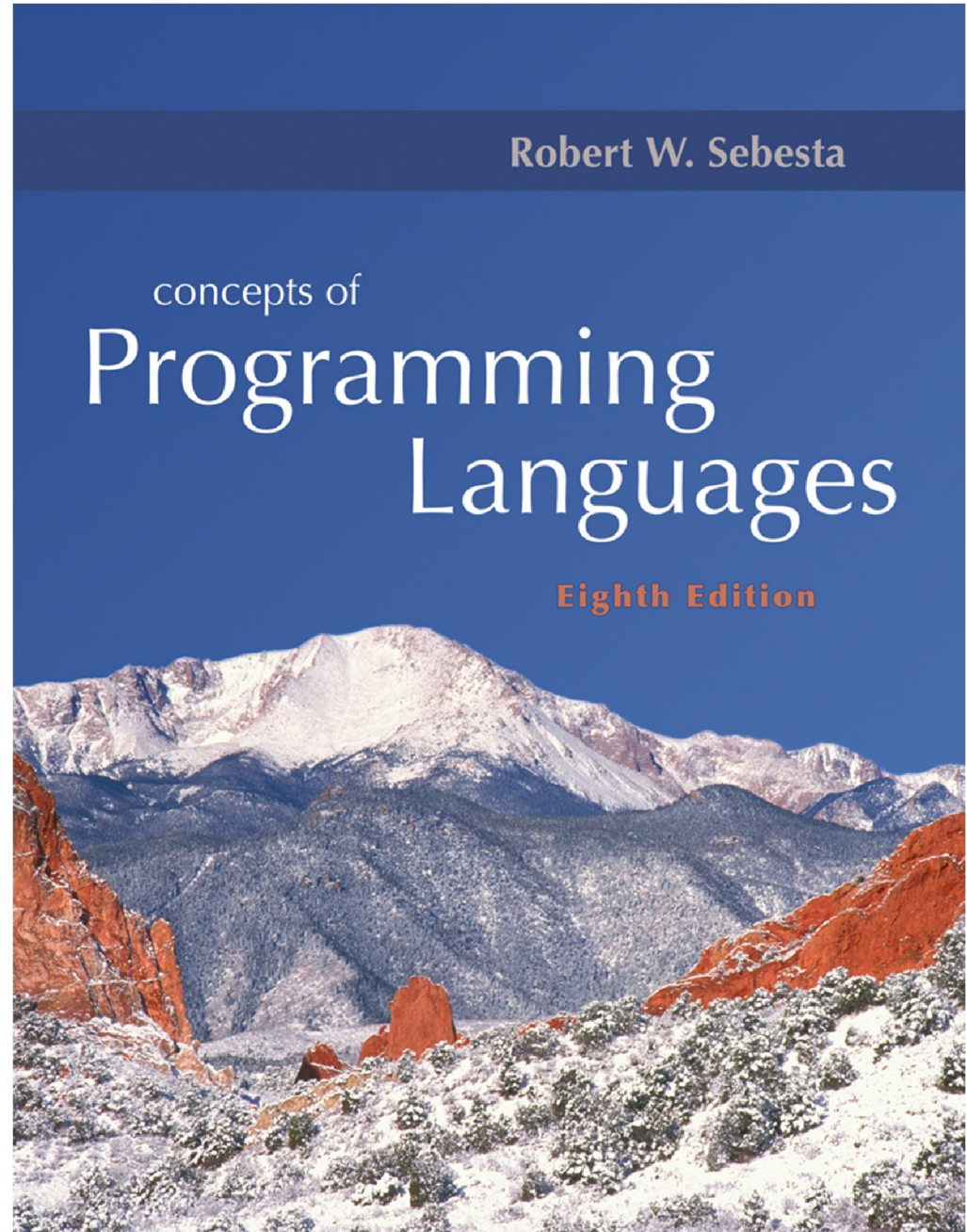


Bölüm 11

Soyut veri tipleri ve
kapsülleme
kavramları



ISBN 0-321-49362-1

11. Bölüm konuları

- Soyutlama kavramı
- Veri soyutlamasına giriş
- Soyut veri tipleri için tasarım konuları
- Dil örnekleri
- Parametrelili soyut veri tipleri
- Kapsülleme yapıları
- İsim kapsülleri

