

ÇOK DEĞİŞKENLİ FONKSİYONLAR

Değişken Sayıda Argüman Alabilen Fonksiyonlar

Değişken sayıda argüman alabilen bir fonksiyona **çok değişkenli fonksiyon** (**variadic function**) denir. C dilindeki güçlü ancak çok nadiren kullanılan özelliklerden biridir. C++ dili de bu fonksiyonları özellik olarak devralmıştır.

Değişken sayıda bağımsız değişkeni olan bir fonksiyon; en az bir sabit bağımsız değişkene sahip olacak şekilde tanımlanır ve ardından derleyicinin değişken sayıda bağımsız değişkeni ayrıştırmasını sağlayan bir **üç nokta simgesi** (**ellipsis**) eklenir.

```
dönüş-tipi fonksiyonkimliği(veri-tipi birinciargüman, ...);
```

Değişken argümanları işlemek için kodunuza **stdarg.h** başlık dosyasını kodunuza dahil etmeniz gerekir;

| Fonksiyon | Açıklama |
|---|--|
| va_start(va_list ap, arg) | Bu fonksiyon, üç nokta ile verilen argümanları va_list değişkenine aktarır. |
| va_arg(va_list ap, type) | Her seferinde, üç nokta ile temsil edilen değişken listesindeki bir sonraki argümanı va_list üzerinden işler ve listenin sonuna ulaşana kadar onu type ile verilen veri tipine dönüştürür. |
| va_copy(va_list dest, va_list src) | va_list 'teki argümanların bir kopyasını oluşturur. |
| va_end(va_list ap) | Bu, va_list değişkenlerine erişimi sonlandırır. |

Tablo 25. CStdarg Fonksiyonları

Aşağıda ilk parametre ile belirlenmiş argüman sayısı kadar argüman alan iki fonksiyon tanımlanmıştır.

```
#include <iostream>
#include <stdarg.h>
using namespace std;

void sayilariYaz(int count, ...) {
    va_list args;
    va_start(args, count);

    for (int i = 0; i < count; ++i) {
        cout << va_arg(args, int) << " ";
    }
    cout << endl;
    va_end(args);
}

int argumanlarinHepsiniTopla(int kacAdet, ...) {
    va_list argumanlar;
    int sayac, toplam = 0;
    va_start(argumanlar, kacAdet);
    for (sayac = 0; sayac < kacAdet; sayac++)
        toplam += va_arg(argumanlar, int);
    va_end(argumanlar);
    return toplam;
}

int main(){
    sayilariYaz(4, 1.0, 2.0, 30, 40);
    cout << "3 Argüman Toplamı =" << argumanlarinHepsiniTopla(3, 10, 20, 30)
        << endl;
    cout << "5 Argüman Toplamı =" <<
        argumanlarinHepsiniTopla(5, 10, 20, 30, 40, 50) << endl;
}
```

```

/*Program Çıktısı:
30 40 2140265872 0
3 Argüman Toplamı =60
5 Argüman Toplamı =150

...Program finished with exit code 0
*/

```

Ana Fonksiyonun Parametreleri

C++ dilinde **ana fonksiyon** (**main function**) programın icra edilmeye başladığı fonksiyondur. Ana fonksiyon;

- Aşırı yüklenemez (**overload**)!
- Satır içi (**inline**) fonksiyon olarak bildirilemez!
- Statik (**static**) fonksiyon olarak bildirilemez!
- Adresi alınamaz!
- Programın başka hiçbir yerinden çağrılmaz (**call**)!

