

A szoftverfejlesztésben **alprogramnak** nevezünk egy függvényt, eljárást, vagy műveletet azaz az alprogram olyan autonóm blokkok, amelyek miniatűr programokként funkcionálnak és a programon belül bárhol futtathatóak. Tágabb értelemben szinte bármilyen körülhatárolható programrészt hívhatunk alprogramnak: egy blokkot, modul, de akár egy elágazás egy ágát is.

Az alprogram fő részei formálisan **három részből** állnak. A **fej**, ami a specifikációt tartalmazza. A **törzs**, ami tetszőleges utasítások, tetszőleges számban és sorrendben. Természetesen figyelni kell az adott nyelv szintaktikai és szemantikai szabályaira, bár utóbbit általában nehezebb felderíteni. A **vég** ami általában külön alapszóval van megadva.

Tartalmilag **négy komponense** van.

A **név** - Ez egy azonosító, ezért az alprogramot is deklarálni kell.

A **formális** paraméterek - Vagy más néven formális paraméterlista. A procedurális absztrakciót teszi lehetővé. Az alprogramok egy problémaosztályt oldanak meg. A probléma akkor konkretizálható, amikor az alprogramot az aktuális paraméterek megadásával meghívjuk. A fejben található, általában kerek zárójelek között. A nyelvek egy része azt mondja, hogy nem kell zárójel, ha az alprogramnak nincs formális paramétere (Pascal), más része szerint pedig a zárójel nem a paraméterekhez, hanem az alprogramhoz tartozik, tehát paraméterek nélkül is ki kell tenni (C). Nyelvfüggő, hogy a paraméterek mivel vannak elválasztva. A paraméterlistán szereplő nevek a törzsben különféle objektumok lehetnek, változók, konstansok, nevesített konstansok, állomány nevek, más alprogramok azonosítói.

A korábbi nyelvekben a listán csak a paraméterek azonosítói szerepeltek, később a név mellett megjelent típusmegjelölés, de lehetnek plusz információk is, amelyek a formális paraméterek futás közbeni viselkedését szabályozzák. Számuk bármennyi lehet, a nulla paraméterrel rendelkező alprogramot paraméter nélküli alprogramnak is hívjuk. A törzsben deklarált objektumok és formális paraméterek az alprogram lokális objektumai.

A **törzs** - Az implementációt tartalmazza. Felépítése: [deklarációs utasítások] végrehajtó utasítások.

A **környezet** - A globális változók együttese.

Az alprogram minden esetben a bemenő adathalmazból előállít egy kimenő adathalmazt, végrehajt egy formális transzformációt.

Két fajtája van.

Eljárás (procedure): valamilyen tevékenységet végez, vagy pedig a paramétereit, illetve a környezetét változtatja meg.

Függvény (function): mindig egy értéket állít elő. Ezért van a függvénynek típusa, az eljárásnak pedig nincs.

A nyelvek általában külön alapszóval jelzik, hogy melyikről van szó. A függvényt majdnem minden nyelvben a function szóval vezetik be, míg az eljárást a procedure szóval.

Az **eljárásokat** mindig utasításszerűen lehet aktivizálni. Azaz ott lehet meghívni, ahol egy végrehajtható utasítást.

Hívó utasítás:

- vagy alapszóval (kód), pl. CALL,
- vagy nincs külön alapszó, csak az eljárás nevével és aktuális paramétereinek megadásával. Ez a nyelvekben egységesen kerek zárójelek között megadott aktuális és formális paraméterlistával történik.

Ha nincs paraméter (üres paraméterlista):

- vagy ki kell tenni a kerek zárójelet (C),
- vagy nem kell (Pascal).

Befejeződés:

- elfogynak az utasítások (ez szabályos befejeződés), ekkor a vezérlés visszaadódik a hívó utasításhoz,
- explicit utasítás létezik arra, hogy befejeződjön az eljárás. Ez bárhol lehet a törzsben. Kevés nyelv ismeri.
- Amelyik nyelvben van GOTO, megengedik, hogy visszaadjuk a vezérlést (de ez nem

javallott), nem szabályos).

- Minden nyelvben létezik olyan eszköz, amely befejezteti a programot.
Pl. PASCAL-ban a HALT, de ez nem utasítás, hanem eljárás.

Függvény:

Hívása: mindig kifejezésben hívjuk meg. Ahol kifejezés szerepel.

Alakja: függvénynév(aktuális paraméter).

Befejezés: hogyan fejeződik be, és mi lesz a visszaadott érték? A visszatérési értéket mindig a neve hordozza.

- A nyelvek egy részében a függvény nevét mint változót használhatjuk. A legutóbb kapott értékkel szerepel a függvény pl. a FORTRAN-ban.

- A függvény nevének értéket kell kapnia, ez a legutoljára kapott érték. (PASCAL.) Befejeződik, ha elérjük a végét.

- Külön utasítás van, s külön utasítás határozza meg a visszatérő értéket.

Pl. RETURN - s ez egyben be is fejezteti a kiértékelést. (Pl. C-ben.)

Generikus (template) alprogramok:

- Az alprogramok paraméterei nem lehetnek típusok, így nincs arra lehetőség hogy különböző típusokon megadott de egyébként hasonló feladatok megoldására egyetlen alprogramot írjunk.

- Erre a problémára kínál megoldást a típussal történő paraméterezhetőség.

- A típusparaméterrel ellátott alprogramokat polimorfnak, azaz többalakúnak vagy sablonnak nevezzük.

- Tulajdonképpen egy vezérlési absztrakció, azaz konkrét típusokra specializált alprogramok helyett típusfüggetlen általános megoldásokat adhatunk.

- Több alprogram megírása helyett egy típussal paraméterezhető alprogramot írunk:

A **korutinok** egymással párhuzamosan futó folyamatok, amelyek egymást aktivizálják. Amikor a vezérlés átadódik a következő korutinnak, annak a végrehajtása ott folytatódik, ahol az előző vezérlésátadáskor megszakadt.