Soyutlama kavramı

- *Soyutlama*, bir varlığın sadece en önemli özelliklerini barındıran görüntü veya temsil şeklidir.
- Programlamada ve bilgisayar bilimlerinde temel önemi vardır
- Hemen hemen tüm dillerde altpogramalar aracılığı ile *süreç soyutlaması* (*process abstraction*) mevcuttur.
- 1980 sonrası tasarlanan hemen hemen tüm dillerde *veri soyutlaması* (*data abstraction*) desteği vardır.

Veri soyutlamasına giriş

- *Soyut veri tipi (SVT)* aşağıdaki özellikleri taşıyan kullanıcı-tanımlı veri tipidir
 - Tip nesnelerinin temsiliyeti ve tip üzerinde mümkün olan işlemler bir sentaktik ünite tanımlıdır
 - Nesnelerin temsiliyeti program ünitelerinden saklıdır ve nesneler üzerinde sadece tip tanımında verilen işlemler uygulanabilir

Veri soyutlamanın avantajları

- Birinci şartın avantajları
 - Programın düzenli olması, değiştirilebilirlik (veri yapısı ile ilgili herşey aynı yerde olduğu için), ve parça parça derleme
- İkinci şartın avantajları
 - Veri temsiliyeti saklı olduğu için kullanıcı kodu direkt olarak bu temsiliyete bağlı olamaz, dolayısı ile temsiliyet kullanıcı kodunu etkilemeden değişebilir

SVTler için dil gereksinimleri

- Tip tanımını kapsülleme için bir sentaktik birim
- Tip isimlerini ve altprogram başlıklarını kullanıcıya gösteren, ama tanımları saklayan bir yöntem

Dil örneği: C++

- Class (sınıf) kapsülleme aracıdır
- Sınıfta tanımlı fonksiyonlar (metodlar) sınıfın tüm örnekleri için geçerlidir
- Sınıfta tanımlı veri üyeleri (data members) her örneğin içinde ayrı ayrı mevcuttur
- Örnekler statik, yığıt–dinamik veya yığın–dinamik olabilirler

Dil örneği: C++ ...

- Bilgi saklama
 - *Private* (gizli varlıklar için)
 - *Public* (arayüz varlıkları için)
 - *Protected* (alt sınıflardan görülebilmesi gereken, ama dışarıdan görülememesi gereken varlıklar için)

Dil örneği: C++ ...

- Yapıcılar (constructors):
 - Örneklerin veri elemanlarını (data members) ilkleme yarar (nesneleri yaratmazlar)
 - Nesnenin bir kısmı yığın–dinamik ise hafızadan yer de alabilirler
 - Nesne ilklemeinde kullanılmak üzere parametre alabilirler
 - Örnek yaratıldığında otomatik olarak çağrılırlar
 - İsimleri sınıf ismi ile aynı

Dil örneği: C++ ...

- Yokediciler (destructors)
 - Örnek yokedilmeden hemen önce çağrılan metod (genellikle sadece yığından alınan hafızayı geri vermek için)
 - Örneğin ömrü bittiğine otomatik olarak çağırılırlar
 - İsimleri: ön tarafına ~ konmuş sınıf ismi

Dil örneği: C++ ...

- Friend (dost) fonksiyonlar / Friend (dost) sınıflar – seçici olarak gizli kısımları dışarıya göstermeye yarar.

