Algoritmalar ve Programlama II – BLM 102

Hafta 10: Şablonlar (Templates)

Fenerbahçe Üniversitesi

Öğretim Elemanları

Öğretim Üyesi: Dr. Vecdi Emre Levent

Ofis: 311

Email: emre.levent@fbu.edu.tr

Asistan: Arş. Gör. Uğur Özbalkan

Ofis: 311

Email: ugur.ozbalkan@fbu.edu.tr

Ders Planı

- Şablonlar
- Fonksiyon Şablonları
- Sınıf Şablonları

Şablonlar (Templates)

- Şablonlar, bir kodu parametrik ve kolay değişiklik yapılabilmesini sağlamak için kullanılan bir yaklaşımdır.
- Programdaki bir fonksiyonun veritipinden bağımsız olmasını sağlar.

Şablonlar (Templates)

Kullanımı

template <typename xyz>
funcDonusTipi funcAdi(Argumanlar …) {
        // Fonksiyon görevleri
}

```
template <typename T>
T Max(T a, T b) {
        return a < b ? b : a;
}
```

Şablonlar (Templates)

```cpp
#include <iostream>
#include <string>

using namespace std;

template <typename T>
T Max(T a, T b) {
return a < b ? b : a;
}
```

```cpp
int main() {
int i = 39;
int j = 20;
cout << "Max(i, j): " << Max(i, j) << endl;

double f1 = 13.5;
double f2 = 20.7;
cout << "Max(f1, f2): " << Max(f1, f2) << endl;

string s1 = "Hello";
string s2 = "World";
cout << "Max(s1, s2): " << Max(s1, s2) << endl;

return 0;
}
```

Çıktı:
Max(i, j): 39
Max(f1, f2): 20.7
Max(s1, s2): World

Şablonlar (Templates)

- Derleyici, derleme zamanında tüm fonksiyonlar için kopyalar oluşturur ve argüman türüne göre spesifik operasyonlar yerleştirir.

- Arguman türüne göre kopya fonksiyonlardan uygun olan çağrılır.

Şablonlar (Templates)

```
#include <iostream>
#include <string>

using namespace std;

int Max(int a, int b) {
return a < b ? b : a;
}

double Max(double a, double b) {
return a < b ? b : a;
}

string Max(string a, string b) {
return a < b ? b : a;
}
```

```
int main() {
int i = 39;
int j = 20;
cout << "Max(i, j): " << Max(i, j) << endl;

double f1 = 13.5;
double f2 = 20.7;
cout << "Max(f1, f2): " << Max(f1, f2) << endl;

string s1 = "Hello";
string s2 = "World";
cout << "Max(s1, s2): " << Max(s1, s2) << endl;

return 0;
}
```

Çıktı:
Max(i, j): 39
Max(f1, f2): 20.7
Max(s1, s2): World

Şablonlar (Templates)

- Şablonlar; fonksiyonlarda kullanıldığı gibi, aynı şekilde sınıflarda da kullanılabilmektedir.

Şablonlar (Templates)

Kullanımı

```
template <class xyz>
class sinifAdi {
        // Sınıf Tanımlamaları
}
```

Şablonlar (Templates)

```
#include <iostream>
using namespace std;

template <class T>
class testSinifi {
T a, b;
public:
testSinifi(T arg1, T arg2)
{
a = arg1; b = arg2;
}
T maxBul();
};

template <class T>
T testSinifi<T>::maxBul()
{
T retval;
retval = a > b ? a : b;
return retval;
}
```

```
int main() {
testSinifi <int> obj1(100, 75);
cout << obj1.maxBul() << "\n";

testSinifi <float> obj2(3.5, 0.75);
cout << obj2.maxBul();

return 0;
}
```

Çıktı:

```
100
3.5
```

Şablonlar (Templates)

- Şablonlarda birden çok veri türü kullanılabilir.

