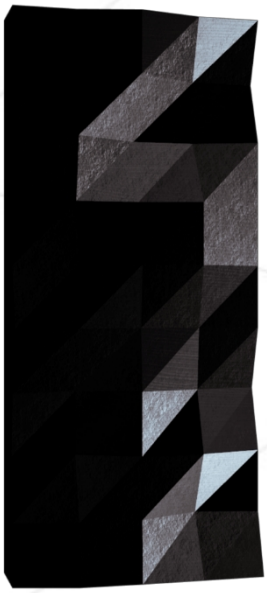

C++ - Modül 07

C++ şablonları

Özet:
Bu doküman C++ modüllerinden Modül 07'nin alıştırmalarını içermektedir.

Sürüm: 6

İçindekiler

Bölüm I Giriş

C++, Bjarne Stroustrup tarafından C programlama dilinin ya da "C with Classes" (Sınıflı C) dilinin bir uzantısı olarak yaratılmış genel amaçlı bir programlama dilidir (kaynak: Wikipedia).

Bu modüllerin amacı sizi **Nesne Yönelimli Programlama** ile tanıştırmaktır. Bu, C++ yolculuğunuzun başlangıç noktası olacaktır. OOP öğrenmek için birçok dil önerilmektedir. Eski dostunuz C'den türetildiği için C++'ı seçmeye karar verdik. Bu karmaşık bir dil olduğundan ve işleri basit tutmak için kodunuz C++98 standardına uygun olacaktır.

Modern C++'ın pek çok açıdan çok farklı olduğunun farkındayız. Dolayısıyla, yetkin bir C++ geliştiricisi olmak istiyorsanız, 42 Common Core'dan sonra daha ileri gitmek size kalmış!

Bölüm II Genel kurallar

Derleme

- Kodunuzu c++ ve -Wall -Wextra -Werror bayrakları ile derleyin
- Eğer -std=c++98 bayrağını eklerseniz kodunuz yine de derlenmelidir

Biçimlendirme ve adlandırma kuralları

- Alıştırma dizinleri şu şekilde adlandırılacaktır: ex00, ex01, ... , exn
- Dosyalarınızı, sınıflarınızı, işlevlerinizi, üye işlevlerinizi ve niteliklerinizi yönergelerde belirtildiği şekilde adlandırın.
- Sınıf adlarını **UpperCamelCase** biçiminde yazın. Sınıf kodu içeren dosyalar her zaman sınıf adına göre adlandırılacaktır. Örneğin: ClassName.hpp/ClassName.h, ClassName.cpp veya ClassName.tpp. Bu durumda, tuğla duvar anlamına gelen "BrickWall" sınıfının tanımını içeren bir başlık dosyanız varsa, adı BrickWall.hpp olacaktır.
- Aksi belirtilmedikçe, her çıktı mesajı bir yeni satır karakteriyle sonlandırılmalı ve standart çıktıda görüntülenmelidir.
- *Güle güle Norminette!* C++ modüllerinde herhangi bir kodlama stili zorunlu değildir. En sevdiğinizi takip edebilirsiniz. Ancak, akran değerlendiricilerinizin anlayamadığı bir kodun not veremeyecekleri bir kod olduğunu unutmayın. Temiz ve okunabilir bir kod yazmak için elinizden geleni yapın.

İzinli/Yasak

Artık C'de kodlama yapmıyorsunuz. C++ zamanı! Bu nedenle:

- Standart kütüphanedeki neredeyse her şeyi kullanmanıza izin verilir. Bu nedenle, zaten bildiklerinize bağlı kalmak yerine, alışkın olduğunuz C işlevlerinin C++-ish sürümlerini mümkün olduğunca kullanmak akıllıca olacaktır.
- Ancak, başka herhangi bir harici kütüphane kullanamazsınız. Bu, C++11 (ve türetilmiş formlar) ve Boost kütüphanelerinin yasak olduğu anlamına gelir. Aşağıdaki fonksiyonlar da yasaktır: *printf(), *alloc() ve free(). Eğer bunları kullanırsanız, notunuz 0 olacaktır ve hepsi bu kadar.

- Aksi açıkça belirtilmedikçe, using namespace <ns_name> ve arkadaş anahtar kelimeleri yasaktır. Aksi takdirde notunuz -42 olacaktır.
- **STL'yi sadece Modül 08 ve 09'da kullanmanıza izin verilmektedir.** Bunun anlamı: o zamana kadar **Kapsayıcı** (vektör/liste/harita/ve benzeri) ve **Algoritma** (<algorithm> başlığını içermesi gereken herhangi bir şey) kullanmayacaksınız. Aksi takdirde notunuz -42 olacaktır.

Birkaç tasarım gereksinimi

- C++'da da bellek sızıntısı meydana gelir. Bellek ayırdığınızda (new anahtar sözcüğü), **bellek sızıntılarından** kaçınmalısınız.
- Modül 02'den Modül 09'a kadar, **aksi açıkça belirtilmediği sürece**, dersleriniz **Ortodoks Kanonik Formunda** tasarlanmalıdır.
- Bir başlık dosyasına konulan herhangi bir işlev uygulaması (işlev şablonları hariç) alıştırma için 0 anlamına gelir.
- Başlıklarınızın her birini diğerlerinden bağımsız olarak kullanabilmelisiniz. Bu nedenle, ihtiyaç duydukları tüm bağımlılıkları içermelidirler. Ancak, **include korumaları** ekleyerek çifte içerme sorunundan kaçınmalısınız. Aksi takdirde notunuz 0 olacaktır.

