

Adatszerkezetek

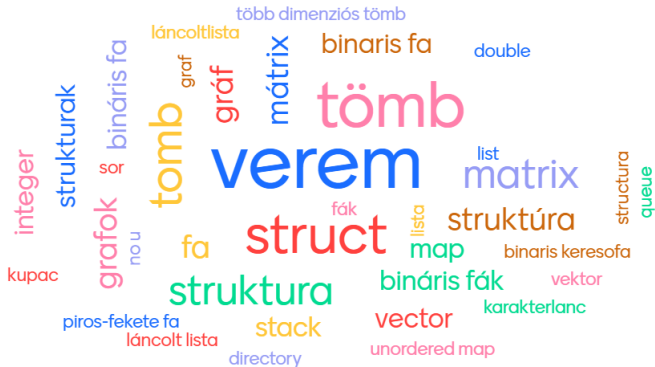
01. Alapfogalmak

Vekov Géza

2023. március 1-8.



Memóriahasználat



Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus

Szerkezet

Definíciók, fogalmak

Adattípus (adatelem)

azonos tulajdonságú **értékek** halmaza, amelyből az adatok származhatnak, valamint azok a **műveletek**, amelyek ezekkel az adatokkal elvégezhetők.

Absztrakt adattípus

Az adattípus leírása és definíciója, de nincs hivatkozás az ábrázolásmódra és az implementálás részleteire.

Értékek szerinti osztályozás

■ Egyszerűek

- atomi, primitív
- nem bonthatóak tovább
- Például: karakterek, egész számok, valós számok, stb.

■ Összetettek

- az értékek alkotóelemekre bonthatóak
- Például: karakterláncok, tétel (striktúra), stb.

Adatszerkezet

Adatok csoportosulása, amelyben az adatok pontos szabályok szerint szerveződnek, valamint az szerkezettel elvégezhető **műveletek**.

Absztrakt adatszerkezet

Az adatszerkezet leírása és definíciója, de nincs hivatkozás az ábrázolásmódra és az implementálás részleteire.

Adottak

- Az adatelemek/értékek
- Az elemek közötti kapcsolat
- A lehetséges műveletek

Példa

Tömb, láncolt lista, gráf, bináris fa, hasítótábla

Absztrakciós szintek

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus
Szerkezet

Absztrakt (logikai) adattípus

- A szerkezet lényeges tulajdonságait veszi figyelembe
- Nem törődik az ábrázolással és az implementáció részleteivel

Virtuális adattípus

- A programozási nyelv szabályainak megfelelően deklarált adattípus

Fizikai adattípus

- A számítógép memóriájában létrehozott adatcsomag (bájtokban tárolva)

Absztrakciós szintek

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Ösztályozás
Memóriahasználat
Elemtípus
Szerkezet

Példa

		Absztrakt	Virtuális	Fizikai
Adattípusok	egyszerű	személyek száma	int x;	egész szám 2 bájton ábrázolva
	összetett	pontszámok listája	int a[100];	legtöbb 100 valós szám, 100 valós számnak lefoglalt helyen.

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus

Szerkezet

Funkcionális specifikáció

Funkcionális specifikáció

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

**Funkcionális
specifikáció**

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus
Szerkezet



Funkcionális specifikáció

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás
Memóriahasználat
Elemtípus
Szerkezet

A **funkcionális specifikáció** leírja, hogyan működik az adatszerkezet.

Érdekel

- Eltárolandó adatok
 - Mit kell eltárolni?
 - Típus, értéktartomány definiálása.
- Szükséges (és lehetséges) műveletek
 - Mit kell végrehajtani?

Nem érdekel

- A "hogyan", az implementációs részletek ezen a szinten **nem fontosak**.

Előfeltételek

- A bemeneti adatok azon tulajdonságai, amelyeknek teljesülniük kell ahhoz, hogy a leírt műveletet végrehajthassuk és az helyesen működjön
- (Az előfeltételeket le kell ellenőrizni, és amennyiben ezek nem teljesülnek, valamilyen módon kezelni kell.)