C++ dilinde bir örnek

```
class stack {  
    private:  
        int *stackPtr, maxLen, topPtr;  
    public:  
        stack() { // a constructor  
            stackPtr = new int [100];  
            maxLen = 99;  
            topPtr = -1;  
        };  
        ~stack () {delete [] stackPtr;};  
        void push (int num) {...};  
        void pop () {...};  
        int top () {...};  
        int empty () {...};  
}
```

Parametrelili soyut veri tipleri

- Değişik tiplerde veri barındırabilen soyut veri tipleri tasarlamaya yarar
- Başka bir adı: jenerik sınıflar
- C++, Ada, Java, ve C# dillerinde parametrelili soyut veri tipi desteği vardır

C++ dilinde parametrelili soyut veri tipi

- Yüyük elemanı tipi sınıfı template aracılığı ile parametrelili hale getirilebilir

-

```
template <class Type>
class stack {
    private:
        Type *stackPtr;
        const int maxLen;
        int topPtr;
    public:
        stack() {
            stackPtr = new Type[100];
            maxLen = 99;
            topPtr = -1;
        }
        ...
}
```

Kapsülleme yapıları

- Büyük programların iki özel ihtiyacı var:
 - Alt programlara ayırma haricinde bir düzen kurabilme
 - Kısmi derleme (partial compilation) yöntemi (tüm programın değil, daha ufak birimlerin derlenebilmesi)
- Çözüm: biribiri ile ilgili altprogramların ayrı olarak derlenebilen birimlerde toplanması (derleme birimleri)
- Böyle birimlere *kapsüller* denir

İçiçe altprogramlar

- Altprogramları, onları kullanan altprogramların içinde tanımlamak sureti ile organize etmek
- Destekleyen dillerin bazıları: Pascal, Ada, Fortran 95, Python, Ruby

C dilinde kapsülleme

- Bir veya daha çok altprogram içeren dosyalar ayrı olarak derlenebilir
- Arayüz *header* dosyasına konur
- Dışarıdan görülmesi istenmeyen fonksiyon ve değişkenlerin önüne *static* sözcüğü konur
- `#include` önişlemci komutu: header dosyalarını programa dahil etmek için kullanılır

C++ dilinde kapsülleme

- C'ye benzer
- Sınıflar, kapsülleme için ek bir olanak sağlar. Birbiri ile alakalı fonksiyonlar sınıf çatısı altında toplanabilir. Ortak global veri sınıf değişkenlerinde saklanabilir.

Ada Paketleri

- Paketlerde isenilen sayıda veri ve altprogram deklarasyonu olabilir
- Paketler ayrı ayrı derlenebilir
- Paketlerin spesifikasyon ve implementasyon kısımları ayrı dosyalardadır ve kendi başlarına derlenebilirler

İsimle kapsülleme

- Çok sayıdaki global ismi (değişken, fonksiyon adı vs.) mantıki gruplara bölme yöntemi
- *İsimlendirerek kapsülleme (naming encapsulation)*
programdaki değişkenler, fonksiyonlar vs. için için yeni bir etki alanı yaratır
- C++ / C# isim uzayları (namespaces)
 - Her kütüphane kendi isim uzayında olabilir
 - Dışarıdan kullanılacak değişkenler ve fonksiyonların önüne isim uzayının adı yazılmalıdır

İsimle kapsülleme...

- Java Paketleri
 - Paketler de isim uzayı yaratabilir
 - Paketlerde birden çok sınıf tanımı olabilir
 - Paket kullanıcıları ya paket adı ile birlikte isimleri verirler, ya da *import* deklarasyonu kullanırlar

Özet

- Soyut veri tipleri (SVT) kavramı ve bu kavramın yeni dillerin tasarımında kullanılması önemli bir dönüm noktası idi
- SVTlerin iki önemli özelliği: veriyi ilgili işlemleri ile birlikte paketleme ve bilgi saklama
- ADA paketler vasıtası ile SVTlerin gerçekleştirilmesine destek verir
- C++, C# ve Java'da SVT desteği sınıflar aracılığı ile yapılır
- Ada, C++, Java, ve C# dillerinde parametrelili SVTler mümkündür
- C++, C#, Java, Ada dillerinde isim kapsüllemesi mümkündür