

Dersin Adı : BLM 211 Nesneye Dayalı Programlama

Bu dersi alan öğrenci Nesneye dayalı programlama kavramını ve altında yatan düşünce mantığını anlayacak, C++ dilini kullanarak Sınıf tanımlarını yapabilecek, Kendi oluşturduğu Sınıfları, Mevcut Sınıfları ve Şablonları tekrar kullanarak Nesneye Yönelik Programlar yazabilecektir.

Konular

Nesneye Dayalı Programlamaya giriş
C++ programlama dilinin Temelleri, Döngü ve Kararlar, Fonksiyonlar
Sınıf (Class) yapısı, Kullanımı ve Özellikleri
Kurucu (constructor) ve Yokedicici fonksiyonlar (destructor)
Nesnelerin fonksiyonlara parametre olarak gönderilmesi
Operatörlere yeni işlevlerin yüklenmesi (operator overloading)
Kalıtım (Inheritance)
Kalıtımın kurucu ve yok ediciler üzerindeki etkisi
Çoklu kalıtım (multiple inheritance)
Nesne işaretçileri (pointers to objects)
Çok Şekillilik (Polymorphism)
Sıra Dışı Durumlar (Exceptions)
Şablonlar (Templates)

Kaynaklar:

1. Object Oriented Programming in C++, Robert Lafore, Fourth Edition, Sams Publishing, 2002, (Kitabın türkçe çevirisi bulunmakta, Nesne Yönelimli C++ Programlama Kılavuzu, Robert Lafore, Alfa yayınları)
2. C++ How to Program, Seventh edition, Harvey M. Deitel and Paul J. Deitel, Deitel & Associates, Inc. 2010.(Kitabın türkçe çevirisi bulunmakta, C ve C++, Deitel & Deitel, Sistem yayıncılık)
3. Thinking in C++ (Free Electronic Book), Bruce Eckel, 2nd ed., Vol. 1 and Vol.2, Prentice-Hall, 2000 and 2003.
4. C++ ve Nesneye Yönelik Programlama, M. Erhan SARIDOĞAN, Papatya yayıncılık, 2004
5. C++ ile Nesne Tabanlı programlama, Bora GÜNGÖREN, Seçkin yayıncılık, 2001

İnternet Kaynakları

6. <http://www.buzluca.com/cpp/notlar.html>
7. <http://web.itu.edu.tr/~mscelebi/cpp/index.html>
8. <http://web.itu.edu.tr/~uyar/programlama/c.pdf>
9. <http://www.cplusplus.com/doc/tutorial/introduction/>
10. http://sundog97.tripod.com/tutorial_cpp/index.html

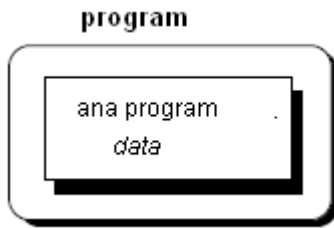
Giriş

Programlama, Yaşadığımız gerçek dünyadaki problemlerin çözümlerinin bilgisayarda ifade edilmesidir. Programcılar problemlerin çözümleri ile ilgili kavramları, varlıkları kullandıkları programlama dili ile ifade ederler[1]. Programlama teknikleri şu şekilde sıralanabilir [7].

