

Pásztor Attila

**Algoritmizálás és programozás
tankönyv
az emeltszintű érettségéhez**

3. ADATTÍPUSOK.....	26
3.1. AZ ADATOK LEGFONTOSABB JELLEMZŐI:	26
3.2. ELEM ADATTÍPUSOK	27
3.3. ÖSSZETETT ADATTÍPUSOK	28
3.4. A SPECIFIKÁCIÓ PONTOSÍTÁSA	29
3.5. ÖSSZEFOGLALÁS	30
3.6. KÉRDÉSEK, FELADATOK	30

3. Adattípusok

3.1. Az adatok legfontosabb jellemzői:

Azonosító:	A változóhoz rendelt név, így hivatkozhatunk rá.
Hozzáférési jog:	Az adat módosíthatósága és lekérdezhetősége.
Kezdőérték:	A deklarációval kap-e kezdőértéket, vagy sem.
Hatáskör:	A programszöveg azon része, ahol a hozzáférés megengedett. Globális: minden eljárásból elérhető. Lokális: Csak az őt közvetlenül tartalmazó eljárásban használható. Saját: Csak az adott eljárásban használható.
Láthatóság::	A program azon része, ahol lehetséges a hozzáférés (egy lokális változó eltakarhat egy külső változót.
Élettartam:	A futási idő azon része, ahol az azonosító ugyanazt az adatot jelenti.
Típus:	Az alábbi adattulajdonságok összessége: értékhalmaz, műveletek, ábrázolás.
Értékhalmaz:	Az a halmaz, amelyből a változó értéket vehet fel. Megkülönböztünk elemi és összetett típust.

Érték-halmaz szempontjából az adat lehet:

- Elemi típus: Szerkezetileg nem bontható tovább, például: egész, valós, logikai, karakter, mutató (adat címe), felsorolás, intervallum.
- Összetett típus: Szerkezettel rendelkezik, például: rekord (különböző típusú, de logikailag összetartozó típusok együttes kezelése), halmaz, szöveg (karakterek sorozata), sorozat, tömb (mátrix). A bonyolultabb adatszerkezetekkel (verem, sor, lista, fa, táblázat, gráf) ebben a könyvben nem foglalkozunk.

A típus meghatározza azt is, hogy milyen műveleteket értelmezhetünk rajta. Az elemi adattípusok lehetséges műveleteit 3 csoportba szokás sorolni:

- értékadó,
- típusátviteli:
 - relációs,
 - szelekciós (egy elem kiválasztásának módja),
 - konstrukciós (hogyan képezzük),
 - számasság (hány elemű az érték-halmaz),
- transzformációs (ugyanarra a típusra képez le, például +, -, min, előző, ...).

Az összetett adattípusok műveletei további csoportokat alkothatnak:

- lekérdező (egy-egy érték kiolvasása),
- bővítő (új adat felvétele),
- szűkítő (adat törlése),
- mozgató (kívánt adat kiválasztása),
- egyéb (hibakezelés, hány eleme van, ...).

A típus harmadik jellemzőjével, az adatok memóriában történő tárolásának módjával ebben a könyvben csak nagyon felületesen foglalkozunk.

Tekintsük át a középiskolás algoritmus és programozás feladatokban leggyakrabban használt adattípusokat!

3.2. Elemi adattípusok

Ebben a fejezetben áttekintjük az elemi adattípusok legfontosabb jellemzőit.

Egész

Értékhalmaza: ábrázolástól függ, például -32768 .. 32767
 Műveletek: értékadás, relációk, +, -, DIV, MOD, -, ^ (egész kit. hatvány)
 Ábrázolás: például 2 byte, 2-es komplement (erre vonatkozik a megadott értékhalmaza)
 Példa: Változó i: egész

Valós

Értékhalmaza: ábrázolástól függ, többféle értékhalmazzal szokás realizálni
 Műveletek: értékadás, relációk, +, -, *, /, -, ^
 Ábrázolás: például 6 byte, exponenciális normalizált alak
 Példa: Változó x: valós

