

## Szubrutinok

- Szubrutin:
  - Egy olyan jól meghatározott, önálló tevékenységsort megvalósító, formailag is elkülönülő programrész, amelynek tényleges végrehajtásához egy másik, a szubrutint aktivizáló, azt „meghívó” programrész is szükséges.
- Előnyök:
  - Egységbezárás
  - Tagoltság, könnyebb karbantarthatóság
- Megjegyzés:
  - Az **objektumorientált programozás** objektumtípusai (osztályai) az adatoknak és a rajtuk dolgozó algoritmusoknak egy még általánosabb egységét valósítják meg azáltal, hogy az adatokhoz nemcsak egy, de tetszőleges számú algoritmus rendelhető.

## Szubrutinok

### ■ Deklarálás:

- Definiáljuk a szubrutin által végrehajtandó tevékenységeket és a külvilággal való kapcsolattartás módját és szabályait.

### ■ Hívás:

- Felhasználjuk a szubrutin szolgáltatásait, vagyis definiálva a kapcsolattartás eszközeit, kiváltjuk a tevékenységsor egy végrehajtását.
- A kapcsolattartás legfőbb eszközei a **paraméterek**.

### ■ Megjegyzés:

- A helyes programozási stílus a **teljes paraméterezésre** való törekvés.



## Szubrutinok

### ■ Deklarálás:

#### ■ Eljárás:

Eljárásnév(Formális paraméterlista)  
/\* Az eljárás utasításai \*/

#### ■ Függvény:

Függvéynév(Formális paraméterlista)  
/\* A függvény utasításai \*/  
**return** Függvényérték

- A *Formális paraméterlista* változók vesszővel elválasztott sorozata.



## Szubrutinok

### ■ Hívás:

#### ■ Eljárás:

Eljárásnév(Aktuális paraméterlista)

#### ■ Függvény:

Pl.

Változó  $\leftarrow$  Függvéynév(Aktuális paraméterlista)

- Az *Aktuális paraméterlista* kifejezések vesszővel elválasztott sorozata.
- Az eljárás a nevével, „utasításszerűen” hívható, míg a függvény kifejezésekben, a kifejezések részeként hívható.



## Szubrutinok

- **Paraméterátadás:**
  - Érték szerint (input jelleg)
  - Cím szerint (input, output jelleg)
- A szubrutin formális paraméterlistáján csak az **input** ill. **output** adatok változói szerepelnek, a munka jellegű adatokat a szubrutin lokális munkaváltozóiban tároljuk.
- A felhasználóval való **kommunikáció** (pl. adatbekérés, eredménykiírás) ekkor pl. a főprogramban, mint hívó környezetben programozható.



## Szubrutinok

- Megjegyzés:
  - A paramétereket a szubrutinok egy speciális memória, a **verem** segítségével, azon keresztül veszik át. Ide kerülnek a szubrutinok lokális változói is.
  - A nagy helyigényű adatok (pl. tömbök) paraméterként való átadásakor célszerű **cím szerinti** paraméterátadást választani.
  - A szubrutinok lehetnek **rekurzívak** is, azaz önmagukat is meghívhatják.
  - Egy szubrutin **eljárás** ill. **függvény** formában is megírható, a két forma egymással ekvivalens módon helyettesíthető.

## Szubrutinok

- Feladat:** Cseréljük fel két egész típusú változó tartalmát!

Funkció	Azonosító	Típus	Jelleg
Egyik szám	A	Egész	I, O
Másik szám	B	Egész	I, O
Csereváltozó	CS	Egész	M

/\* Deklarálás \*/

CSEREL(A,B)

CS  $\leftarrow$  A

A  $\leftarrow$  B

B  $\leftarrow$  CS

/\* Hívás \*/

Be: A,B

CSEREL(A,B)

Ki: A,B

## Szubrutinok

- **Feladat:** Határozzuk meg két egész szám közül a kisebbet!

Funkció	Azonosító	Típus	Jelleg
Egyik szám	A	Egész	I
Másik szám	B	Egész	I
Az eredmény	ER	Egész	O

/\* Deklarálás \*/

MINIMUM(A,B)

**if** A<B

ER ← A

**else**

ER ← B

**return** ER

/\* Hívás \*/

Ki: "5 és 3 közül a kisebb:", MINIMUM(5,3)



## Szubrutinok

- Hatékonyság:
  - Ha egy algoritmus pl.  $2n^2+3n+1$  műveletet végez (ahol  $n$  jelöli az algoritmusban résztvevő elemek számát), akkor kellően nagy  $n$ -re a legnagyobb kitevős (jelen esetben a négyzetes) tag dominál. A konstans szorzóktól és a kisebb kitevős tagoktól eltekintve azt mondjuk, hogy ez az algoritmus  $n^2$  **nagyságrendű**.



## Minimumhelyek keresése

- **Feladat:** Adott egy  $N$  elemű adatsorozat. Keressük meg a legkisebb elemét, állapítsuk meg hányszor és hol található meg ez az elem a sorozatban!

## Minimumhelyek keresése

- **Feladat:** Adott egy  $N$  elemű adatsorozat. Keressük meg a legkisebb elemét, állapítsuk meg hányszor és hol található meg ez az elem a sorozatban!

