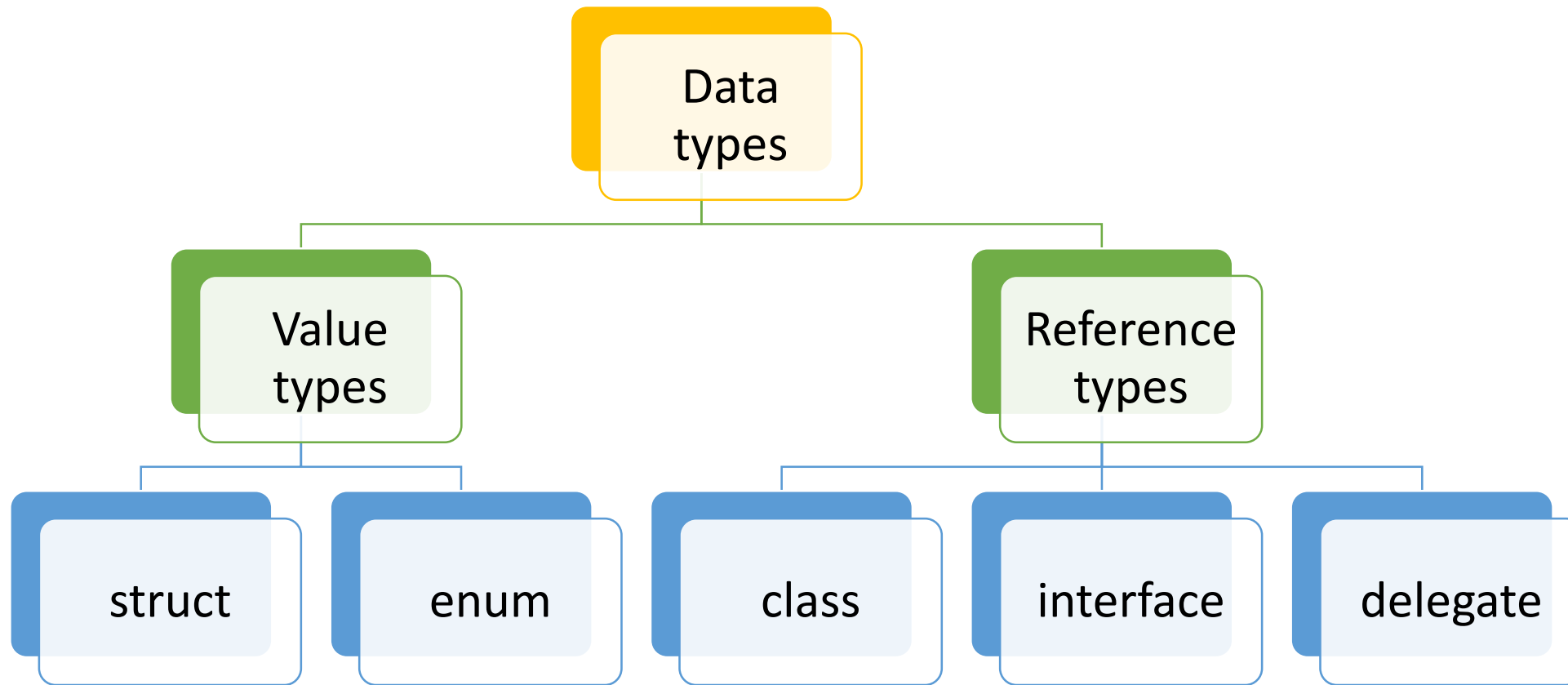


VERİ YAPILARILARI VE ALGORİTMALAR

Veri Tipleri (Data Types)



```

value_type
: struct_type
| enum_type
;

numeric_type
: integral_type
| floating_point_type
| 'decimal'
;

struct_type
: type_name
| simple_type
| nullable_type
;

integral_type
: 'sbyte'
| 'byte'
| 'short'
| 'ushort'
| 'int'
| 'uint'
| 'long'
| 'ulong'
| 'char'
;

simple_type
: numeric_type
| 'bool'
;

floating_point_type
: 'float'
| 'double'
;

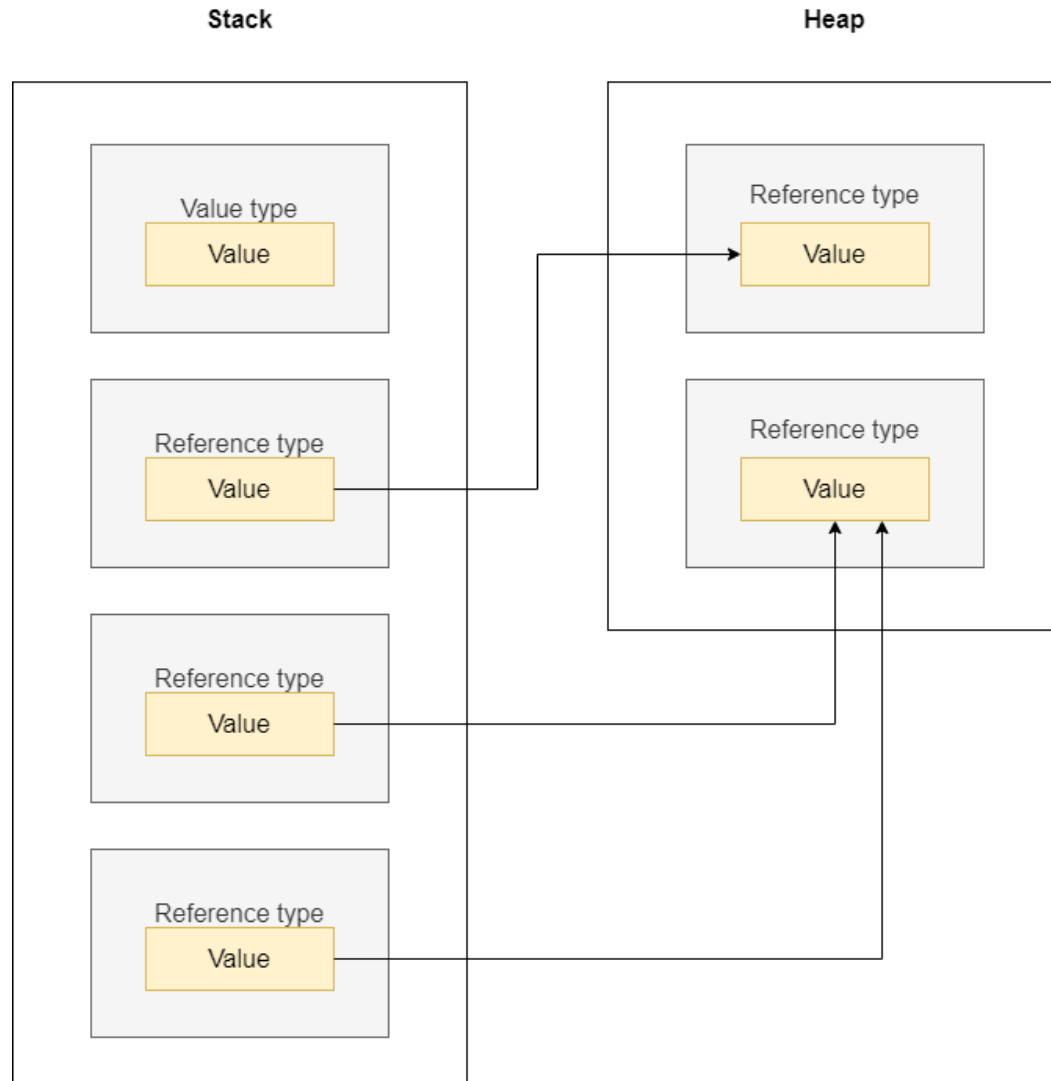
nullable_type
: non_nullable_value_type '?'
;

non_nullable_value_type
: type
;

enum_type
: type_name
;

```

Veri Tipleri



Veri Tipleri

- Programlama sırasında değer ve referans türleri arasındaki farklar çok kritik olabilir ve programın akışını tamamen değiştirebilir.
- Bu nedenle çalışan veri türlerinin belirtilen gruplardan hangisine ait olduğunu bilmek oldukça önemlidir.

Yerleşik Veri Türleri

Built-in data type (value)

short, int, double, char, decimal



Built-in data type (reference)

object, string, dynamic

Kullanıcı Tanımlı Veri Türleri

Custom/user defined data type (value)

struct



Custom/user defined data type (reference)

class

struct

- Farklı veri türlerini tek bir yapı altında toplamaya ihtiyaç duyulduğunda **struct** kullanılabilir.
- Temel amaç verinin bütününe temsil edecek alt veri türlerini bir yapı altında toplamaktır.
- Verinin organize edilmesi için kullanılan en eski programlama bileşenlerinden biridir.

struct

