

Konteynerlar

Reja:

1. Konteynerlar haqida.
2. Ketma-ket konteynerlar;
3. array sinfi;
4. vector sinfi;
5. deque sinfi;
6. Forward_list sinfi;
7. list sinfi;
8. Xulosa.

PhD: Mallayev O.U.

<https://t.me/daslabkiAqil>

Konteynerlar kutubxonasi

- **Konteynerlar kutubxonasi** - bu dasturchilarga **navbat, ro'yxat va stek** kabi keng tarqalgan ma'lumotlar tuzilmalarini osonlikcha amalga oshirishga imkon beradigan sinf andozalari va algoritmlarning universal to'plamidir. Uch xil konteyner mavjud:
- **ketma-ket konteynerlar,**
- **assosiativ konteynerlar** va
- **tartibga solinmagan assotsiativ konteynerlar.**
- Ularning har biri turli xil operatsiyalar to'plamini qo'llab-quvvatlashga mo'ljallangan.

- Konteyner uning elementlari uchun ajratilgan xotirani boshqaradi va ularga to'g'ridan-to'g'ri yoki iteratorlar orqali kirish uchun funksiyalarni ta'minlaydi (ko'rsatkichlarga o'xshash xususiyatlarga ega obyektlardir).

Ketma-ket konteynerlar

- Sinf a'zolariga ketma-ket kirish huquqiga ega bo'lgan ma'lumotlar tuzilmasini amalga oshirishda ketma-ket konteynerlardan foydalaniladi.
- **Ketma-ket konteynerlarni turlari:**
 - **array** - statik doimiy massiv;
 - **vector** – dinamik doimiy massiv;
 - **deque** - ikki tomonlama navbat;
 - **forward_list** - bog'langan ro'yxat;
 - **list** – ikki tomonlama bog'langan ro'yxat.

array sinfi shablони

- `template<class T,
std::size_t N>
struct array;`

array - N o'lchamdagi massivni o'rab turadigan konteyner.

Ketma-ket konteynerda ishlatiladigan turlar

Tur	Aniqlanishi
value_type	T
size_type	size_t
difference_type	std::ptrdiff_t
reference	value_type&
const_reference	const value_type&
pointer	T*
const_pointer	const T*
iterator	RandomAccessIterator
const_iterator	Константный итератор с произвольным доступом
reverse_iterator	std::reverse_iterator<iterator>
const_reverse_iterator	std::reverse_iterator<const_iterator>

array ning funksiya – a'zolari

Nomi	Izoh
at	Ko'rsatilgan elementga indeks tekshiruvi bilan kirishni ta'minlaydi
operator[]	Belgilangan elementga kirishni ta'minlaydi
front	Birinchi elementga kirishni ta'minlaydi
back	Oxirgi elementga kirishni ta'minlaydi
data (C++11)	Massivning birinchi haqiqiy elementiga ko'rsatgichni qaytaradi

Iteratorlar

Iteratorlar to'plam elementlariga kirishni ta'minlaydi. Iteratorlardan foydalanib, elementlarni takrorlash juda qulay. Iterator turi iterator tomonidan tavsiflanadi. Ammo har bir to'plam uchun iteratorning o'ziga xos turlari mavjud.

Nomi	Izoh
begin, cbegin	Iterator birinchi elementni qaytaradi.
end, cend	Iterator oxirgi elementni qaytaradi.
rbegin, crbegin	Iteratorni birinchi elementga teskarisini qaytaradi.
rend, crend	Oxirgi elementning teskarisini qaytaradi.


array sinfining o'lchamlari va jarayonlari

Nomi	Izoh
empty	Konteynerning qoldirilgan elementlarini tekshirish.
size	Konteyner elementlarining sonini qaytaradi.
max_size	Konteynerdagi elementlarning maksimal miqdorini qaytaradi.
fill	Konteynerni belgilangan qiymatga to'ldirish.
swap	Tarkibni almashtirish.