Utófeltételek

- A kimeneti adatok azon tulajdonságai, amelyeknek feltétlenül teljesülniük kell a feladat követelményeinek megfelelően.
- Nem tartalmazzák azt, hogy hogyan jutunk az illető adatokhoz, csak azt, hogy azok milyenek kell legyenek.

Példa

Az első szeminárium anyagában (Canvas → Files)

Funkcionális specifikáció

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus
Szerkezet

Fontos

A specifikációban **nincs** szó arról, hogy:

- hogyan/milyen adatszerkezetben tároljuk el az adatokat;
- a műveleteket hogyan implementáljuk;
- **DE**, az pontosan ki kell derüljön, hogy az adott művelet mit kell elvégezzen!

A specifikációban **lehet** szó arról, hogy:

- mi legyen a műveletek végrehajtási idejének **felső korlátja**.

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

**Implementálási
szempontok**

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus

Szerkezet

Implementálási szempontok

Hogyan/milyen adatszerkezetben tároljuk az adatokat?

- Mit kell eltárolni?
- Milyen műveletekre lesz szükség?

Például

Karakterlánc esetén (ld. szeminárium)

```
int n;          // karakterlánc aktuális hossza  
char s[80];    // maga a karakterlánc
```


Adattípus implementálása

A program azon része, amely tartalmazza a:

- Típus ábrázolását
- Műveleteket leíró alprogramokat

Lehetőségek

- Modulok implementálása
- Struktúrák használata
- Osztályok használata

Adatok védelme

- A felhasználó előtt rejtve vannak az adatok és az implementáció részletei.
- A felhasználó programban az eltárolt adatokhoz csak a specifikált műveletek segítségével férhetünk hozzá.

Függetlenség

Az esetleges módosítások még akkor sem fogják a felhasználót érinteni, ha teljesen megváltoztatunk mindent: az adatszerkezetet és a műveletek implementálását.

- A funkciók változatlanul ugyanazok maradnak
- A felhasználó programon nem kell változtatni
- A hatékonyság változhat

Implementálás: részletek

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás
Memóriahasználat
Elemtípus
Szerkezet

Hatékonyság

Az adatszerkezet kiválasztásakor két fő szempontot kell figyelembe venni:

- Minél hatékonyabban tároljuk az adatokat.
- Minél hatékonyabban tudjuk implementálni a szükséges műveleteket.

Legtöbbször nincs univerzálisan jó megoldás, DE a feladat függvényében keressük meg a lehető legjobb kompromisszumot.

Irányelv

Lehet-e jobban?

Egy példa

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus
Szerkezet

Például

Egy *névlista* ábrázolható

- **Tömbben**, így ha szükségünk van egy adott névhez csatolt egyéb információra, lineáris (szekvenciális) keresést végzünk.
- **Láncolt listában**, így minden elem, a feldolgozandó információkon kívül tartalmazza az illető elem után következő elem címét is. (Még a rendezett listában is szekvenciális kereséssel dolgozunk.)
- **Rendezett tömbben**, így alkalmazható a bináris keresés algoritmus
- **Keresőfában**
- **Hasító táblában**
- Stb.

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus

Szerkezet

Összefoglaló

Az eddigiek alapján

Absztrakt adatszerkezetek használatának az előnyei:

- pontos specifikáció
- modularitás
- adatok védelme
- egyszerűség
- egységesség
- az implementálás függetlensége

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus

Szerkezet

Osztályozás

Osztályozás - memóriahasználat szerint

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus

Szerkezet

Változhat-e az adatszerkezet elemeinek a száma?

Aszerint, hogy használat közben változhat-e az adatszerkezet elemeinek a száma, három csoportba osztjuk őket:

- **statikus**
- **félstatikus**
- **dinamikus**

Statikus adatszerkezet

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus
Szerkezet

Tulajdonságok

- Az adatszerkezet által elfoglalt tárrész mérete ugyanakkora a végrehajtás teljes időtartama alatt
- Ezen belül az elemek ugyanazon a helyen találhatóak
- Az adatszerkezet egy adott pillanatban "megszületik", és változatlan állapotban létezik a program végrehajtása alatt