Beni okuyun

- Gerekirse bazı ek dosyalar ekleyebilirsiniz (örneğin, kodunuzu bölmek için). Bu ödevler bir program tarafından doğrulanmadığından, zorunlu dosyaları teslim ettiğiniz sürece bunu yapmaktan çekinmeyin.
- Bazen bir alıştırmanın yönergeleri kısa görünebilir ancak örnekler, yönergelerde açıkça yazılmayan gereksinimleri gösterebilir.
- Başlamadan önce her modülü tamamen okuyun! Gerçekten, okuyun.
- Odin tarafından, Thor tarafından! Beynini kullan!!!

Çok sayıda sınıf uygulamanız gerekecek. En sevdiğiniz metin düzenleyicinizi kodlayamadığınız sürece bu sıkıcı görünebilir.

Egzersizleri tamamlamanız için size belirli bir miktar özgürlük verilir. Ancak, zorunlu kurallara uyun ve tembellik etmeyin. Pek çok faydalı bilgiyi kaçırırsınız! Teorik kavramlar hakkında okumaktan çekinmeyin.

Bölüm III

Alıştırma 00: Birkaç fonksiyonla başlayın

	Egzersiz : 00
Birkaç işlevle başlayın	
Giriş dizini : *ex00/*	
Teslim edilecek dosyalar : `Makefile, main.cpp, whatever.{h, hpp}`	
Yasak fonksiyonlar : `Hiçbiri`	

Aşağıdaki işlev şablonlarını uygulayın:

- swap: Verilen iki bağımsız değişkenin değerlerini değiştirir. Hiçbir şey döndürmez.
- min: Argümanlarında geçirilen iki değeri karşılaştırır ve en küçük olanı döndürür. İkisi eşitse, `ikincisini` döndürür.
- max: Bağımsız değişkenlerinde aktarılan iki değeri karşılaştırır ve en büyük olanı döndürür. İkisi eşitse, `ikincisini` döndürür.

Bu fonksiyonlar herhangi bir argüman türü ile çağrılabilir. Tek şart, iki argümanın aynı tipte olması ve tüm karşılaştırma operatörlerini desteklemesidir.

`Şablonlar başlık dosyalarında tanımlanmalıdır.`

Aşağıdaki kodu çalıştırıyorum:

```
int     main( void ) {

    int a = 2;
    int b = 3;

    ::swap( a, b );
    std::cout << "a = " << a << ", b = " << b << std::endl;
    std::cout << "min( a, b ) = " << ::min( a, b ) << std::endl;
    std::cout << "max( a, b ) = " << ::max( a, b ) << std::endl;

    std::string c = "chaine1";
    std::string d = "chaine2";

    ::swap(c, d);
    std::cout << "c = " << c << ", d = " << d << std::endl;
    std::cout << "min( c, d ) = " << ::min( c, d ) << std::endl;
    std::cout << "max( c, d ) = " << ::max( c, d ) << std::endl;

    0 döndür;
}
```

Çıkmalı:

```
a = 3, b = 2
min(a, b) = 2
maks(a, b) = 3
c = chaine2, d = chaine1
min(c, d) = chaine1
max(c, d) = chaine2
```

Bölüm IV

Alıştırma 01:

Yineleme

	Egzersiz : 01
Iter	
Giriş dizini : *ex01/*	
Teslim edilecek dosyalar : `Makefile, main.cpp,`	
`iter.{h, hpp}`	

Yasak fonksiyonlar : `Hiçbiri`

Üç parametre alan ve hiçbir şey döndürmeyen bir iter işlev şablonu uygulayın.

- İlk parametre bir dizinin adresidir.
- İkincisi ise dizinin uzunluğudur.
- Üçüncüsü, dizinin her elemanı üzerinde çağrılacak bir fonksiyondur.

Testlerinizi içeren bir main.cpp dosyası teslim edin. Bir test çalıştırılabilir dosyası oluşturmak için yeterli kod sağlayın.

iter işlev şablonunuz herhangi bir dizi türüyle çalışmalıdır. Üçüncü parametre, örneklenmiş bir işlev şablonu olabilir.

Bölüm V Alıştırma

02: Dizi

	Alıştırma : 02
Dizi	
Giriş dizini : *ex02/*	
Teslim edilecek dosyalar : `Makefile, main.cpp,` `Array.{h, hpp} ve isteğe bağlı dosya: Array.tpp`	
Yasak fonksiyonlar : `Hiçbiri`	

T türünde öğeler içeren ve aşağıdaki davranış ve işlevleri uygulayan **bir Array** sınıf şablonu geliştirin:

- Parametresiz yapı: Boş bir dizi oluşturur.
- Parametre olarak unsigned int n ile oluşturulur: Varsayılan olarak ilklendirilmiş n elemanlı bir dizi oluşturur.
  İpucu: `int * a = new int();` öğesini derlemeyi deneyin ve ardından *a öğesini görüntüleyin.
- Kopyalama ve atama operatörü ile oluşturma. Her iki durumda da, kopyalamadan sonra orijinal diziyi veya kopyasını değiştirmek diğer diziyi etkilememelidir.
- Bellek ayırmak için new[] operatörünü kullanmalısınız. Önleyici tahsis (belleğin yerini önceden belirleme) yasaktır. Programınız tahsis edilmemiş belleğe asla erişmemelidir.
- Elemanlara alt simge operatörü aracılığıyla erişilebilir: [ ].
- Bir elemana [ ] işleci ile erişirken, indeksi sınırların dışındaysa std::exception atılır.
- Dizideki eleman sayısını döndüren size() üye işlevi. Bu üye fonksiyon parametre almaz ve geçerli örneği değiştirmemelidir.

Her zamanki gibi, her şeyin beklendiği gibi çalıştığından emin olun ve testlerinizi içeren bir main.cpp dosyası teslim edin.