2023.10.02

String sınıfı

String Elemanlarına Erişme:

```cpp
#include <string>

int main()
{
    using namespace std;

    string str{ "burak kose" };

    for (size_t i{}; i < str.length(); ++i)
    {
        cout.put(str[i]);
    }

    str[2] = 's';
    // str[12] tanımsız  davranış

    try
    {
        auto c = str.at(345); // exception verir at fonksiyonunda
    }
    catch (const std::exception& ex)
    {
        std::cout << "exception caught: " << ex.what() << '\n';
    }

    str.front() = 'd'; // ilk
    str.back() = 'i'; // son
}
```

Range-Based For Loop:

```cpp
// range-based for loop

int main()
{
    using namespace std;

    string str {"bugun gunlerden pazartesi kirmizi pazartesi"};

    // range-based for loop

    for (auto iter = str.begin(); iter != str.end(); ++iter)
    {
        cout << *iter << " ";
    }
    // derleyici yukarıdaki gibi kod yazar
    for (auto i : str)
    {
        cout << i << " " ;
    }
}
```

```
/*
    for (auto i : a)  // kopyalama semantiği
    for (auto &i : a) // referans semantiği ( değeiştirme yapacaksak)
    for (const auto i : a) // kopyalama semantiği
    for (const auto &i : a) okuma yapacaksak


    for (auto iter = begin(con); iter != end(con); ++iter)
    {
        auto i = *iter;
        auto &i = *iter;
        const auto i = iter;
        const auto &i = iter;
    }

*/
```

```
int main()
{
    string str { "kirmizi pazartesi" };
    for (auto c : str)
    {
         c = "!"
    }

    cout << str << "\n"; // yazı değişmez

    for (auto &c : str)
    {
         c = "!"
    }

    cout << str << "\n"; // yazı değişir
}
```

Dikkat !

std::string bir STL kabıdır ve iterator arayüzüne sahiptir

String Sınıfı Atama İşlemleri

```
std::string get_name()
{
    return "kerim denizoglu";
}

int main()
{
    using namespace std;
    string name = "tayyip erguder";
    string str {" mustafa "};

    cout << "|" << str << "|\n";

    str = name; // copy assignment
    cout << "|" << str << "|\n";

    // move assignment çünkü  ='in sağ tarafı r value expr
    str = get_name();
    cout << "|" << str << "|\n";

    str = "alican"; // cstring atama operator fonksiyonu
    cout << "|" << str << "|\n";

    str = 'X'; // char parametreli atama operator fonksiyonu
    cout << '|' << str << "|\n";

    str = {'a', '1', 'i' }; // init list parametreli atama operator fonksiyonu
    cout << '|' << str << "|\n";


    name.resize(20); // boyutu büyür null karakter koyarlar
    name.resize(20, '.')
    name.resize(2); // silme argümanı olarak kullanabiliriz


    name.pushback(".")

}
```

String Sınıfı Container Silme

```
int main()
{
    string name = "tayyip erguder";
    // container silme
    name.clear();
    name.resize(0);
    name = "";
    name = {};
    name = string{}; // default ctor
}
```

String Sınıfı Yazı Ekleme

```
int main()
{
    string str {"emre bahtiyar"};

    str.push_back(".");
    str += "gano";
    str += 's';
    str += {"1", "9"};


    string str;
    string s = "neslihan";

    str.assing(s, 3); // lihan
    str.assing(s, 1, 3); //esl
}
```

String Sınıfı Arama Fonksiyonları

static constexpr string::size_type npos = -1

```
int main()
{
    //size_t' in en büyük değeri
    std::cout << "nops = " << std::srting::nops;


    constexp auto x = std::string::nops;
    std::string::npos = 57687u // legal değil npos const

    /*
        string sınıfın arama fonksiyonları indeks döndürürler
        bu değer index bulunamaz ise
        std::string::npos olur
    */

    /*
        Arama fonksiyonları
        find
        rfind
        find_first_of
        find_last_Of
        find_first_not_of
        find_last_not_of

        hepsi std::string::size_type döndürür
        eğer aranan bulunamazsa hepsi std::string::npos döndürür

    */

    string str = "emre bahtiyar";
    char c = 'a';

    // kodu böyle yazmak scope leak neden olur
    string::size_type idx = str.find(c);
```

```
    if (idx != string::npos)
    {
        //code
    }
    // cpp 17 ile böyle
    if (string::size_type idx = str.find(c); idx != string::npos)
    {
        //code
    }

    // cpp 17'den eskilerde
    {
        string::size_type idx = str.find(c);
        if (idx != string::npos)
        {
            //code
        }
    }

    /*
        Cpp 20 ile:
            yazı aradağım varlıkla mi baslıyor
            yazı aradağım varlıkla mi bitiyor

            starts_with
            end_withs
    */
    // Cpp 20
    string str = "emre_bahtiyar.bin";
    if (str.start_with("emre")) // return bool
    {
    }

    if (str.end_withs("bin"))
    {

    }

    // Cpp 23 bu substring içeriyor mu?
    if (str.contains("bahtiyar")) // return bool
    {

    }

    str.rfind('a'); //aramayı sondan yapıyor
    str.find_first_of("ptik"); // bu karakterlerden birini bulur ilk geleni
    str.find_first_not_of("resim"); //bu karaterlerde olmayan ilkini bulur
    str.find_last_of("resim"); // bu karakterlerden sondan olanı
    str.find_last_not_of("resim"); // bu karakterlerden olmayan sondan
}
```

String Yazıyı Değiştiren Fonksiyonlar

```cpp
int main()
{
    using namespace std;

    string s1  = "omer faruk";
    string s2 =  "selma deniz";
    // append
    s1.append(s2, 5); // 5. indeksten başlayarak
    s1.append(s2, 1, string::npos); // 1. indeksten geriye kalanları almak için

    /*
        STL Container interfacer gelen
        iterator interface sahiplerdir:
        insert ve erase fonksiyonları

        con.insert(iter, value)

    */

    string s{"mehmet bal"};

    // iterator interface ile yazının başına ! karakteri ekleyin.

    s.insert(s.begin(), '!');
    cout << s;

    while (!s.empty())
    {
        cout << s;
        s.erase(s.begin());
        // son silme
        s.erase(s.pushback());
        s.erase(s.end());
    }

    s.erase(s.begin(), s.begin() + 3 ); // ilk 3 karakter silincek
    s.erase(s.begin()  + 1, s.end() - 1);


    string s1{"emre"};
    string name {"bahtiyar"};

    s.insert(4, name, 2, 3); // 4.indekse name'ın 2. indeksinden 3. indeksine
kadar


}
```

String Sınıfı Karşılaştırma İşlemleri

```
int main()
{
    string name{"ayhan"};
    string word{"ekrem"};

    if (name == word)
    if (name < word)
    if (name > word)

}
```

Shrink

```
int main()
{
    string str(100'000, 'A');

    cout << "str.size()" << str.size() << "\n";
    cout << "str.capacity()" << str.capacity() << "\n";

    str.erase(10) // ilk 10 dışındakileri sil

    cout << "str.size()" << str.size() << "\n"; // size 10
    cout << "str.capacity()" << str.capacity() << "\n"; // capacity değişmedi

    str.shrink_to_fit(); // capacity size uygun bi değere getirir

}
```