Fontos

- A változatlanság a szerkezetre értendő, és nem az elemek értékére

Példa

- tömb, halmaz

Dinamikus adatszerkezet

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus

Szerkezet

Tulajdonságok

- Nem ismerjük előre a feldolgozandó elemek számát
- A szükséges tárrész a végrehajtás ideje alatt "igény szerint" foglalódik le

Fontos

- A tárfoglalás dinamikus az elemek szintjén és az adatszerkezet szintjén is

Példa

- láncolt lista, fa

Félstatikus adatszerkezet

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat
Elemtípus
Szerkezet

Tulajdonságok

- Az adatszerkezet számára lefoglalt tárrész mérete állandó
- Ezen belül az elemek a végrehajtás időtartama alatt változtatják a helyüket

Fontos

- a tárfoglalás statikus a strukrúra szintjén és dinamikus az elemek szintjén
 - Mivel ezt a tulajdonságot nem támogatja a programozási környezet, ezért vagy statikusan vagy dinamikusán ábrázoljuk ezeket.

Példa

- verem, várakozási sor, hasító tábla

Statikus, félstatikus, dinamikus adatszerkezetek

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus
Szerkezet

Megjegyzés

- A fenti osztályozás az absztrakt adatszerkezetre érvényes.
- Ugyanazt az absztrakt adatszerkezetet implementálhatjuk statikusan vagy dinamikusan is

Például

- Egy halmazt általában statikusan implementálunk
- Egy vermet általában dinamikusan implementálunk
(így a beszúrásokat és törléseket hatékonyabban végezhetjük el)

Osztályozás - elemtípus szerint

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

Memóriahasználat

Elemtípus
Szerkezet

Milyen az adatszerkezet elemeinek a típusa?

- **Homogén** - minden elem ugyanolyan típusú
- **Heterogén** - az elemek típusa különbözik

Osztályozás - elemek közötti kapcsolat szerint

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

Implementálási
szempontok

Összefoglaló

Osztályozás

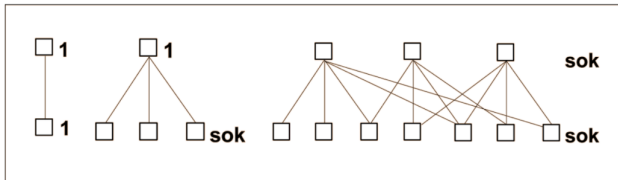
Memóriahasználat

Elemtípus

Szerkezet

Milyen kapcsolatban állnak egymással az adatelemek az adatszerkezetben?

- Az elemek között nincs kapcsolat, csak egy szerkezethez tartoznak
- Az elemek között létezik kapcsolat
 - 1 egy kapcsolatban van eggyel ($1 : 1$)
 - 2 egy kapcsolatban van sokkal ($1 : N$)
 - 3 sok kapcsolatban van sokkal ($N : N$)



Nincs kapcsolat

- Az elemek egymástól függetlenek
- Például: *halmaz*

Lineáris adatszerkezet

- Létezik egyetlen első és utolsó elem
- Minden elemnek van előző és rákövetkező eleme
- Sorszámozható
- Például: *lista, tomb, verem, stb.*

Fa típusú adatszerkezet

- Egy elem több elemhez kapcsolódik egy hierarchia szerint

Gráf típusú adatszerkezet

- Több elem kapcsolódik több elemhez

Osztályozás - elemek közötti kapcsolat szerint

Adatszerkezetek

Vekov Géza

Definíciók,
fogalmak

Adattípus
Adatszerkezet

Funkcionális
specifikáció

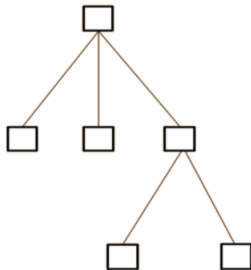
Implementálási
szempontok

Összefoglaló

Osztályozás
Memóriahasználat
Elemtípus
Szerkezet



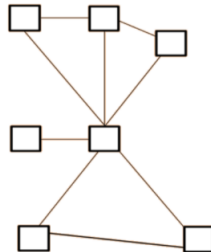
halmaz



fa



lineáris lista



gráf