

Lineáris adatszerkezetek: tömb és vektor listaszerkezetek, gyűjtemények. FIFO és LIFO vermek megvalósítása.

Egyenesvonalú (lineáris) adatszerkezetek

- Absztrakt $\leftarrow \rightarrow$ ☐ Konkrét adatszerkezetek
- Specifikáció $\leftarrow \rightarrow$ Megvalósítás
- Szekvenciális elérés (fájl, perifériák)
- Véletlen elérés (vektor, fájl, perifériák, RDBMS-ek, Random Access Memory)
- (Alap)műveletek és hatékonyságuk
 - Olvasás (Szelektor)
 - Beírás
 - Létrehozás (Konstruktor) és Beszúrás
 - Törlés
 - Keresés
 - Rendezés

A tömb azonos típusú elemekből álló, sorosan tárolt adatcsoporthoz.

Amennyiben hossza előre ismert, statikus tömbnek nevezik, ha hossza a program futása során (run time) változik, dinamikus tömbnek, vagy listának nevezzük.

Többdimenziós tömbök

- Kétdimenziós tömb: egydimenziós tömbök tömbje. Pl: m méretű tömbökből felépített n méretű tömb:
- `típus[n][m]` nevek;

Rekordszerkezet

- Több, esetleg különböző részekből összetett adattípus
- Osztály, függvények nélkül, csak látható tagokkal
- Véges, ha az összetevők végesek, és ha nem rekurzív
- Pascal: változtatható rekord: típusegyesítés
- Feltehetőleg: helytakarékoság célzatából

Vermek(FIFO és LIFO)

LIFO (Last In First Out)

Amit legelsőnek raktunk bele azt utoljára tudjuk csak kivenni.

FIFO (First In First Out)

Amit elsőnek rakunk be azt elsőként is tudjuk kivenni. (Sor)

Vermek megvalósítása

```
public class verem {
    int max = 100;
    int tomb[] = new int[max];
    int teteje=0;
    public boolean ures() {
        return (teteje == 0);
    }
    public void beszur(int elem) {
        tomb[teteje++] = elem;
    }
    public int kivesz() {
        return (tomb[--teteje]);
    }
}
```

```

public boolean tele() {
    return (teteje==max);
}
public class sor {
int max = 100;
int tomb[] = new int[max];
int feje=0;
int farka=0;
boolean ures=true;
boolean tele=false;
    public boolean ures() {return (ures);}
    public boolean tele() {return (tele);}

public void beszur(int elem) {
    tomb[farka]=elem;
    ures = false;
    if (farka<max-1) farka++;
        else farka=0;
    if (farka == feje) tele = true;}
public int kivesz() {
    int elem;
    elem = tomb[feje];
    tele = false;
    if (feje<max-1) feje++;
        else feje = 0;
    if (feje == farka)
        ures = true;
    return (elem);}

```