

C++ Dersi: Nesne Tabanlı Programlama

2. Baskı



Bölüm 17: `string` Sınıfı

Çiğdem Turhan
Fatma Cemile Serçe

İçerik



17.1 string Yapıcı Fonksiyonları

17.2 string ile Girdi/Çıktı

17.3 string Operatörleri

17.4 string Üye Fonksiyonları

17.5 string Nesnesine Sayı Değeri Ekleme

Çözümlü Sorular



Hedefler

- Farklı `string` yapıcı fonksiyonlarını kullanma
- `cin` veri yolu ile `string` verilerini okuma
- `getline()` fonksiyonunun kullanım amacını anlatma
- `string` değerinde yer alan karakterlere erişme
- Verilen iki `string` nesnesini birleştirme
- `string` nesnelere başka `string` nesneleri ekleme
- Verilen `string` nesnesinin uzunluğunu bulma
- Bir `string` nesnesinin verilen indeksteki karakterine erişme
- `string` nesnesinin içeriğini temizleme
- Verilen iki `string` nesnesini karşılaştırma
- Bir `string` nesnesini, başka bir `string` nesnesine kopyalama
- Bir `string` nesnesinin içeriğinin bir bölümünü silme
- Bir `string` nesnesi içerisinde başka bir `string` nesnesini ya da karakteri arama
- Bir `string` nesnesinin bir bölümünü başka bir `string` nesnesi ile değiştirme
- Bir `string` nesnesine sayı ekleme

string

- Birden fazla karakter içeren veri tipidir
- `<string>` kütüphanesinde yer alır
- Bir string nesnesini aşağıdaki şekilde tanımlayabiliriz:
`string kelime;`
- Bu tanımla, string sınıfının varsayılan yapıcı fonksiyonu kelime nesnesine boş bir dizgi atar.

17.1 string Yapıcı Fonksiyonları

Tablo 17.1 string Yapıcı Fonksiyonları

string Nesnesi	Açıklama
<code>string kelime;</code>	kelime isimli boş bir string nesnesi tanımlanır.
<code>string kisi("Ali Ozkan");</code>	İlk değeri "Ali Ozkan" olan kisi isimli bir string nesnesi tanımlanır.
<code>string kisi="Ali Ozkan";</code>	İlk değeri "Ali Ozkan" olan kisi isimli bir string nesnesi tanımlanır.
<code>string n1(n2);</code>	n1 isimli string nesnesi oluşturulur ve n2 nesnesi n1'in içine kopyalanır. n2 bir string nesnesi veya karakter dizgisi olabilir.
<code>string parca(cumle,5);</code>	parca isimli string nesnesi tanımlanır ve cumle isimli karakter dizgisinin 5. karakterinden itibaren içine kopyalanır.
<code>string dolu(6 , 'X');</code>	dolu isimli string nesnesi tanımlanır ve içine 6 adet 'X' karakteri kopyalanır.
<code>string parca(cumle,0,5);</code>	parca isimli string nesnesi oluşturulur ve cumle isimli string nesnesinin 0. karakterinden başlayarak 5 karakter içine kopyalanır.

17.2 string ile Girdi/Çıktı

- Diğer veri tiplerinde kullanıldığı gibi cin standart veri yolu ile string tipindeki değişkenleri okuyabiliriz. Bu tipteki değişkenler okunurken baştaki boşluklar atlanır ve kelimenin sonunda yer alan boşluğa kadar okunur.

Örnek 17.1

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string bir, iki, uc;
    cin>>bir>>iki>>uc;           // Üç string nesnesi okunur
    return 0;
}
```

Çıktı

Bu kose yaz kosesi bu kose kis kosesi_

17.2 string ile Girdi/Çıktı...

- **getline()**
- string nesnesine tüm satırı, '\n' e (enter) görene kadar okur. cin komutu ile boşluğa kadar okuyabildiğimiz için bir cümle veya satır okumamız gerektiğinde bu fonksiyon kullanılır.

Örnek

```
string satir;  
getline(cin,satir);           // Tüm satır okunur  
cout<<satir<<"okundu";
```

Örnek Girdi

Bu kose yaz kosesi ↵

Örnek Çıktı

Bu kose yaz kosesi okundu

Örnek