array sinfi uchun qayta yuklamagan operatorlar

Nomi	Izoh
operator==	Bir qatordagi qiymatlarni leksikografik jihatdan taqqoslaydi
operator!=	
operator<	
operator<=	
operator>	
operator>=	

```
1 #include "stdafx.h"
2 #include <string>
3 #include <iterator>
4 #include <iostream>
5 #include <algorithm>
6 #include <array>
```



```
3 2 1
a b
```

```
8 int main()
9 {
10 // konstruktor agregat boshlang'ichni ishlatadi
11 std::array<int, 3> a1={ {1,2,3} }; // ikki qavs kerak
12 std::array<int, 3> a2 = {1, 2, 3};
13 // tayinlash operatsiyalari bundan mustasno
14 std::array<std::string, 2> a3 = { {std::string("a"), "b"} };
15
16 // umumlashtirilgan algoritmlar qo'llab-quvvatlanadi
17 std::sort(a1.begin(), a1.end());
18 std::reverse_copy(a2.begin(), a2.end(),
19 std::ostream_iterator<int>(std::cout, " "));
20
21 // qo'llab-quvvatlanadigan qatorga qo'yildi
22 std::cout<<std::endl;
23 for(auto& s: a3)
24     std::cout << s << ' ';
25 getchar();
26 }
```

Masala:

- **Talaba** nomli class yaratish. Konteynerlarning array sinfi yordamida to'plamlar yaratish. **Array** ning maxsus metodidan foydalanib, Talabalar to'plamini talabalarning yoshlari bo'yicha saralovchi dastur tuzish.
- **Masalani yechish g'oyasi:**
- talaba nomli yangi nolar fazosi yaratiladi. Uning tarkibiga Talaba classi va Ism, Fam, Yosh, Step nomli to'plamlar joylashtiriladi. Talaba classining ruxsat berilmagan maydonlaridan foydalanish uchun friend funksiya yaratiladi.

```
67 int n 1 - talaba haqida haqida
68     Ismi: Islom
69     Familiyasi: Salimov
70     Yoshi: 22
71     Stependiyasi: 400000
72     2 - talaba haqida haqida
73     Ismi: Oqil
74     Familiyasi: Qodirov
75     Yoshi: 19
76     Stependiyasi: 600000
77     3 - talaba haqida haqida
78     Ismi: Komil
79     Familiyasi: Tashev
80     Yoshi: 20
81     Stependiyasi: 120000
82     Yoshlari bo'yicha saralangan talabalar ro'yxati
83     cout<< "labalar ro'yxati"<<endl;
84     Ismi: Oqil
85     Familiyasi: Qodirov
86     Yoshi: 19
87     Stependiyasi: 600000
88     2 - talaba haqida haqida
89     Ismi: Komil
90     Familiyasi: Tashev
91     Yoshi: 20
92     Stependiyasi: 120000
93     end(),
94     " "));
95     Ismi: Islom
96     Familiyasi: Salimov
97     Yoshi: 22
98     Stependiyasi: 400000
99 }
```

vector sinfi

- **Vector sinfining shablони:**

```
template< class T,  
class Allocator = std::allocator<T>  
> class vector;  
namespace pmr {  
    template <class T>  
        using vector = std::vector<T,  
std::polymorphic_allocator<T>>;  
}
```

- 1) **std :: vector** - o'zgaruvchan kattalikdagi ketma-ket konteynerni qamrab oluvchi massiv.
- 2) **std :: pmr :: vector** shablon taxalluslari polimorf ajratuvchi yordamida.
- **Allocator** - elementlar uchun xotira ajratishda ishlatiladigan sinf.

vector sinfining funksiya – a'zolari

Nomi	Izoh
at	Ko'rsatilgan elementga indeks tekshiruvi bilan kirishni ta'minlaydi
operator[]	Belgilangan elementga kirishni ta'minlaydi
front	Birinchi elementga kirishni ta'minlaydi
back	Oxirgi elementga kirishni ta'minlaydi
data (C++11)	Massivning birinchi haqiqiy elementiga ko'rsatgichni qaytaradi
operator=	Konteynerdagi qiymatlarni o'rnatadi
assign	Konteynerdagi qiymatlarni o'rnatadi

vector sinfining modifikatorlari

Nomi	Izoh
clear	Konteynerni tozalaydi.
insert	Konteynerga element qo'shadi.
emplace (C++11)	Elementlarni "joyida" quradi va berilgan pozitsiyadan boshlab ularni joylashtiradi.
erase	Konteynerdan element ochirish.
push_back	Oxiriga element qo'shadi.
emplace_back (C++11)	Konteyner oxiridan elementlarni qo'shadi.
pop_back	Oxirgi elementni o'chirish.
resize	Saqlangan elementlar sonini o'zgartiradi.
swap	Tarkibni almashtirish.