Funkció	Azonosító	Típus	Jelleg
A sorozat elemei	A	Egydimenziós, tetszőleges elemtípusú tömb	I
A sorozat elemeinek száma	N	Egész	I
A legkisebb elem értéke	MIN	Az A tömb elemeivel megegyező típusú	M, O
A legkisebb elem darabszáma	DB	Egész	M, O
A legkisebb elem indexei	HELY	Egydimenziós egész tömb	M, O
Ciklusváltozó	I	Egész	M

## Minimumhelyek keresése

```
/* Minimumhelyek keresése */  
MINKERESES(A,N,MIN,DB,HELY)  
/* Minimum meghatározása */  
MIN  $\leftarrow$  A[1]  
for I  $\leftarrow$  2,N  
    if A[I]<MIN  
        MIN  $\leftarrow$  A[I]  
/* Minimumhelyek */  
DB  $\leftarrow$  0  
for I  $\leftarrow$  1,N  
    if A[I]=MIN  
        DB  $\leftarrow$  DB+1  
        HEY[DB]  $\leftarrow$  I
```



## Átlag és szórás

- **Feladat:** Adott egy  $N$  elemű számsorozat. Határozzuk meg az elemek átlagát és szórását!



## Átlag és szórás

- **Feladat:** Adott egy  $N$  elemű számsorozat. Határozzuk meg az elemek átlagát és szórását!

Funkció	Azonosító	Típus	Jelleg
A sorozat elemei	A	Egydimenziós valós tömb	I
A sorozat elemeinek száma	N	Egész	I
Az elemek átlaga	ATL	Valós	O
Az elemek szórása	SZ	Valós	O
Az összegzésekhez	OSSZ	Valós	M
Ciklusváltozó	I	Egész	M

## Átlag és szórás

```
/* Átlag és szórás */  
SZAMOL(A,N,ATL,SZ)  
/* Átlag */  
OSSZ  $\leftarrow$  0  
for I  $\leftarrow$  1,N  
    OSSZ  $\leftarrow$  OSSZ+A[I]  
ATL  $\leftarrow$  OSSZ/N  
/* Szórás */  
OSSZ  $\leftarrow$  0  
for I  $\leftarrow$  1,N  
    OSSZ  $\leftarrow$  OSSZ+SQR(A[I]-ATL)  
SZ  $\leftarrow$  SQRT(OSSZ/N)
```



## Előfordulási statisztika

- **Feladat:** Adott egy  $N$  elemű adatsorozat. Gyűjtsük ki különböző elemeket és ezek darabszámát!





## Előfordulási statisztika

- **Megoldás:** Egy olyan táblázatot készítünk, amelynek első oszlopában a sorozatban előforduló különböző elemek értékei lesznek, a második oszlopában ezek előfordulási száma.

Funkció	Azonosító	Típus	Jelleg
A sorozat elemei	A	Egydimenziós, tetszőleges elemtípusú tömb	I
A sorozat elemeinek száma	N	Egész	I
A táblázat első oszlopa	T	Az A tömbbel megegyező típusú	M, O
A táblázat második oszlopa	DB	Egydimenziós egész tömb	M, O
A különböző elemek száma	K	Egész	M, O
A vizsgált sorozatelem indexe	I	Egész	M
Segédváltozó a kereséshez	J	Egész	M

## Előfordulási statisztika

```
/* Előfordulási statisztika */  
STATISZTIKA(A,N,T,DB,K)  
/* Kezdőérték */  
K  $\leftarrow$  0  
for I  $\leftarrow$  1,N  
    /* Felvétel */  
    K  $\leftarrow$  K+1  
    T[K]  $\leftarrow$  A[I]  
    DB[K]  $\leftarrow$  0  
    /* Keresés */  
    J  $\leftarrow$  1  
    while A[I]  $\neq$  T[J]  
        J  $\leftarrow$  J+1  
    /* Darabszám növelés */  
    DB[J]  $\leftarrow$  DB[J]+1  
    /* Ha volt már ilyen, töröljük a táblázat végéről */  
    if J < K  
        K  $\leftarrow$  K-1
```



## Jelstatisztika

- **Feladat:** Adott egy sztring. Készítsünk statisztikát a benne előforduló jelekről!



# Jelstatisztika

## ■ Típus

ELEM Rekord

/\* A gyakorisági táblázat egy eleme \*/

JEL

Karakter

DARAB

Egész

Funkció	Azonosító	Típus	Jelleg
A vizsgálandó jelsorozat	S	Sztring	I
A gyakorisági táblázat	T	Egydimenziós ELEM tömb[255]	O
A különböző jelek száma	K	Egész	O
A darabszámok tömbje	DB	Egydimenziós egész tömb[255]	M
A vizsgált jel kódja	KOD	Egész	M
Ciklusváltozó	I	Egész	M

## Jelstatisztika

```
/* Jelstatisztika */  
STATISZTIKA(S,T,K)  
/* Kezdőérték */  
for I  $\leftarrow$  0,255  
    DB[I]  $\leftarrow$  0  
/* Darabszámok meghatározása */  
for I  $\leftarrow$  1,LENGTH(S)  
    KOD  $\leftarrow$  ASC(S[I])  
    DB[KOD]  $\leftarrow$  DB[KOD]+1  
/* Eredménytáblázat elkészítése */  
K  $\leftarrow$  0  
for I  $\leftarrow$  0,255  
    if DB[I]>0  
        K  $\leftarrow$  K+1  
        T[K].JEL  $\leftarrow$  CHR(I)  
        T[K].DARAB  $\leftarrow$  DB[I]
```