Logikai

Értékhalmaza: igaz, hamis (true, false)
 Műveletek: nem (NOT), és (AND), vagy (OR), kizáró vagy (XOR), értékadás (:=)
 Ábrázolás: 1 biten, vagy (1 byte-on)
 Példa: Változó L: logikai

Karakter

Értékhalmaza: 0..255 kódú jelek halmaza
 Műveletek: értékadás, ord (hányadik a kódja szerint), chr (kódból karakter), relációk (kód alapján), + (konkatenáció, azaz összefűzés)
 Ábrázolás: általában 1 byte
 Példa: Változó c: karakter
 c:= 'a'

Felsorolás

Értékhalmaza: konstansok felsorolásával adjuk meg
 Műveletek: értékadás, relációk, ord (hányadik), pred (előző), succ (következő)
 Figyelem! Az előző és a következő nem ciklikusan értendő!
 Beolvasni, kiírni nem lehet! Szimulációra jó.
 Ábrázolás: sorszámot ábrázoljuk
 Példa: Konstans hétvége=(szombat,vasárnap)

Intervallum

Értékhalmaza: az alaptípus összefüggő részhalmaza
 Műveletek: megegyezik az alaptípus műveleteivel
 Példa: 1..8, 'A'..'Z'

Mutató

Értékhalmaza: nem negatív egész
 Műveletek: értékadás, new (lefoglal), dispose (felszabadít)
 Ábrázolás: például 4 byte

Egy memóriacím, ha nem definiált, akkor NIL értékű (ez nem azonos a nullával). Adott típusú változó tárolásához lehet memóriát foglalni, felszabadítani.

Példa: Típus mutató=[^]Karakter (karakterre mutat)

Változó p: mutató

Ez egyúttal példa arra is, hogy új típust hogyan adhatunk meg. A mutató típus byte-nyi adatra mutat. A p nevű változó egy ilyen mutató. Az adatterületet a new(p) utasítással lehet majd lefoglalni és a dispose(p) utasítással lehet felszabadítani.

3.3. Összetett adattípusok

Az értékhalmoz mellett a szerkezete (struktúrája) is lényeges, ábrázolásukat a legtöbb esetben már meg sem említem.

Rekord

Akkor használjuk, ha különböző típusú, de logikailag össze tartozó adatokat szeretnénk kezelni.

Értékhalmoz: alaptípus direktorzata (például: szöveg x egész)

Műveletek: értékadás, reláció (csak =, <>), kiválasztás (pont-tal jelöljük)

Ábrázolás: mezőnként folytonos memóriaterületen

Példa: Típus tanuló=Rekord

(név: szöveg;

kor: egész)

Változó: valaki: tanuló

valaki.név a tanuló nevét jelenti,

valaki.kor a tanuló korára hivatkozik.

Halmaz

Értékhalmoz: az alaptípus iteráltja, amely rendelkezik a halmaz tulajdonsággal (minden elem különböző, nincs sorrendiség)

Műveletek: értékadás, halmaz relációk, unio, metszet, különbség, elemek (in), üreshalmaz (ez hozza létre), jele: []

Példa: Típus napok=(hétfő,kedd, szerda, csütörtök, péntek, szombat, vasárnap)

Típus Halmaztípus= Halmaz(napok), vagy SET OF napok

Szöveg

Értékhalmoz: karakterek direktorzata, lehet fix hosszúságú, vagy változó.

Műveletek: értékadás, relációk (karakter kódra), + (konkatenáció), Jele(szöveg,i) a szöveg i-edik karaktere, melyet általában 1-től számolunk, esetleg szövegkezelő függvények

Példa: Változó s: szöveg

s := 'Ez egy példa.'

Sorozat

Értékhalmoz: alaptípus iteráltja és a lineáris rendezettség jellemzi: pontosan egy megelőző és következő létezik.