```
int no;  
string satir;  
cin>>no;           // Tamsayı okunur  
getline(cin,satir); // Tüm satır okunur
```

Örnek Girdi

100 ↵
Ali topu at ↵

17.3 string Operatörleri

- `+` : birleştirme operatörü ile iki `string` nesnenin içeriğini birleştirir.
- `=` : atama operatörü ile eşitliğin sağında bulunan değerler, eşitliğin solunda bulunan `string` nesnesine atanır.

Örnek 17.2

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string str1, str2, str3;
    str1 = "Merhaba";           // Karakter dizgisi atanır
    str2 = 'A';                 // Tek karakter atanır
    str3 = str1 + str2;         // + operatörü ile iki string birleştirilir
    cout<<str3<<endl;
    return 0;
}
```

Çıktı

MerhabaA

17.3 string Operatörleri...

- **+=** : bileşik atama komutu ile eşitliğin sağ tarafında yer alan string nesnesi ile sol tarafında yer alan string nesnesi birleştirilerek sol tarafta yer alan string nesnesine atanır.

Örnek 17.3

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str = "Merhaba";
    str += " C++ ";           // Merhaba ve C++ birleştirilir
    cout<<str<<endl;
    return 0;
}
```

Çıktı

Merhaba C++

17.3 string Operatörleri...

- `[]` : indeksli veri erişimi operatörü ile string nesnesinin, belirtilen indeksteki karakteri döndürülür.

Örnek 17.4

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string str;
    str = "ABC";
    cout<<str[0]<<endl; // 0. indeksteki karakter, A
    cout<<str[1]<<endl; // 1. indeksteki karakter, B
    cout<<str[2]<<endl; // 2. indeksteki karakter, C
    return 0;
}
```

Çıktı

A
B
C

17.4 string Üye Fonksiyonları

- **length():** bu fonksiyon bir string nesnesinin karakter uzunluğunu döndürür.

Örnek 17.5

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str = "C++ Programlama";
    cout <<"Uzunluk:"<<str.length()<< endl;    // Karakter sayısını yazar
    return 0;
}
```

Çıktı

Uzunluk:15

17.4 string Üye Fonksiyonları...

- **append(...):** Bu fonksiyon ile string nesnesinin içeriğine ekleme yapılır. Bu fonksiyonda nesnenin içerisindeki bilgiler silinmez. Farklı parametre listesi ile append() fonksiyonuna aşağıdaki gibi yüklemeler yapılmıştır.
 - obje.append(n, 'c') :obje nesnesine n sayıda 'c' karakterini ekler.
 - obje.append(str):obje nesnesine, parametre olarak gönderilen str nesnesinin içeriğini ekler.
 - obje.append(str, x, n) :str nesnesinin x. karakterinden başlayıp n adet karakteri obje nesnesine ekler.

17.4 string Üye Fonksiyonları...

- `append(...)`...

Örnek 17.6

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str1 = "C++ ";
    string str2 = "programlama ";
    string str4 = "ve bu cok eglenceli";
    str1.append(str2);           // C++ programlama
    str1.append("ogreniyorum",5); // C++ programlama ogren
    str1.append(3, '.');         // C++ programlama ogren...
    str1.append(str4,6,13);      // C++ programlama ogren...cok eglenceli
    cout<<str1<<endl;
    return 0;
}
```

Çıktı

C++ programlama ogren...cok eglenceli

17.4 string Üye Fonksiyonları...

- **assign(...)** : string nesnesinin içeriği başka bir string nesnesinin içeriği ile değiştirilir ve daha önceki içerik silinir. Farklı parametre listesi ile assign() fonksiyonuna aşağıda anlatıldığı gibi yüklemeler yapılmıştır.
 - obje.assign(str) : obje nesnesinin içeriğine str nesnesinin içeriği aktarılır.
 - obje.append(str, n) : obje nesnesine str nesnesinin ilk n karakterinden oluşan içerik aktarılır.
 - obje.assign(str, x, n) : obje nesnesine str nesnesinin x. karakterinden başlayıp n sayıda karakter atanır.
 - obje.assign(n,'c') : n adet 'c' karakteri obje nesnesine atanır.

