

Egyszerű adattípusok, összetett adattípusok

Egyszerű adattípusok

Minden programnyelvben megtalálhatóak típusok, idetartoznak az egész és valós számok, a logikai értékek és a karakterek tárolására alkalmas típusok. A egyes kategóriába tartozó numerikus típusok értékészlete a programnyelv illetve processzor függvénye. Szokták még natív adattípusnak is nevezni, jelezve, hogy a processzor számára egyszerűen, legtöbbször egy művelettel feldolgozható adatokról van szó.

Néhány példa egyszerű adattípusra:

- **Egész típusok:** byte, shortint, smallint, integer, longint
Méretük: 8 -128 bit
- **Valós típusok:** real, float, single, double
Méretük: 48 -128 bit
- **Logikai típus:** boolean (true - false, 1 - 0)
Méretük: 1-32 bit (annak ellenére, hogy a tároláshoz elég 1 bit is)
- **Karakter:** char.(egy karakter tárolására alkalmas)
Méretük: 8-32 bit, attól függően, hogy egyszerű ASCII vagy egy bővített (pl.: UTF-8) karakter készletet használunk.

Összetett adattípusok

Összetett adattípus esetén több adat tárolását feltételezzük.

Ezen több adat lehet több elemi típusú adat, de lehet több összetett típusú adat is, sőt, akár vegyes is lehet.

Összetett adattípusnak minősül a sztring típus, amennyiben karakterenként kezeljük (a sztring több elemi adattípus, a karakterből felépített összetett adattípus).

Hasonlóan összetett adattípusnak minősül a rekord, a tömb, a sor, a verem, a lista, és még számtalan más is.

Összetett adattípusok tárolás szerint

Összetett adattípus esetén több adat tárolását feltételezzük.

Amennyiben ezen adatok mindegyike egyforma típusú, úgy 'homogén'-nek nevezzük az összetett adatszerkezetet. Ilyen például a tömb.

Amennyiben ezen adatok különböző típusúak, úgy 'inhomogén', vagy 'heterogén' adatszerkezetéről beszélünk. Erre a legjobb példa a rekord.

Amennyiben több elem a memóriában egymást követő területen helyezkedik el, úgy folytonos reprezentációjúnak nevezzük. Ilyen a rekord is, a tömb is.

Amennyiben az elemek a memóriában független területeken (elszórva) helyezkednek el, úgy szétszórt reprezentációjúnak nevezzük. Ilyen a klasszikus sor adatszerkezet.

Amennyiben több adat tárolása esetén a darabszámot előre be kell jelenteni a deklaráció során, úgy statikus helyfoglalású összetett adatszerkezetéről beszélünk. Ilyen a rekord és a tömb is.

Amennyiben előre nem kell bejelenteni a darabszámot, hanem az futás közben szabadon változtatható, úgy dinamikus méretű adatszerkezetéről beszélünk.

Összetett adattípusok kezelés szerint

Összetett adattípus esetén több adat tárolását feltételezzük.

A több adat esetén beszélni kell az adathozzáférési sebességről, stratégiákról.

Amennyiben ezen adatok bármelyikének elérése egyforma sebességgel történik, úgy 'véletlen elérésű' (Random Access) adatszerkezetéről beszélünk (ilyenek a tömbök és a rekordok például).

A klasszikus lista esetén a listában hátrébb lévő elemek elérése tovább tart, mint az előrébb lévő elemeké. Az ilyen adatszerkezetet szekvenciális elérésűnek nevezzük.

A SOR adatszerkezet speciális adatelérési technikáját FIFO-nak (First In First Out) nevezzük.

A VEREM adatszerkezet speciális adatelérési technikáját LIFO-nak (Last In First Out) nevezzük.