирования, таких как Python, Java и другие. Методы хеш-структуры данных, такие как `put()`, `get()`, `remove()`, `containsKey()`, позволяют выполнять операции добавления, получения, удаления и проверки данных. Эффективность и скорость хеш-структур имеют важное значение в программировании.

**Ключевые слова:** Хеш-структура данных, хеш-функция, пара ключ-значение, методы, поиск данных, эффективность.

## KIRISH

Hash ma'lumotlar tuzilmasi — bu dasturlashda ma'lumotlarni samarali saqlash va tezkor qidirish uchun ishlatiladigan kuchli vosita. Ma'lumotlarni tartibga solish va saqlash jarayonini tezlashtirish uchun hash funksiyasi yordamida har bir elementga indeks beriladi. Bu tuzilma asosan "key-value" (kalit-qiymat) juftliklari orqali ma'lumotlarni tashkil etadi, ya'ni har bir ma'lumotning o'ziga xos kaliti bo'ladi va shu kalit orqali unga murojaat qilinadi. Hash ma'lumotlar tuzilmasi yuqori samaradorlikka ega bo'lib, ko'plab dasturlash tillarida (masalan, Python, Java, C++) keng qo'llaniladi. Ushbu maqolada hash ma'lumotlar tuzilmasi, uning xususiyatlari, metodlari va ularning dasturlashdagi ahamiyati haqida batafsil so'z yuritiladi. Hash ma'lumotlar tuzilmasi (hash table yoki hash map) — bu ma'lumotlarni samarali saqlash va tezkor qidirish imkonini beruvchi ma'lumotlar strukturasidir. U ma'lumotlarni "key-value" (kalit-qiymat) juftliklari tarzida saqlaydi. Hash ma'lumotlar tuzilmasi hash funksiyasidan foydalanadi, bu funksiya orqali ma'lumotlar indekslanadi, ya'ni har bir elementga o'ziga xos bir kalit beriladi. Bu kalit yordamida ma'lumotlarni osonlik bilan izlash va olish mumkin. Hash tuzilmasi asosida ishlov berish jarayonlari juda samarali bo'lib, ular  $O(1)$  vaqt murakkabligida amalga oshiriladi. Bu esa ma'lumotlarga tezkor kirishni ta'minlaydi. Hash tuzilmasida, har bir element (ma'lumot) uchun hash funksiyasi tomonidan

hisoblangan indeks yordamida joy ajratiladi. Indekslar asosan sonlar bo‘lib, ular elementlarni jadvalning turli joylariga joylashtiradi. Bu hash funksiya orqali amalga oshiriladi. Ma‘lumotlar jadvalga joylashtirilganidan so‘ng, kerakli ma‘lumotni topish uchun kalitdan foydalanish mumkin. Jadvalda saqlangan har bir kalit unikaldir, bu esa qayta ishlashni va qidirishni juda osonlashtiradi.

Hash funksiyasi — bu ma‘lumotni qabul qilib, uni mos ravishda indeks yoki boshqa shaklda qaytaruvchi algoritmdir. Bu indeks ma‘lumotning jadvaldagi joylashuvini aniqlaydi. Hash funksiyasining asosiy vazifasi — ma‘lumotni tezda saqlash va tezda olish imkoniyatini yaratishdir. Hash funksiyasi to‘g‘ri ishlashi uchun quyidagi xususiyatlarga ega bo‘lishi kerak:

**Deterministiklik:** Bir xil kirish qiymatiga nisbatan har doim bir xil chiqish qiymatini qaytarishi kerak.

**Tezlik:** Hash funksiyasi tez ishlashi lozim, chunki uning vazifasi tezkor qidirishni ta‘minlashdir.

**To‘liq taqsimot:** Har bir kirish qiymatiga mos ravishda hash qiymati teng ravishda taqsimlanishi kerak, bu esa hash jadvalining to‘ldirishini oldini oladi.

Har bir hash funksiyasi o‘ziga xos bo‘lib, turli maqsadlar uchun turli algoritmlar ishlab chiqilgan. Misol uchun, **MD5**, **SHA-256** kabi hash funksiyalari ma‘lumotlarni indekslashda ishlatiladi.

Hash ma‘lumotlar tuzilmasi bir nechta muhim metodlarga ega bo‘lib, ular yordamida ma‘lumotlar ustida ishlash amalga oshiriladi:

**put(key, value):** Ushbu metod hash jadvaliga yangi ma‘lumotlarni qo‘shish uchun ishlatiladi. Agar berilgan kalit jadvalda allaqachon mavjud bo‘lsa, qiymat yangilanadi. Agar kalit mavjud bo‘lmasa, yangi kalit-qiymat juftligi qo‘shiladi. `put('name', 'John')` — bu metod 'name' kaliti bilan 'John' qiymatini qo‘shadi.

**Get:** Bu metod yordamida ma‘lum bir kalit asosida qiymat olinadi. Agar kalit jadvalda mavjud bo‘lsa, metod unga tegishli qiymatni qaytaradi, aks holda null yoki tegishli xato xabari qaytariladi. `get('name')` — bu metod 'name' kaliti bo‘yicha qiymatni (masalan, 'John') qaytaradi.

**remove:** bu metod 'name' kaliti bilan bog‘liq ma‘lumotni jadvaldan o‘chiradi. Ushbu metod yordamida jadvaldan ma‘lum bir kalitga tegishli qiymatni o‘chirish mumkin. Agar kalit mavjud bo‘lsa, uning bilan bog‘liq qiymat o‘chiriladi.