vector sinfi uchun qayta yuklanmagan operatorlar

Nomi	Izoh
<code>operator==</code>	Bir qatordagi qiymatlarni leksikografik jihatdan taqqoslaydi
<code>operator!=</code>	
<code>operator<</code>	
<code>operator<=</code>	
<code>operator></code>	
<code>operator>=</code>	


```
1 #include "stdafx.h"
2 #include <iostream>
3 #include <vector>
4
5 int main ( ) {
6     // Butun sonlarni o'z ichiga olgan vektor
7     int A[] = {7, 5, 16, 8};
8     std::vector<int> v(4);
9     v[0] = A[0];
10    v[1] = A[1];
11    v[2] = A[2];
12    v[3] = A[3];
13    // Vektorga yana ikkita butun sonni qo'shish
14    v.push_back(25);
15    v.push_back(13);
16
17    // Qiymatlar chiqishi uchun vektor orqali o'tish
18    for ( int n : v ) {
19        std::cout << n << '\n';
20    }
21    getchar();
22 }
```

```
7
5
16
8
25
13
```

- **Masala:** Berilgan int turidagi to'plamda – ishorali elementlarning eng kattasini o'chiruvchi dastur tuzing.
- **Masalani yechish g'oyasi:**
- **talaba** nomli yangi nomlar fazosi yaratiladi. Uning tarkibiga **Talaba** classi va **Ism, Fam, Yosh, Step** nomli to'plamlar joylashtiriladi. Talaba classining ruxsat berilmagan maydonlaridan foydalanish uchun **friend** funksiya yaratiladi.

```

24 void Max_Vector(std::vector<int> &A, int n){
25     for (int i = 0; i < n; i++)
26     {
27         if(A[i] < 0 && max < A[i]) {max = A[i]; index=i;}
28     }
29     Berilgan int turidagi to'plamda - ishorali
30     elementlarning maksimalini o'tchiruvchi dastur
31     5
32     1 son: 4
33     2 son: -55
34     3 son: -3
35     4 son: -4
36     5 son: 23
37     To'plam elementlari
38     4 -55 -3 -4 23
39     Max= -3
40     To'plam elementlari
41     4 -55 -4 23
42
43     Vectorlar::Max_Vector(A,n);
44     Vectorlar::Delete_Max_Vector(A);
45     Vectorlar::get_Vector(A,n);
46     getchar();
47 }

```

deque sinfi

- Deque sinfining shabloni:
- `template<class T,`
- `class Allocator = std::allocator<T>`
- `> class deque;`
- `std :: deque` (ikki tomonlama navbat) - indekslangan ketma-ket konteyner, bu sizga elementlarni boshidan va oxiridan tezda kiritish va olib tashlash imkonini beradi. Bundan tashqari, ikki tomonlama navbatning ikkala uchiga o'rnatish va o'chirish ko'rsatgichlar va boshqa elementlarga bog'lanishlarni qoldiradi.
- `std :: vektor`dan farqli o'laroq, `deque` elementlari doimiy ravishda saqlanmaydi: odatda bu belgilangan o'lchamdagi ajratilgan qatorlar to'plamidan foydalanib amalga oshiriladi. `deque` avtomatik ravishda qayta ishlanadi, kerak bo'lganda kengayadi. `deque` kengaytmasi `std :: vector` kengaytmasiga qaraganda qulayroq, chunki u mavjud elementlarni yangi xotiraga nusxalashni talab qilmaydi.