17.4 string Üye Fonksiyonları...

- assign(...)

Örnek 17.7

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str;
    string cumle="Ankara Turkiye'nin baskentidir.";
    str.assign(cumle);
    cout<<str<< endl;           // Ankara Turkiye'nin baskentidir.
    str.assign(cumle,7,7);
    cout<<str<<endl;           // Turkiye
    str.assign("Ankara ic anadolu bolgesindedir.",6);
    cout<<str<< endl;           // Ankara
    str.assign("Anitkabir");
    cout<<str<<endl;           // Anitkabir
    str.assign(9, '.');
    cout<<str<<endl;           // .....
    return 0;
}
```

Çıktı

```
Ankara Turkiye'nin baskentidir.
Turkiye
Ankara
Anitkabir
.....
```

17.4 string Üye Fonksiyonları...

- **at(x)** : obje nesnesinin x. karakterini döndürür. [] indeks erişim operatörü gibi çalışır ancak, yanlış indekse erişilmek istendiğinde kural dışı durum olarak kabul edip hata fırlatır.

Örnek 17.8

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{   string str ("C++ Programlama");
    int i;
    for (i=0; i < str.length(); i++){ // str'nin uzunluğu kadar döner
        cout<<str.at(i)<< "_";       // str'nin içindeki karakterleri yazar
    }
    return 0;
}
```

Çıktı

C_+_+_P_r_o_g_r_a_m_l_a_m_a_

17.4 string Üye Fonksiyonları...

- `at(x)` ...

Örnek 17.9

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str ("C++ Programlama");
    try{
        cout << str.at(20);    // Geçersiz indeks
    } catch(...){
        cout<<"Gecersiz indeks erisimi"<<endl;
    }
    return 0;
}
```

Çıktı

Gecersiz indeks erisimi

17.4 string Üye Fonksiyonları...

- **clear()** : string nesnesinin içeriğini siler ve 0 karaktere sahip olacak şekilde temizler.

Örnek 17.10

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{   string str ("Bu bir ornek cumledir.");
    cout<<"icerik:"<<str<<endl;
    str.clear();                      // str'nin içeriğini siler
    cout<<"icerik:"<<str<<endl;
    return 0;
}
```

Çıktı

```
icerik:Bu bir ornek cumledir.
icerik:
```

17.4 string Üye Fonksiyonları...

- **compare(...)** : string nesnesinin içeriğini parametre olarak gelen string nesnelerinin içeriği ile karşılaştırır. İçerikler aynı ise '0' değil ise '0' dışında bir sayı döndürür. Farklı parametre listeleri ile compare() fonksiyonuna aşağıda anlatıldığı gibi yüklemeler yapılmıştır.
 - obje.compare(str):obje nesnesinin içeriği ile str nesnesinin içerikleri karşılaştırılır.
 - obje.compare(x, n, str):obje nesnesinin içeriğinin x. karakterinden başlayıp n adet karakterinden oluşan kısmı ile str nesnesini karşılaştırır.

17.4 string Üye Fonksiyonları...

- **compare(...)** ...

Örnek 17.11

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str1 ("siyah renk");
    string str2 ("mavi renk");

    if (str1.compare(str2) != 0)           // str1 ve str2 karşılaştırılır
        cout<<str1<<" ve "<<str2<<" aynı renk degildir."<<endl;
    else
        cout<<str1<<" ve "<<str2<<" aynı renkler."<<endl;

    if (str1.compare(6,4,"renk") == 0) // str1 ve "renk" karşılaştırılır
        cout<<str1<<" bir renktir"<<endl;
    if (str2.compare(5,4,"renk") == 0) // str2 ve "renk" karşılaştırılır
        cout<<str2<<" bir renktir"<<endl;
    return 0;
}
```

Çıktı

```
siyah renk ve mavi renk aynı renk degildir.
siyah renk bir renktir
mavi renk bir renktir
```

17.4 string Üye Fonksiyonları...

- **copy(d, n, x):** string nesnesinin içeriğinde x. karakterden başlayarak n karakteri, d karakter dizisine kopyalar ve dizinin uzunluğunu döndürür. Dizinin sonuna '\0' karakteri eklenmez.

Örnek 17.12

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    int length;
    char dizi[10];
    string str("Bu bir ornek cumledir.");
    // str'nin bir kısmı dizi'ye kopyalanır ve uzunluğu atanır
    length=str.copy(dizi,5,7);
    dizi[length]='\0';
    cout<<"Karakter dizisinin icerigi:"<<dizi<<endl;
    return 0;
}
```

Çıktı

Karakter dizisinin icerigi:ornek

17.4 string Üye Fonksiyonları...

