

14. Şablonlar (Templates)

Şablonlar yazılımları tekrar kullanabilmenin bir yöntemidir. Bu sayede tek bir kod yazarak farklı tiplerde değer alan fonksiyonlar sınıflar tanımlayabiliriz. Gövdeleri aynı olan fonksiyonları şablon olarak tanımlayabiliriz. Şablon kullanmanın iki önemli avantajı vardır. İki avantajı vardır.

1. Program kodunu kısaltır. Birden fazla fonksiyon yazacağımıza tek bir fonksiyon yazarız.
2. Hata düzeltmeyi kolaylaştırır. Eğer kod yazarken hata yapılmışsa tek bir yede düzeltmek yeterli olacaktır.

Aşağıda verilen programda GetMax fonksiyonu aşırı yüklenmiştir. Bu fonksiyon ilkinde int türünden sayıları parametre olarak almaktra ve büyüğünü geri döndürmektedir. İkincisinde ise float türünden sayıları parametre olarak almaktadır. Kod kısmı ise aynıdır.

```
#include <iostream>
#include <conio.h>

using namespace std;

int GetMax (int a, int b) {
    int result;
    result = (a>b)? a : b;
    return (result);
}

float GetMax (float a, float b) {
    float result;
    result = (a>b)? a*3 : b+10;
    return (result);
}

int main ()
{
    int i=5, j=6, k;
    float l=10.6, m=5.1, n;
    k=GetMax(i,j);
    n=GetMax(l,m);
    cout << k << endl;
    cout << n << endl;

    getch();
    return 0;
}
```

Yukarda verilen iki fonksiyon yerine tek bir fonksiyon yazılabilir. Bunun için Fonksiyon şablon olarak tanımlanmalıdır. Aşağıda bunun örneği görülmektedir.

```
template <class T>
```

```
T GetMax (T a, T b) {
```

Yukarda verilen satırda a ve b int, float veya başka bir türden olabilir. Ne verilirse ona göre program derlenerek çalıştırılacaktır. Örneğin aşağıdaki satırda i ve j değişkenleri <int> ifadesi ile tamsayı olacağı belirtilmiştir.

```
k=GetMax<int>(i,j);
```

Eğer i ve j değişkenleri int i,j; şeklinde önceden int türünden oldukları belirtilmişse <int> ifadesini koymaya gerek yoktur. Bu taktirde

```
int i,j;  
k=GetMax(i,j);
```

Şeklinde de çağrılabilir.

Aşağıda verilen örnekleri inceleyiniz.

```
//-----
```

```
#include <iostream>
```

```
#include <conio.h>
```

```
using namespace std;
```

```
template <class T>
```

```
T GetMax (T a, T b) {
```

```
    T result;
```

```
    result = (a>b)? a : b;
```

```
    return (result);
```

```
}
```

```
int main ()
```

```
{
```

```
    int i=5, j=6, k;
```

```
    float l=10.6, m=5.7, n;
```

```
    k=GetMax<int>(i,j);
```

```
    n=GetMax<float>(l,m);
```

```
    cout << k << endl;
```

```
    cout << n << endl;
```

```
    getch();
```

```
    return 0;
```

```
}
```

```
//-----
#include <iostream>
#include <conio.h>

using namespace std;
template <class T, class U>

T Alan (T a, U b) {
    T result;
    result = a*b;
    return result;
}

int main ()
{

    float y=9.5, z;
    int x=4,k;

    z=Alan(y,x);
    k=Alan(x,y);
    cout << z << endl;
    cout << k << endl;

    getch();
    return 0;
}
```

```
//-----
#include <iostream>
#include <conio.h>

using namespace std;
template <class T>

T GetMax (T a, T b) {
    T result;
    result = (a>b)? a : b;
    return (result);
}

int main ()
{
    int i=5, j=6, k;
    float l=10.6, m=5.7, n;
    k=GetMax(i,j);
    n=GetMax(l,m);
    cout << k << endl;
    cout << n << endl;

    getch();
    return 0;
}
```

Sınıflarda template kullanımı. Aşağıda verilen örneği inceleyiniz.

```
//-----
#include <iostream>
#include <conio.h>
using namespace std;

template <class T>

class Dortgen {
    T x, y;
public:
    void DegerVer(T,T);
    T Alan (void);
};

template <class T>
void Dortgen<T>::DegerVer(T a, T b) {
    x = a;
    y = b;
}

template <class T>
T Dortgen<T>::Alan (void) {
    T deger;
    deger=x*y;
    return deger;
}

int main ()
{
    Dortgen <int>D1 ;
    Dortgen <float>D2;

    int a=3,b=4;

    D1.DegerVer(3,4);
    D1.DegerVer(a,b);

    D2.DegerVer(7.4,2.5);
    cout << "Alan1: " << D1.Alan()<<'\n';
    cout << "Alan2: " << D2.Alan()<<endl;

    getch();
    return 0;
}
```