Műveletek: az alaptípussal megegyezik, kiválasztás
 Megadása: név(indextípus:értéktípus) alakban történik
 A sorozatot egydimenziós tömbbel valósítjuk meg.
 Példa: Változó s: sorozat(1..10:egész) NEM használatos, helyette:
 Változó s: tömb(1..10:egész), vagy tömb[1..10:egész]
 Ki: s(1)

Tömb

Értékhalmoz: alaptípus iteráltja és a lineáris rendezettség jellemzi: pontosan egy megelőző és következő létezik.
 Műveletek: az alaptípussal megegyezik, kiválasztás
 Megadása: név(indextípusok felsorolása:értéktípus) alakban történik
 Példa: Változó t: tömb(1..10,1..10:egész), vagy:
 Változó t: tömb[1..10,1..10:egész] ez egy egészezből álló 10x10-es mátrix.
 Ki: t(2,4)

3.4. A specifikáció pontosítása

A korábban tárgyalt két példa specifikációja és algoritmusai a változók megadásával, mondatos leírással:

```
Feladat: Számold ki a háromszög területét, ha adott az
egyik oldala és a hozzá tartozó magasság!
Bemenet: a (alap, valós szám), m (magasság, valós szám)
Kimenet: t (terület, valós szám)
Előfeltétel: a és m pozitív
Utófeltétel: t=a*m/2
```

```
Program Háromszög_terület:
Változó a,m: valós [pozitív]
  Be: a,m
  t:= a*m/2
  Ki: t
Program vége.
```

Az algoritmus „elegánsabb”, ha az adatok bekérésekor tudjuk, hogy mit is kell megadnunk. Az értékadó utasítás is elhagyható, hiszen az adatkiviteli utasítás tartalmazhat kifejezést:

```
Program Háromszög_terület:
Változó a,m: valós
  Ki: 'Alap:'; Be: a
  Ki: 'Magasság:'
  Be: m
  Ki: 'A háromszög területe:'
  Ki, a*m/2
Program vége.
```

A számkitaláló program specifikációja és algoritmusai:

```

Feladat:      Készíts algoritmust a következő számkitaláló játék
              ra! Gondolj egy egész számot 0 és 100 között, és
              jelöld Ezt g-vel. Aki ki akarja találni a számot, a
              következő módszer szerint jár el: A játékos mond
              egy számot (jelöljük ezt x-szel), te pedig megmon-
              dod, hogy a gondolt szám ennél nagyobb, vagy ki-
              sebb. Ezt mindaddig ismétlitek, amíg a kérdező ki-
              találja a gondolt számot!

Bemenet:      x [a tipp, egész]
Kimenet:      Szöveges üzenet, hogy a gondolt szám nagyobb, ki-
              sebb,          vagy egyenlő a tippel.
Előfeltétel:  x 0 és 100 közötti
Utófeltétel:  'A gondolt szám nagyobb.', ha g>x
              'A gondolt szám kisebb.', ha g<x
              'Kitaláltad a számot.', ha g=x

Változó:
  x,g: egész
  x:= véletlenszám [1 és 100 között]

Program Számkitaláló:
Ciklus
  Ki: 'Mondj egy számot!'
  Be: g
  Ha g>x akkor Ki: 'A gondolt szám nagyobb.'
  Ha g<x akkor Ki: 'A gondolt szám kisebb.'
  Ha g=x akkor Ki:
  amíg g<>x
  Ciklus vége
Program vége.

```

3.5. Összefoglalás

Ebben a fejezetben megismertük az adatok legfontosabb jellemzőit. A legfontosabb elemi és összetett típusok értékalmazát, műveleteit áttekintettük, valamint megismertünk alkalmazásukra néhány példát.

3.6. Kérdések, feladatok

- Keress ábrázolásokat az egész számok kettes komplementjére!
- Keress konkrét példákat exponenciális alakú normalizált valós szám ábrázolásra!
- Keress példákat, amikor egy adott értékalmazon megvalósított művelet kivezethet az értékalmazból!
- Ismételd át amit a komplement kódal ábrázolt számok összeadásáról, kivonásáról, szorzásáról általános iskolában tanultál!