- **empty()** : string nesnesinin içeriğinin boş olup olmadığını döndürür. string nesnesinin içeriğini temizlemez.

Örnek 17.13

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string icerik;
    string satir;
    cout<<"*** Lütfen iceriginizi giriniz"<<endl;
    cout<<"*** Sonlandirmak icin bos satir giriniz.\n";
    do {
        getline(cin,satir);           // Tüm satır okunur
        icerik += satir + '\n';       // Sonuna \n eklenir
    } while (!satir.empty());         // Satır boş okunana kadar devam eder
    cout<<"*** Girmis oldugunuz icerik:\n"<<icerik<<endl;
    return 0;
}
```

Çıktı

```
*** Lütfen iceriginizi giriniz
*** Sonlandirmak icin bos satir giriniz.
C++ programlama
Gostergeler
Siniflar

*** Girmis oldugunuz icerik:
C++ programlama
Gostergeler
Siniflar
```

17.4 string Üye Fonksiyonları...

- **erase(x,n)** : string nesnesinin x. karakterinden itibaren n karakterini siler.

Örnek 17.14

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str("C++ Programlama");
    str.erase(3,12);           // str1'nin bir kısmını siler
    cout<<str<<endl;
    return 0;
}
```

Çıktı

C++

17.4 string Üye Fonksiyonları...

- **find(...)** : string nesnesinin içeriğinde, gönderilen string ya da karakter parametre değerlerini arar. Eğer aranan değerler bulunduysa, bulunan karakter'in ya da string'in ilk karakterinin, string içindeki indeks numarası döndürülür. Eğer arama sonucunda bulma işlemi gerçekleşmemişse, string sınıfına ait npos statik üye döndürülür. find() fonksiyonuna aşağıda verildiği gibi yüklemeler yapılmıştır.
 - obje.find(str): obje nesnesi içerisinde str değeri aranır.
 - obje.find(str, x): obje nesnesi içerisinde, x. karakterden başlayarak str değeri aranır.
 - obje.find('c',x): obje nesnesi içerisinde, x. karakterden başlayarak 'c' karakteri aranır.

17.4 string Üye Fonksiyonları...

- `find(...)` ...

Örnek 17.15

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string icerik("C++ nesne tabanlı bir programlama dilidir.");
    string kelime("programlama");
    int bulundu;

    bulundu = icerik.find(kelime);    // icerik icinde kelime aranır
    if (bulundu != string::npos)
        cout << "'programlama' kelimesi icerikte bulundu."<<endl;

    bulundu = icerik.find("bir");    // icerik icinde "bir" aranır
    if (bulundu != string::npos)
        cout << "'bir' kelimesi icerikte bulundu."<<endl;

    // icerik icinde "bir" 22. karakterden sonra aranır
    bulundu = icerik.find("bir",22);
    if (bulundu != string::npos)
        cout << "'bir' kelimesi icerikte bulundu."<<endl;
    else
        cout << "'bir' kelimesi icerikte bulunamadi."<<endl;
    return 0;
}
```

Çıktı

```
'programlama' kelimesi icerikte bulundu.
'bir' kelimesi icerikte bulundu.
'bir' kelimesi icerikte bulunamadi.
```

17.4 string Üye Fonksiyonları...

- **insert(...)** : string nesnesini başka string değerleri ekleyerek genişletir. string nesnesi içinde herhangi bir yere ekleme yapılabilir. insert() fonksiyonuna aşağıda görüldüğü gibi farklı parametre listeleri ile yüklemeler yapılmıştır.
 - obje.insert(x,n,'z'): obje nesnesinin x. karakter indeksinden başlayarak 'z' karakteri n kere eklenir.
 - obje.insert(x,str): obje nesnesinin x. karakter indeksine str nesnesi eklenir.

17.4 string Üye Fonksiyonları...

- insert(...) ...

Örnek 17.16

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{   string str1="Turkiye'nin Ankara'dir";
    string str2=" baskenti ";
    str1.insert(11,str2);           // str1'in 11. indeksine str2 eklenir
    cout << str1 << endl;
    str1.insert(32,3,'. ');        // str1'in 32. indeksine '.' 3 kere eklenir
    cout << str1 << endl;
    return 0;
}
```