deque sinfining funksiya – a'zolari

Nomi	Izoh
at	Ko'rsatilgan elementga indeks tekshiruvi bilan kirishni ta'minlaydi
operator[]	Belgilangan elementga kirishni ta'minlaydi
front	Birinchi elementga kirishni ta'minlaydi
back	Oxirgi elementga kirishni ta'minlaydi
get_allocator	Bog'langan ajratuvchini qaytaradi
operator=	Konteynerdagi qiymatlarni o'rnatadi
assign	Konteynerdagi qiymatlarni o'rnatadi

deque sinfining modifikatorlari

Nomi	Izoh
clear	Konteynerni tozalaydi.
insert	Konteynerga element qo'shadi.
emplace (C++11)	Elementlarni "joyida" quradi va berilgan pozitsiyadan boshlab ularni joylashtiradi.
erase	Konteynerdan element ochirish.
push_back	Oxiriga element qo'shadi.
emplace_back (C++11)	Konteyner oxiridan elementlarni qo'shadi.
pop_back	Oxirgi elementni o'chirish.
resize	Saqlangan elementlar sonini o'zgartiradi.
swap	Tarkibni almashtirish.
push_front	Ro'yxatning boshiga elementlarni joylashtiradi.
emplace_front (C++11)	Ro'yxatning boshidan boshlab, elementlarni yaratadi.
pop_front	Birinchi elementni o'chirish.

deque sinfi uchun qayta yuklanmagan operatorlar

Nomi	Izoh
operator==	Bir qatordagi qiymatlarni leksikografik jihatdan taqqoslaydi
operator!=	
operator<	
operator<=	
operator>	
operator>=	

- **Masala:**

- Berilgan **string** turidagi to'plamda eng uzun so'zning belgilar sonini aniqlovchi dastur tuzing.

- **Masalani yechish g'oyasi:**

- **deque** nomli yangi nolar fazosi yaratiladi. Uning tarkibiga **Ism**, **Fam**, **Yosh**, **Step** nomli to'plamlar joylashtiriladi. Talaba classining ruxsat berilmagan maydonlaridan foydalanish uchun **friend** funksiya yaratiladi.


```
51 int main()
52 { Berilgan string turidagi to'plamda eng
53   uzun so'zni belgilar sonini aniqlovchi dastur
54   To'plam elementlari soni n= 3
55   1 - satrni kiriting:
56   Salom dunyo
57   2 - satrni kiriting:
58   Hello world
59   3 - satrni kiriting:
60   Assalomu aleykum
61   To'plam elementlari
62   1 - satr:
        Salom dunyo
        2 - satr:
        Hello world
        3 - satr:
        Assalomu aleykum

        Salom dunyo Hello world Assalomu aleykum

        So'zlarning uzunliklari:
        5 5 5 5 8 7 Max= 8
```

```
<<endl;
cin.ignore
```

forward_list sinfi

- **forward_list** sinfining shabloni:
- **template <class T,**
class Allocator = std::allocator<T>
- **> class forward_list;**
- **forward_list** - konteynerdan elementlarni kiritish va olib tashlash mexanizmini ta'minlaydigan **sinf**. Tez tasodifiy kirish qo'llab-quvvatlanmaydi. U bir yo'naltirilgan ro'yxat sifatida amalga oshiriladi va C tilidagi shunga o'xshash dastur bilan solishtirganda qo'shimcha xarajatlarga ega emas: **std :: list** dan farqli o'laroq, ushbu turdagi konteyner ikki tomonlama iteratsiyani qo'llab-quvvatlamaydi.

forward_list sinfining o'lchamlari va jarayonlari

Nomi	Izoh
merge	Ikkita tartiblangan ro'yxatlarni birlashtirish.
splice_after	Elementlarni boshqa forward_listdan ko'chiradi.
remove remove_if	Ma'lum belgilarga javob beradigan elementlarni olib tashlaydi.
reverse	Elementlarning tartibini o'zgartiradi.
unique	Ketma-ket takrorlanadigan elementlar o'chiriladi.
sort	Elementlarni tartiblash.

- **Masala:**

- Berilgan **int** turidagi to'plam qiymatlarining raqamlari yig'indisini **Z** to'plamga joylashtiruvchi va ularni ekranga chiqaruvchi dastur tuzing.