Yapısal olmayan programlama
Yapısal programlama
Nesne Yönelimli programlama

Yapısal olmayan programlama

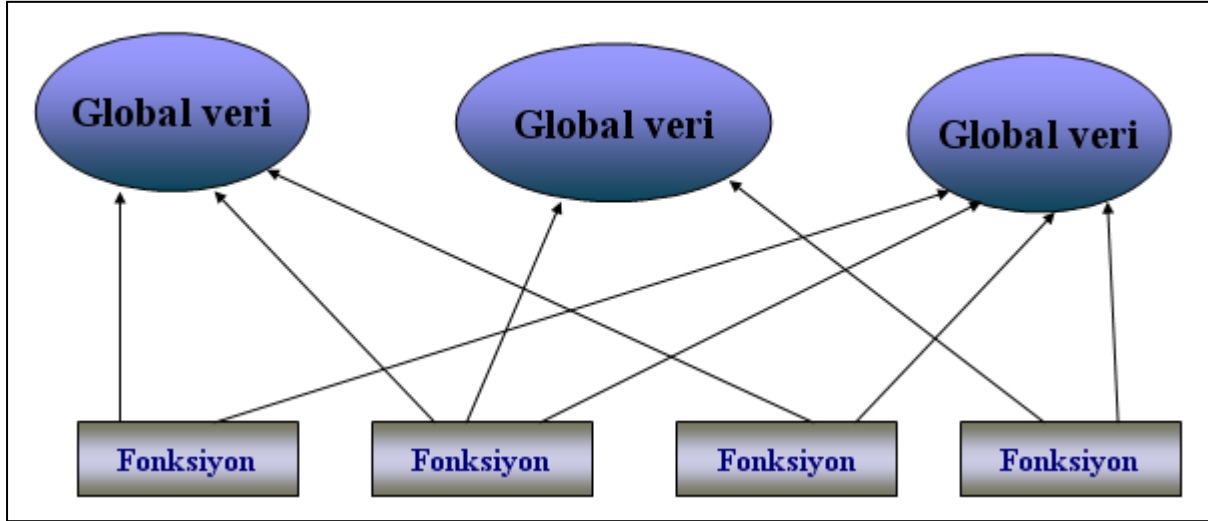
Bu yöntemde program bir tek ana program yer alır. Ana program ard arda gelen komut dizilerinden oluşur. Altprogramlar ve bloklama yoktur. Ana program içinde kullanılan komutlar global olarak tanımlanmış değişkenlere erişerek kullanılır. Programlar büyüdükçe tanımlanan değişkenlerin bozulması mümkün olabilmektedir. Ayrıca program için kullanılan ve ard arda gelen komut dizilerini tekrar kullanmak gerektiğinde bu komut dizilerini tekrar uygun yere kopyalayıp kullanmak gerekmektedir. Bu ise program boyunu gereksiz yere uzatmaktadır. Programın okunabilirliğini, anlaşılabilirliğini azaltmakta ve hataları bulmayı zorlaştırmaktadır.



Şekil 1. Yapısal olmayan programlama

Yapısal (structural) programlama

Procedural (işleve dayalı, altprogram kullanmaya dayalı) programlama olarak da ifade edilebilir. Bu programlama yönteminde bloklar halinde program yazma ve altprogramların kullanımı ön plana çıkmaktadır. Altprogram(procedure, subroutine) ve Fonksiyonlar(Function) küçük farklarla benzer kavramları ifade etmektedirler. Her ikisi de belirli bir amaca yönelik işlemleri yerine getiren program parçacıklarıdır. İhtiyac duyulan her noktada ana program akışı içinde herhangi bir noktadan isimleri yazılarak defalarca çağrılıp kullanılabilirler. Ana program sıralı bir şekilde değişik altprogramların çağrılıp kullanılması ile oluşturulur. Ana programdan altprograma veri işlenmek üzere gönderilir. Altprogram bunu işleyerek anaprograma tekrar geri dönderir. Altprogramlar bir kez hatasız bir şekilde yazıldıktan sonra defalarca kullanılabilir. Bir hata oluştuğunda hatanın kaynağını bulmak daha kolaydır. Programın anlaşılabilirliği, okunabilirliği daha yüksektir. Ancak Şekil 2' de görüldüğü gibi global verilere erişimde bir sınırlama getirilmemiştir. Veriler ve bu veriler üzerinde işlem yapan fonksiyonlar ayrı ayrı düşünülmüştür. Verilerin bozulmasına karşı herhangi bir önlem alınmadığından veri güvenliği yetersizdir. Bu yöntemde verilerden ziyade fonksiyonlara ağırlık verilmektedir. Tasarım aşamasında ilk sorulan soru program ne iş yapacak sorusudur.