Çıktı

```
Turkiye'nin baskenti Ankara'dir
Turkiye'nin baskenti Ankara'dir...
```

17.4 string Üye Fonksiyonları...

- **replace(...)** : string nesnesinin bir kısmı, başka bir string nesnesi ile değiştirilir. Farklı parametrelerle aşağıda verildiği gibi replace() fonksiyonunun üzerine yüklemeler yapılmıştır.
 - obje.replace(x,n,str): obje nesnesinin x. karakterinden başlayarak n adet karakteri, str nesnesi ile değiştirir.
 - obje.replace(x,n,str,x2,n2): obje nesnesinin x. karakterinden başlayarak n adet karakteri, str nesnesinin x2. karakterinden başlayarak n2 adet karakteri ile değiştirir.

17.4 string Üye Fonksiyonları...

- `replace(...)` ...

Örnek 17.17

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str1="Java nesne tabanlı bir programlama dilidir";
    string str2="C++";
    string str3="guclu bir";
    cout << str1 << endl;
    str1.replace(0,4,str2);           //str1'in 0-4 indeksleri str2 ile değiştirilir
    cout << str1 << endl;
    //str1'in 4-13 indeksleri str3'ün bir kısmı ile değiştirilir
    str1.replace(4,13,str3,0,5);
    cout << str1 << endl;
    return 0;
}
```

Çıktı

```
Java nesne tabanlı bir programlama dilidir
C++ nesne tabanlı bir programlama dilidir
C++ guclu bir programlama dilidir
```

17.4 string Üye Fonksiyonları...

- **substr(x,n)** : string nesnesinin x. karakterinden başlayarak n karakteri içeren string değerini döndürür.

Örnek 17.18

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str1 ="Bu ornek bir C++ programidir.";
    string str2;
    str2 = str1.substr (13,3); //str1'in 13. indeksten başlayarak 3 karakteri atanır
    cout<<str2<< endl;
    return 0;
}
```

Çıktı

C++

17.4 string Üye Fonksiyonları...

- **swap(str):** string nesnesinin içeriği ile str nesnesinin içeriği değiş tokuş yapılır.

Örnek 17.19

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str1 ("siyah");
    string str2 ("beyaz");
    cout<<str1<<endl;
    cout<<str2<<endl;
    str1.swap (str2);           //str1 ve str2 değiş tokuş yapılır
    cout<<str1<<endl;
    cout<<str2<<endl;
    return 0;
}
```

Çıktı

```
siyah
beyaz
beyaz
siyah
```

17.5 string Nesnesine Sayı Değeri Ekleme

- Bir string nesnesine, `+` operatörü ya da `append()` fonksiyonu ile başka string nesnesi eklenebilir. Ancak aynı yöntemlerle bir string nesnesine bir sayı değerini eklemek mümkün değildir.
- Bu amaca yönelik uygulanabilecek çözüm yollarından bir tanesi, diğer veri türlerini string nesnesine çeviren `sstream` başlık dosyasında bulunan `stringstream` sınıfını kullanmak olacaktır.
- Bu sınıfta tanımlı `<<` çıktı operatörü kullanılarak, sayının string karşılığını tutan bir nesne tanımlanır. Daha sonra bu nesnenin içerisindeki string bilgisini çekmek için `str()` fonksiyonu kullanılır.

17.5 string Nesnesine Sayı Değeri Ekleme...

Örnek 17.20

```
#include <string>
#include <sstream>
#include <iostream>
using namespace std;
int main()
{
    string str = "C++";
    int versiyon = 2011;
    stringstream ss;
    ss<<versiyon;
    str += ss.str();           // Okunan sayının string karşılığı str'e eklenir
    cout<<str<<endl;
    return 0;
}
```

Çıktı

C++2011