- Bazen program tanımları yaparken küçük ölçekli verileri organize etmek gerekebilir, bu tarz durumlarda sınıfların yerine **struct** yapısı da kullanılabilir.
- **struct** tanımı incelendiğinde class tanımlama yapısına oldukça benzediği görülmektedir.

struct

- Struct yapısı değer (**value**) tiplidir; sınıflar gibi referans tipli değildir.
- Struct kullanımı tamamlandıktan sonra bellekten hızlıca kaldırılabilir.
- Bu yapıların bellekten kaldırılmasını beklemek üzere **Garbage Collection**'ın beklenmesine gerek yoktur.
- Yani, bir başka ifadeyle sınıf yapısına kıyasla görece performansı daha iyidir.

struct

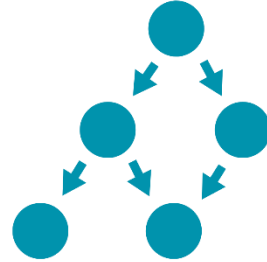
- Kalıtımı desteklemezler.
- struct yapısının da `System.ValueType`'dan türetilir.
- struct yapısı kalıtımın uygulanmasını desteklemez ancak `interface inheritance` destekler.
- struct yapısı bir metoda parametre olarak uygulanacak ise dikkat edilmelidir.

struct

- `struct` yapısı her zaman stack (yığında) tutulmaz. Bazı durumlarda, heap (öbek) bölgesinde de tutulabilirler.
- C# 7.2 ile birlikte `construct`, referans türler heap bölümünde ve `value` türleri tipik olarak stack bölümünde tutulurlar.
- Value tipleri sadece stack bölümünde tutulurlar.

class

- Nesne yönelimli programlamanın temel ögesidir.
- İçerisinde field, property, method, operatör gibi pek çok farklı üye barınır.
- Referans tipli bir veri türüdür.
- Sınıf ve arayüz kalıtımını destekler.



Veri Yapıları ve Algoritmalar

ZAFER CÖMERT

Öğretim Üyesi