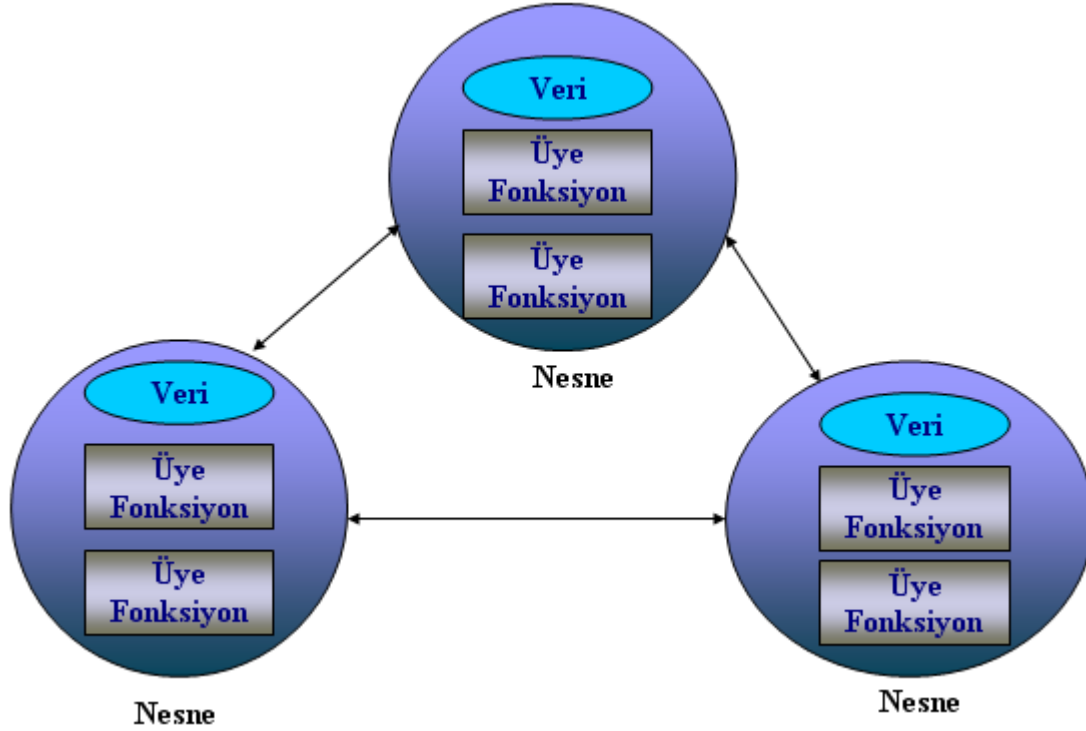
Şekil 2 Fonksiyonların Global değişkenlere erişimi

Nesneye Yönelimli programlama(NYP)

Nesne Yönelimli Programlama yöntemi uzun ve karmaşık programlar geliştirmek için tasarlanmıştır. Günümüzde bilgisayarda çözülmek istenen problemler eskiye oranla daha karmaşık ve büyüktür. Programlar uzadıkça, problemi analiz etmek ve çözmek zorlaşmaktadır. Donanım maliyetleri düşerken yazılım maliyeti artmaktadır.

Daha Güvenilir, dayanıklı, esnek, geliştirilebilir, Tekrar kullanılabilir ve daha kolay bakımı yapılabilen kaliteli programlar yazabilmek için yazılım problemlerine Nesne Yönelimli Analiz ile yaklaşmak daha fazla avantajlar sağlamaktadır. Şekil 1.5 de Nesne Yönelimli yaklaşım görülmektedir.

NYP yönteminde veriler ve bu veriler üzerinde işlem yapan fonksiyonlar bir araya getirilerek nesne (object) adı verilen bir yapı oluşturulmuştur. Nesneler birbirleri ile mesaj göndererek haberleşebilirler. Her bir nesnenin verilerle ifade edilen Nitelikleri(özellikleri), ve Fonksiyonlarla ifade edilen (Davranışları) bulunur. Veri ve fonksiyonların bir arada bulunmasına Paketleme(Encapsulation) adı verilir. Verilerin diğer nesnelerden gizlenebilmesi veya erişiminin kısıtlanması mümkündür. Buna veri gizleme(data hiding) denilmektedir.



Şekil 3 Nesne yönelimli yaklaşım

Nesne yönelimli programlamada programları güncellemek veya yeni eklentiler yapmak daha kolaydır. Nesne yönelimli programlama yönteminde sistem içinde bir çok nesne yer alır. Bu nesneler birbirlerinden farklı özellikler taşıyabildikleri gibi aynı veya benzer özellikler de taşıyabilirler. Nesneye dayalı programlamada tasarım yapılırken öncelikle veriler esas alınır.

Nesne Yönelimli programlamada Paketleme(Encapsulation), Soyutlama(Abstraction), Türeme (Derivation), Miras alma veya Kalitim (Inheritance) gibi kavramlar yer almaktadır.

Neler nesne olarak modellenenebilir.
Memur, müdür, Öğretmen, Öğrenci,
Nokta, Çizgi, Çember, Silindir,
Matris, Vektörler, Karmaşık sayı,...
Çerçeve, menü, Pencere,...

Nesneye dayalı yöntemin değerlendirilmesi.

Problemlerin çözümüne ilişkin daha gerçekçi bir model oluşturulabilir.
Oluşturulan bir nesne daha sonra defalarca tekrar kullanılabilir(reusability)
Programları güncellemek, hataları bulmak, bakımını yapmak daha kolaydır
Ekip çalışmaları için uygundur.