Adatbázis és szoftverfejlesztés elmélet

Témakör 5.

Összefoglalás

1. Eljárások, függvények

Egy összetettebb programot nehéz megérteni az elejétől a végéig. Ezért fejlesztés során a feladatot több részfeladatra bontjuk, és ezeket egymástól függetlenül készítjük el. Így elegendő csak egy-egy részletre koncentrálnunk. Továbbá ha egy gyakori feladat megoldását elkészítettük, azt a program több pontján is fel tudjuk használni, így időt is megtakarítunk (ez a kód újrahasznosítása).

A szubrutinok

A szubrutinok (alprogramok) a program önálló részei, melyeknek a futás során át lehet adni a vezérlést. Ezt nevezzük meghívásnak. Ekkor a szubrutin utasításai hajtódnak végre, majd azután a program a rutin meghívását követő első utasításon folytatódik. Jellemzően a programban többször is ismétlődő feladatok végrehajtására használhatók. Például: hibaüzenetek kiírása, ismételt beolvasási műveletek stb.

Szubrutinok fajtái:

- Eljárás (procedure)
- Függvény (function)

A programban deklarált eljárások és függvények arra adnak lehetőséget, hogy a programunkat logikailag és funkcionálisan meghatározott, különálló egységekre bontsuk**.**

Az ***eljárás*** utasítások olyan sorozata, amelyik kiszámol valamilyen értéket vagy elvégez valami cselekvést. Az eljárás egy olyan szabályrendszer, amelyet végrehajtva, valamilyen eredményhez jutunk. Az algoritmus is egy ilyen sorozat, de egy algoritmus mindig befejeződik, míg egy eljárás nem feltétlenül.

A ***függvények*** pontosan úgy működnek, mint az *eljárások*, azzal a különbséggel, hogy **visszatérési értéket** is szolgáltatnak. A visszatérési értéket a függvény törzsében állíthatjuk be (pl. nyelvtől függően: függvénynév=érték, return érték vagy result=érték utasítással). A függvény deklarálásakor jelezni kell a visszatérési érték típusát.

Eljárások és függvények meghívása

A szubrutinokat a nevükkel indíthatjuk el a főprogramból, esetleg más eljárásokból vagy függvényekből. Természetesen csak akkor, ha a szubrutin deklarációja megelőzte az arra való hivatkozást.
Az eljárások meghívásának általános formája a következő:

eljárásnév(paraméterek)

A függvények esetében hasonlóan történik a hívás, azonban a függvény visszatérési értékének természetesen jelentősége van:

azonosító = függvénynév(paraméterek)

Amint az előbbi meghívási módoknál látható, a függvényeknek és eljárásoknak átadható egy vagy több paraméter. A paramétereket – sok programozási nyelv esetében - zárójelbe kell tenni a függvény vagy eljárás neve után. A formális paraméterlista, a változódeklarációhoz hasonlóan, a paraméterek azonosítóját és típusát tartalmazza. A paraméterek az szubrutinban helyi változókként viselkednek, azzal a különbséggel, hogy nem a rutinban belül kapnak értéket, hanem az eljárás hívásakor, kívülről. Az eljárásoknak és függvényeknek nem kötelező paramétert átadni, azaz a paraméterlista üres is lehet, ilyenkor paraméter nélküli eljárásról vagy függvényről beszélünk.

Paraméterátadás fajtái

- érték szerinti paraméterátadás
- cím szerinti paraméterátadás

Érték szerinti paraméterátadás

Ez az a paraméterátadási mód, amikor az alprogram törzsében a bemenő paraméter értékét használjuk. A formális paraméterlistában szereplő változó az alprogram lokális változója lesz, az eljárás vagy függvény hívásakor ebbe a változóba kerül az aktuális, meghíváskor megadott, paraméter értéke. Az aktuális értékparaméter lehet konstans, változó vagy kifejezés is. Leegyszerűsítve úgy is tekinthetjük, hogy a meghíváskor az értékek átmásolása történik a formális paraméterlista és az aktuális paraméterlista megfelelő elemei között. Ennél az átadási módnál csak ún. bemenő paraméterek lehetnek.

Cím szerinti paraméterátadás

Az a paraméterátadási mód, amikor a formális paraméterlistában szereplő változó nem az aktuális paraméter értékét kapja meg, hanem az aktuális paraméter memóriacíme lesz a formális paraméterhez rendelve. Ez azt jelenti, hogy ha az alprogram törzsében megváltozik az ilyen paraméter értéke, az megváltoztatja a hívó programrészben az aktuális paraméter változójának értékét is, mivel közös memóriaterületet használnak. Ezt egyszerűsítve úgy is mondhatjuk, hogy a szubrutin meghívásakor ugyan egy értékadás történik a formális paraméterlista és az aktuális paraméterlista megfelelő elemei között, azonban annak lefutása után ez az értékadás visszafelé is megtörténik. Ennél az átadási módnál lehetnek bemenő és kimenő paraméterek is.

1. Utasítások

Utasításfajták

- Egyszerű utasítás
- Összetett utasítás

Egyszerű utasítások

Az egyszerű utasítás olyan utasítás, aminek semelyik része sem tartalmaz másik utasítást. Ilyenek:

- az értékadó utasítás,
- az eljáráshívás,
- a kilépő utasítások,
- az állítások,
- és az üres utasítás.

Összetett utasítások

Az összetett utasítások más utasítások konstrukciójából állnak, amiket szekvenciálisan, feltételesen, vagy ismételve hajthatunk végre. A szekvenciális végrehajtásnál blokkokat hozhatunk létre, melyekben új változókat deklarálhatunk, ezáltal egy belső láthatósági kört teremtve. Létrehozhatunk blokkokat, amelyek deklarációk és utasítások egymáshoz rendelésére szolgálnak. Például egy változó láthatósági köre lehet egy blokk, egy elágazási ág vagy eset, vagy egy ciklustörzs.

Kérdések:

1. Mi indokolja, hogy egy összetettebb programot szubrutinokra osszunk fel?
2. Írja le a szubrutin meghatározását!
3. Mi az eljárás?
4. Mi a függvény?
5. Hogyan hívhatjuk meg a szubrutinokat? Adja meg az eljáráshívás és a függvényhívás általános formáját!
6. Mi a paraméterátadás két fajtája?
7. Ismertesse az érték szerinti paraméterátadást!
8. Ismertesse a cím szerinti paraméterátadást!
9. Sorolja fel az egyszerű utasításokat!
10. Mi jellemző az összetett utasításokra?