Yazdığımız programlar da çalıştırılırken konsoldan argüman alabilir. Şu ana kadar argümansız ana fonksiyonu gördük. Argüman alan **ana fonksiyon** (**main function**) aşağıdaki iki şekilde gibi tanımlanır;

```

int main(int argc,
         char* argv[]) {
    // Ana fonksiyon Gövdesi
}
//YADA
int main(int argc,
         char* argv[],
         char ** envp) {
    // Ana fonksiyon Gövdesi
}

```

Aşağıda konsoldan çalıştırılırken alınan argümanları gösteren bir program örneği verilmiştir;

```

#include <iostream>
using namespace std;
int main( int argc, char *argv[] )
{
    for ( int i = 0; argv[i] != NULL; ++i )
        cout << i << ": " << argv[i] << "\n";
}
/* "C:\Users\ILHANOZKAN>main.exe 10 20 -v" şeklinde çalıştırıldığında:
0: main.exe
1: 10
2: 20
3: -v
*/

```

Aşağıda konsoldan çalıştırılırken girilen parametreler ile işletim sisteminin **ortam değişkenleri** (**environment variable**) konsola yazan bir program örneği verilmiştir;

```

#include <iostream>
using namespace std;
int main( int argc, char *argv[], char *envp[] )
{
    for ( int i = 0; argv[i] != NULL; ++i )
        cout << i << ": " << argv[i] << "\n";

    for ( int i = 0; envp[i] != NULL; ++i )
        cout << i << ": " << envp[i] << "\n";
}

```

```

/* "C:\Users\ILHANOZKAN>main.exe 10 20 -v" şeklinde çalıştırıldığında:
0: main.exe
1: 10
2: 20
3: -v
0: ALLUSERSPROFILE=C:\ProgramData
1: APPDATA=C:\Users\ILHANOZKAN\AppData\Roaming
2: CommonProgramFiles=C:\Program Files\Common Files
3: CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
4: CommonProgramW6432=C:\Program Files\Common Files
5: COMPUTERNAME=ILHANOZKAN
6: ComSpec=C:\Windows\system32\cmd.exe
7: DriverData=C:\Windows\System32\Drivers\DriverData
8: EFC_25916=0
9: FPS_BROWSER_APP_PROFILE_STRING=Internet Explorer
10: FPS_BROWSER_USER_PROFILE_STRING=Default
11: HOMEDRIVE=C:
12: HOMETHINK=C:\Users\ILHANOZKAN
13: LOCALAPPDATA=C:\Users\ILHANOZKAN\AppData\Local
14: LOGONSERVER=\\ILHANOZKAN
15: NUMBER_OF_PROCESSORS=20
16: OneDrive=C:\Users\ILHANOZKAN\OneDrive
17: OS=Windows_NT
18: Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program
Files\dotnet\;C:\Users\ILHANOZKAN\AppData\Local\Microsoft\WindowsApps;D:\msys64\ucrt64\bin;C:\
Users\ILHANOZKAN\dotnet\tools
19: PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
20: PROCESSOR_ARCHITECTURE=AMD64
21: PROCESSOR_IDENTIFIER=Intel64 Family 6 Model 186 Stepping 2, GenuineIntel
22: PROCESSOR_LEVEL=6
23: PROCESSOR_REVISION=ba02
24: ProgramData=C:\ProgramData
25: ProgramFiles=C:\Program Files
26: ProgramFiles(x86)=C:\Program Files (x86)
27: ProgramW6432=C:\Program Files
28: PROMPT=$P$G
29: PSMODULEPATH=C:\Program
Files\WindowsPowerShell\Modules;C:\Windows\system32\WindowsPowerShell\v1.0\Modules
30: PUBLIC=C:\Users\Public
31: SESSIONNAME=Console
32: SystemDrive=C:
33: SystemRoot=C:\Windows
34: TEMP=C:\Users\ILHANO~1\AppData\Local\Temp
35: TMP=C:\Users\ILHANO~1\AppData\Local\Temp
36: UATDATA=C:\Windows\CCM\UATData\D9F8C395-CAB8-491d-B8AC-179A1FE1BE77
37: USERDOMAIN=ILHANOZKAN
38: USERDOMAIN_ROAMINGPROFILE=ILHANOZKAN
39: USERNAME=ILHANOZKAN
40: USERPROFILE=C:\Users\ILHANOZKAN
41: windir=C:\Windows
42: ZES_ENABLE_SYSMAN=1
*/

```