**containsKey:** Bu metod kalitning mavjudligini tekshiradi. Agar kalit jadvalda mavjud bo'lsa, metod true qaytaradi, aks holda false qaytaradi.

: containsKey:— bu metod 'name' kaliti jadvalda mavjud bo'lsa, true qaytaradi.

**size():** Ushbu metod hash jadvalidagi elementlar sonini qaytaradi. Bu metod yordamida jadvaldagi jami elementlarni bilish mumkin.

size() — bu metod jadvaldagi jami elementlar sonini qaytaradi.

**clear():** Bu metod yordamida hash jadvalidagi barcha ma'lumotlar o'chiriladi

clear() — bu metod jadvaldagi barcha kalit-qiymat juftliklarini o'chiradi.

#### **AFZALLIKLARI:**

**Tezlik:** Hash ma'lumotlar tuzilmasi  $O(1)$  murakkablikda ishlaydi, ya'ni ma'lumotlarni qo'shish, qidirish va o'chirish operatsiyalari juda tez amalga oshiriladi.

**Samardorlik:** Katta hajmdagi ma'lumotlar bilan ishlashda hash tuzilmasi o'zining samaradorligini ko'rsatadi, chunki har bir elementga mos indeks ajratilgan.

**Kalit-qiymat juftliklari:** Ma'lumotlar oson va tushunarli shaklda saqlanadi, har bir elementni kalit orqali osongina topish mumkin.

#### **KAMCHILIKLARI:**

**Koliziyalar:** Hash koliziyasi — bu ikki turli ma'lumotning bir xil hash qiymatiga ega bo'lishi. Bu muammo hash funksiyasining samarali ishlashini qiyinlashtirishi mumkin. Koliziyalarni hal qilish uchun maxsus usullar (masalan, zanjirli ro'yxatlar) qo'llaniladi.

**Xotira resurslari:** Hash jadvali ba'zida juda katta xotira talab qiladi, chunki uning samarali ishlashi uchun yaxshi taqsimlangan jadval kerak.

**Dinamik o'lcham:** Hash jadvalining o'lchamini dinamik ravishda oshirish kerak bo'lishi mumkin, bu esa ba'zida qo'shimcha vaqt va resurslarni talab qiladi.

#### **XULOSA**

Hash ma'lumotlar tuzilmasi dasturlashda ma'lumotlarni samarali saqlash va ularga tezkor kirish imkoniyatini yaratadi. Bu tuzilma keng qo'llaniladi, chunki u juda tez ishlaydi va ko'plab dasturlash tillarida qo'llaniladigan standart metodlarga ega. Hash funksiyasi va uning metodlari yordamida ma'lumotlarni saqlash, olish va boshqarish juda samarali amalga oshiriladi. Biroq, hash koliziyalari va boshqa

noxush holatlar ham yuzaga kelishi mumkin, ammo bu muammolarni hal qilish uchun turli xil yondashuvlar mavjud. Shu boisdan, hash ma‘lumotlar tuzilmasi dasturlashning eng muhim va keng tarqalgan vositalaridan biridir.

### FOYDALANILGAN ADABIYOTLAR:

1) **Knuth, D. E.** (1997). *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley. Ushbu asar, ma‘lumotlarni saqlash va qidirish metodlari, shu jumladan hash ma‘lumotlar tuzilmasi va uning ishlash tamoyillari haqida keng tushuncha beradi.

2) **Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C.** (2009). *Introduction to Algorithms* (3rd ed.). MIT Press. Bu kitob algoritmlar va ma‘lumotlar tuzilmalarini o‘rganishda asosiy manbalardan biridir. Hash tuzilmalariga oid bo‘limlar juda muhim.

3) **Weiss, M. A.** (2013). *Data Structures and Algorithm Analysis in C++* (4th ed.). Addison-Wesley. C++ dasturlash tilida ma‘lumotlar tuzilmalari, shu jumladan hash ma‘lumotlar tuzilmasining metodlari va ularning ishlash prinsiplarini tushuntiradi.

4) **Sedgewick, R., & Wayne, K.** (2011). *Algorithms* (4th ed.). Addison-Wesley. Hash tuzilmalari va boshqa algoritmlar haqida batafsil tushunchalar beradi, bu kitobning algoritmik konsepsiyalari hash funksiyasining ishlashini tushunishda yordam beradi.

5) **Meyers, S.** (2005). *Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library*. Addison-Wesley. Hash ma‘lumotlar tuzilmasining STL (Standard Template Library) yordamida ishlash usullari haqida tushuncha beradi.

6) **Wikipedia Contributors** (2025). *Hash Table*. Wikipedia. [https://en.wikipedia.org/wiki/Hash\\_table](https://en.wikipedia.org/wiki/Hash_table)

Hash ma‘lumotlar tuzilmasining umumiy tushunchasi, ishlash prinsipi, metodlari va turli xil hash algoritmlari haqida ma‘lumot beradi.

7) **Fagin, R., Lotem, A., & Naor, M.** (2000). *Optimal Aggregation Algorithms for Middleware*. ACM Transactions on Database Systems (TODS). Ushbu maqola, hash ma‘lumotlar tuzilmalari va ulardan samarali foydalanish bilan bog‘liq bo‘lgan ilmiy tadqiqotlarni taqdim etadi.

8) **Sarma, A., & Ravichandran, R.** (2017). *Data Structures and Algorithms in Java*. Wiley. Java dasturlash tilida hash ma‘lumotlar tuzilmasi va metodlari haqida qo‘llanma.