- **Masalani yechish g'oyasi:**

- **forward_list** konteyneri to'plami yaratiladi. **forward_list** ning **push_front()** iteratoridan foydalanib, **A** to'plamga qiymatlar o'zlashtiriladi. **Z** to'plamga esa **A** to'plam qiymatlarini raqamlari yig'indisi yoziladi.

```

25 int main()
26 {
27     cout << "Forward_list"
28     int 5
29     1 son: 222222
30     std 2 son: 5555
31     std 3 son: 3333
32     Set 4 son: 999
33     5 son: 111111
34     aut A to'plam qiymatlari
35     int 1111111
36     cout << 1111111 << endl;
37     while (999
38         3333
39         5555
40         222222
41     s+
42     } Z to'plam qiymatlari
43     cout << 12 << endl;
44     for (20; i++)
45     { 12
46         27
47     }
48     get
49 }

```

list sinfi

- **list sinfi shablони:**

```
template < class T,  
          class Allocator = std::allocator<T>  
> class list;
```

- **List** - bu konteynerning har qanday pozitsiyasidan elementlarni tezda kiritish va olib tashlashni qo'llab-quvvatlaydigan sinf. Tez tasodifiy kirish qo'llab-quvvatlanmaydi. Ikkala bog'langan ro'yxat sifatida amalga oshiriladi. **std::forward_list**-dan farqli o'laroq, ushbu konteyner ikki tomonlama iteratsiyani ta'minlaydi, shu bilan birga foydalanilgan xotiraga nisbatan unumli emas.

- **Masala:**

- Berilgan **int** turidagi to'plam qiymatlarini juftlarini **Z** to'plamga joylashtiruvchi va ularni ekranga chiqaruvchi dastur tuzing.

- **Masalani yechish g'oyasi:**

- **list** konteyneri to'plami yaratiladi. **list** ning **push_back()** iteratoridan foydalanib, **A** to'plamga qiymatlar o'zlashtiriladi. **Z** to'plamga esa **A** to'plam qiymatlarini juftlari yoziladi.

```
17 int main() {
18     cout << "Enter 5 numbers: ";
19     int i;
20     for (i = 1; i <= 5; i++)
21         set.insert(i);
22     Set A(1, 2, 3, 4, 5);
23     Set Z(2, 3, 4, 5);
24     auto it = A.begin();
25     int sumA = 0;
26     while (it != A.end())
27         sumA += *it;
28     cout << "Sum of A: " << sumA << endl;
29     auto it2 = Z.begin();
30     int sumZ = 0;
31     while (it2 != Z.end())
32         sumZ += *it2;
33     cout << "Sum of Z: " << sumZ << endl;
34     for (i = 1; i <= 5; i++)
35         cout << i << " ";
36     cout << endl;
37     return 0;
38 }
39 }
```

```
Forward_list
5
1 son: 22
2 son: -33
3 son: -2
4 son: 334
5 son: -56
A to'plam qiymatlari
-56
334
-2
-33
22
Z to'plam qiymatlari
4
-33
-2
10
-56
```

```
endl;
();
ri" << endl;
ri" << endl;
=Z.end(); i++){
```


Foydalanilgan adabiyotlar

- <https://ru.cppreference.com/w/cpp/container>
- <https://ru.cppreference.com/w/cpp/container/array>
- <https://ru.cppreference.com/w/cpp/container/vector>
- <https://ru.cppreference.com/w/cpp/container/deque>
- https://ru.cppreference.com/w/cpp/container/forward_list
- <https://ru.cppreference.com/w/cpp/container/list>

Xulosa

C++ da 3 xil konteynerlar mavjud:

- **ketma-ket** konteynerlar,
- **assosiativ** konteynerlar va
- **tartibga solinmagan assotsiativ** konteynerlar.

- **Ketma-ket konteynerlar:**

array - statik doimiy massiv;

vector – dinamik doimiy massiv;

deque - ikki tomonlama navbat;

forward_list - bog'langan ro'yxat;

list – ikki tomonlama bog'langan ro'yxat.