

YMH 213 MESLEKİ İNGİLİZCE

Dr. Öğr. Üyesi Bihter DAŞ

Data Structures And The Models

(Veri Yapıları Ve Modeller)

Data structure means keeping the information in memory or storage in a meaningful format; in other words; it can be called "a kind of boxing process." In order to operate on an input value, it should be kept in the memory of computer. Programs operate on data stored in memory and store the results again in memory as data.

Veri yapısı, bilgilerin hafızada anlamlı bir formatta saklanması anlamına gelir; başka bir deyişle; buna "bir tür kutulama işlemi" denebilir. Bir giriş değeri üzerinde işlem yapabilmek için bilgisayarın belleğinde tutulması gerekir. Programlar hafızada saklanan veriler üzerinde çalışır ve sonuçları tekrar veri olarak hafızaya kaydeder.

Basic Data Structures

(Temel Veri Yapıları)

In general, there are many basic data structures or data types such as character, integer numbers, fractional numbers (real numbers) and string in all programming languages. Some programming languages also support different data structures such as complex number, logical value etc.

Genel olarak, tüm programlama dillerinde karakter, tamsayı sayılar, kesirli sayılar (gerçek sayılar) ve dizi gibi birçok temel veri yapısı veya veri türü vardır. Bazı programlama dilleri ayrıca karmaşık sayı, mantıksal değer vb. gibi farklı veri yapılarını destekler.

Characters (Karakterler)

Being one of the most basic data structures character is a structure for characters or words and sentences. Encoding process is done according to a specific **character table** such as **ASCII** or **Unicode**; when the table is changed, **corresponding** character codes will also change and thus information contained in data will be different.

En temel veri yapılarından biri olan karakter, cümleler ve kelimeler veya karakterler için bir yapıdır. Kodlama işlemi, ASCII veya Unicode gibi belirli bir karakter tablosuna göre yapılır; tablo değiştirildiğinde, karşılık gelen karakter kodları da değişecek ve bu nedenle verilerin içerdiği bilgiler farklı olacaktır.

Strings (Diziler)

String is a data structure that is used to hold the whole text or any sentence, word and syllable of the text in memory. In fact, it contains a sequence of characters; however it differs from an ordinary character sequence in that number of its elements is known or an end of string character is used.

Diziler, tüm metni veya metnin herhangi bir cümlesini, kelimesini ve hecesini bellekte tutmak için kullanılan bir veri yapısıdır. Aslında, bir dizi karakter içerir; bununla birlikte, elemanlarının sayısının bilinmesi veya bir dizge sonu karakteri kullanılması bakımından sıradan bir karakter dizisinden farklıdır.

Arrays and Matrices (Diziler ve Matris)

In array data structure, data which belongs to the same set is kept in memory **consecutively**. So if you know the **starting address** or the name of an array, you can easily reach (read, change, add etc.) any elements you want.

Dizi veri yapısında, aynı sete ait veriler arka arkaya bellekte tutulur. Yani bir dizinin başlangıç adresini veya adını biliyorsanız, istediğiniz öğeye kolayca ulaşabilirsiniz (okuma, değiştirme, ekleme vb.).

Creating a Struct (Struct Oluşturmak)

Creating a struct means bringing more than one data structure together and creating a new data structure. Linked list, tree and other special data models require such new data structures; data structures containing more than one variable such as time, data etc. can also be declared by using structs.

Yapı oluşturmak, birden fazla veri yapısını bir araya getirmek ve yeni bir veri yapısı oluşturmak anlamına gelir. Bağlantılı liste, ağaç ve diğer özel veri modelleri, bu tür yeni veri yapılarını gerektirir; zaman, veri vb. birden fazla değişken içeren veri yapıları da yapılar kullanılarak bildirilebilir.

Data Models and Data Types

(Veri Modelleri ve Veri Türleri)

Data models are **conceptual** methods to show the relational and sequential order between data that may be considered as belonging to the some set. Each program is based on a data model at least; while small-scale programs are based on basic data structure of programming language, some programs requires complex data structure and models.

Veri modelleri, bir kümeye ait olduğu düşünülebilecek veriler arasındaki ilişkisel ve sıralı düzeni gösteren kavramsal yöntemlerdir. Her program en azından bir veri modeline dayanmaktadır; küçük ölçekli programlar, programlama dilinin temel veri yapısına dayanırken, bazı programlar karmaşık veri yapısı ve modelleri gerektirir.