Şablonlar (Templates)

```
#include<iostream>
using namespace std;

template<class T, class U>
class A {
T x;
U y;
public:
A(T arg1, U arg2) {
x = arg1;
y = arg2;
}
U islemYap() {
return (U)x + y;
}
void bastir() {
cout << islemYap() << "\n";
}
};
```

```
int main() {
A<int, int> obj1(3, 5);
A<int, double> obj2(3, 12.5);
A<int, char> obj3('a', 2);

obj1.bastir();
obj2.bastir();
obj3.bastir();

return 0;
}
```

Çıktı:

```
8
15.5
c
```

Şablonlar (Templates)

- Şablonlar kullanılmış bir sınıftan obje türetilirken, objede veri türü belirtilmemesi durumuna karşın, default olarak veri türü ayarlanabilmektedir.

Şablonlar (Templates)

```
#include<iostream>
using namespace std;

template<class T, class U = char>
class A {
T x;
U y;
public:
A(T arg1, U arg2) {
x = arg1;
y = arg2;
}
U islemYap() {
return (U)x + y;
}
void bastir() {
cout << islemYap() << "\n";
}
};
```

```
int main() {
A<int> obj1(5, 65);
A<int, double> obj2(3, 12.5);
A<int, char> obj3('a', 2);

obj1.bastir();
obj2.bastir();
obj3.bastir();

return 0;
}
```

ASCII Karşılığı ‘A’

Çıktı:

```
F
15.5
c
```

Şablonlar (Templates)

- Statik değişkenler, şablon fonksiyonları ve sınıflarında kullanılabilirler.
- Farklı türler kullanıldığında fonksiyonların kopyaları yaratıldığı için, static tanımlar her değişken türü için farklı olurlar.
- Ancak bir statik değişken şablon tanımı türünde tanımlanamaz.
- Statik değişkenler tüm sınıflar için derleme zamanında ortak açıldığı için, derleme zamanında tek bir değişken türüne sahip olabilir.

Şablonlar (Templates)

```
#include <iostream>

using namespace std;

template <typename T>
void fun(const T& x)
{
static int i = 10;
i = i + x;
cout << i;
return;
}
```

```
int main()
{
fun<int>(1);
cout << endl;
fun<int>(2);
cout << endl;
fun<double>(20.1);
cout << endl;
fun<int>(-3);
cout << endl;
return 0;
}
```

Çıktı:

```
11
13
30
10
```

Şablonlar (Templates)

- Kod parçacığında 1 integer, 1 double için farklı fonksiyonlar yaratılmıştır.
- Dolayısıyla integer ve double fonksiyonlarının ayrı ayrı static değişkenleri vardır.

Şablonlar (Templates)

```
#include <iostream>

using namespace std;

template <class T> class Test
{
public:
static int count;
Test(T val)
{
count++;
}
};

template<class T>
int Test<T>::count = 0;
```

```
int main()
{
Test<int> a(3);
Test<int> b(5);
Test<double> c(-1.5);

cout << Test<int>::count << endl;
cout << Test<double>::count << endl;

getchar();
return 0;
}
```

Çıktı:

```
2
1
```

Şablonlar (Templates)

- Kod parçacığında 1 integer, 1 double için farklı sınıflar yaratılmıştır.
- Dolayısıyla integer ve double fonksiyonlarının ayrı ayrı static değişkenleri vardır.

Şablonlar (Templates)

- Şablonlara sabit değişkenlerde argüman olarak gönderilebilmektedir.
- Bu yaklaşım genel olarak sabit bir değişkenin kullanılma ihtiyacı olduğu durumlarda tercih edilir.

Şablonlar (Templates)

```
#include <iostream>
using namespace std;

template <class T, int max>
int arrMin(T arr[], int n)
{
int m = max;
for (int i = 0; i < n; i++)
if (arr[i] < m)
m = arr[i];

return m;
}
```

```
int main()
{
int arr1[] = { 10, 20, 15, 12 };
int n1 = sizeof(arr1) / sizeof(arr1[0]);

char arr2[] = { 1, 2, 3 };
int n2 = sizeof(arr2) / sizeof(arr2[0]);

cout << arrMin<int, 10000>(arr1, n1) << endl;
cout << arrMin<char, 256>(arr2, n2);

return 0;
}
```

Çıktı:
10
1