Linked List Data Model

(Bağlı Liste Veri Modeli)

The list data model is based on the principle of keeping data belonging to the some set **consecutively** in memory. Data may or may not have a particular order (sequential etc.); the important thing is to keep all the data is **successive** order.

Liste veri modeli, bazı kümelere ait verilerin ardışık olarak bellekte tutulması ilkesine dayanmaktadır. Veriler belirli bir sıraya sahip olabilir veya olmayabilir (sıralı vb.); önemli olan tüm verileri ardışık sırada tutmaktır.

Linked List Data Model

(Bağlı Liste Veri Modeli)

The most simple list data model is a **one-dimensional array**; each element of array holds one of the elements of set and unless otherwise stated, elements are listed according to the order they are added to the list.

En basit liste veri modeli tek boyutlu bir dizidir; dizinin her elemanı setin elemanlarından birini tutar ve aksi belirtilmedikçe elemanlar listeye eklendikleri sıraya göre listelenir.

Tree Data Model (Ağaç Veri Modeli)

In tree data model, data belonging to same set are interconnected in such a way that they form an **inverted tree** as. The **roots** of tree are at the top of the tree data model and the **leaves** of tree are shown at the bottom. Tree data model provides an effective solution especially when data set is large and the process of searching is used frequently.

Ağaç veri modelinde, aynı kümeye ait veriler, tersine çevrilmiş bir ağaç oluşturacak şekilde birbirine bağlanır. Ağacın kökleri, ağaç veri modelinin tepesindedir ve ağacın yaprakları altta gösterilmiştir. Ağaç veri modeli, özellikle veri setinin büyük olduğunda ve arama süreci sıkça kullanıldığında etkili bir çözüm sağlar.

Graph Data Model

(Grafik Veri Modeli)

In the graph data model, data belonging to the same set are by using **nodes** and **edges** to link them each other as. **Nodes** show the **junction points**, and edges show the relationship between nodes. Data or a portion o data can be kept both in nodes and the information part of edges.

Grafik veri modelinde, aynı kümeye ait veriler düğümler ve kenarlar kullanılarak birbirlerine bağlanırlar. Düğümler birleşme noktalarını gösterir ve kenarlar düğümler arasındaki ilişkiyi gösterir. Veriler veya verilerin bir kısmı hem düğümlerde hem de kenarların bilgi kısmında tutulabilir.

State Machine Data Model (Durum Makinesi Veri Modeli)

State machine data model is used to describe and reveal the behavior of a system; it uses **states** in order to describe the behavior of system in operating systems, compilers and controlling software, and exposes the system by using the transition conditions between **states**.

Durum makinesi veri modeli, bir sistemin davranışını tanımlamak ve ortaya çıkarmak için kullanılır; işletim sistemlerindeki sistemin davranışını açıklamak için durumlar kullanılır, derleyiciler ve kontrol yazılımı ve durumlar arasındaki geçiş koşullarını kullanarak sistemi ortaya çıkarır.

State Machine Data Model

(Durum Makinesi Veri Modeli)

State machine is available in many areas of the software application; for example a robot arm movement, decryption, process control in real-time operating systems, and software solution for control subsystems in general.

Durum makinesi, yazılım uygulamasının birçok alanında mevcuttur; örneğin bir robot kolu hareketi, şifre çözme, gerçek zamanlı işletim sistemlerinde işlem kontrolü ve genel olarak kontrol alt sistemleri için yazılım çözümü.

Database Relational Data Model (Veritabanı İlişkisel Veri Modeli)

Database relational data model is one of four or five classes used in database application. Data is based on **relationships** established through tables.

Veritabanı ilişkisel veri modeli, veritabanı uygulamasında kullanılan dört veya beş sınıftan biridir. Veriler, tablolar aracılığıyla kurulan ilişkilere dayanır.

Database Relational Data Model (Veritabanı İlişkisel Veri Modeli)

By using a query language named **SQL** (Structured query Language), queries can be done in database. Many database management systems including Access, **Microsoft SQL**, **ORACLE**, **SYBASE** and Ingres support relational data model.

SQL (Structured Query Language) adlı bir sorgu dili kullanılarak veri tabanında sorgulama yapılabilir. Access, Microsoft SQL, ORACLE, SYBASE ve Ingres gibi birçok veritabanı yönetim sistemi ilişkisel veri modelini destekler.

References

- Rifat Çölkesen, Introduction to Computer Engineering
- Doç. Dr. Murat Karabatak ders notları
- <https://www.pearson.com/>
- <https://www.britishcouncil.al/>