**A programkészítés folyamata**

1. **Specifikálás** (miből?, mit?) → *specifikáció*
2. **Tervezés** (mivel?, hogyan?) → *adat- + algoritmus-leírás*
3. **Kódolás** (a gép hogyan?) → *kód* (reprezentáció + implementáció)
4. **Tesztelés** (hibás-e?) → *hibalista* (diagnózis)
5. **Hibakeresés** (hol a hiba?) → *hibahely*, -*ok*
6. **Hibajavítás** (hogyan jó?) → *helyes program*
7. **Minőségvizsgálat, hatékonyság** (jobbítható-e?, hogyan?) → *jó program*
8. **Dokumentálás** (hogyan működik, használható?) → *használható program*
9. **Használat, karbantartás** (még mindig jó?) → *évelő (időtálló) program*

**A specifikáció fogalma**

„Interface” a megbízó és a fejlesztő között

**Célja:**

a feladat formális megragadása.

**Összetevői:**

1. Bemenő adatok (azonosító, értékhalmaz [mértékegység])
2. Ismeretek a bemenetről (előfeltétel)
3. Eredmények (azonosító, értékhalmaz)
4. Az eredményt meghatározó állítás (utófeltétel)
5. A használt fogalmak definíciói
6. A megoldással szembeni követelmények
7. Korlátozó tényezők

[6-7.: Egyéb „technikai” körülmények (pl. megjelenítés, memóriakorlát, speciális adatszerkezet…)]

**Tulajdonságai:**

1. „Egyértelmű”, pontos, teljes
2. Rövid, tömör; formalizált
3. Szemléletes, érthető (fogalmak)

**Specifikációs eszközök:**

1. Szöveges leírás
2. Matematikai megadás

[T/1.: A specifikációtól nem a „belőle származó” algoritmus egyértelműségét kívánjuk meg, sőt: ebben igen nagy „szabadságfokot” kínál; a feladat legyen egyértelmű!

T/2-3.: inkább lehetőség, mint alaptulajdonság, nagyban függ az ember „leleményességétől”, fogalomalkotó képességétől, kreativitásától]

**Az algoritmus fogalma**

Az algoritmusok **összeállítási mód**jai:

* Szekvencia (egymás utáni végrehajtás)
* Elágazás (választás 2 vagy több tevékenységből)
* Ciklus (ismétlés adott darabszámszor vagy adott feltételtől függően)
* Alprogram (egy összetett tevékenység, egyedi néven – absztrakció)

[Alprogram : nevet adunk az összetett tevékenységnek (definiáljuk), hogy azután ahányszor kell elegendő legyen csak a nevével hivatkozni rá.]

**Példa: háromszög (specifikáció)**

**Feladat**:

*3 szám lehet-e egy derékszögű háromszög 3 oldala?*

**Specifikáció**:

* **Bemenet**: x,y,z∈R (R=Valós számok **halmaz**a)
* **Kimenet**: lehet∈L (L=Logikai értékek **halmaz**a)
* **Előfeltétel**: x>0 **és** y>0 **és** z>0
* **Utófeltétel**: lehet**=**(x2+y2=z2)

Megjegyzés: a 3 szám sorrendjét ezek szerint rögzítettük – z az átfogó hossza!

[A pirossal kiemeltek logikai operátorok, azaz értelmezési tartományuk: L×L.

A kékkel kiemeltek R–beli operátorok, azaz értelmezési tartományuk: R×R.

Mint látható ugyanaz a jel (=) kétféle operátorhoz is tartozhat. (Ez az ún. **polimorfizmus**.)

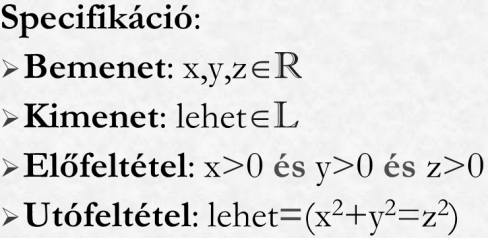
A halmazokhoz az **Imprint MT** fontot fogjuk használni.]

**Specifikáció = függvény**:

* **Bemenet**: x,y,z∈R (Független változók)  
  a függvény értelmezési tartománya: R×R×R=R3 (amelynek egyes komponenseire lehet hivatkozni a specifikációban x-szel, y-nal, z-vel)
* **Kimenet**: lehet∈L (Függő változó)  
  a függvény értékkészlete: L (amelyre hivatkozhatunk a specifikációban lehet-tel)
* **Előfeltétel**: x>0 **és** y>0 **és** z>0  
  a függvény értelmezési tartományának (R3) szűkítése (R+3 )
* **Utófeltétel**: lehet**=**(x2+y2=z2)  
  mi igaz a végeredményre

**Példa: háromszög (algoritmus)**

**Algoritmus**:

A programunk 4 fő részből áll:  
az adatok deklarálása, beolvasása, az   
eredmény kiszámítása, az eredmény   
kiírása:

|  |
| --- |
| Be: x, y, z [x > 0 és y > 0 és z > 0] *(előfeltétel) (x,y,z valós változók, valós számok típusa)* |
| lehet := (x2+y2=z2) *(lehet logikai változó, logikai értékek típusa)* |
| Ki: lehet |

A deklarációt, az „elemi” utasításokat egy-egy „dobozba” írjuk.

Később a be- és kimenetet nem algoritmizáljuk!

[A „lehet:=(x2+y2=z2)” utasítással ekvivalens, csak nála jóval hosszabb   
a „**Ha** x2+y2=z2 **akkor** lehet:=Igaz **különben** lehet:=Hamis” utasítással.]

Egy másik **algoritmus** a lényegi részre:

|  |
| --- |
| xx := x2 |
| yy := y2 |
| zz := z2 |
| lehet := (xx+yy=zz) |

xx, yy, zz valós változók

Bevezethetők/-endők segéd (belső, saját) változók.

[Megelégszünk a specifikációban nem szereplő adatok deklarálásával, hiszen ezek bukkantak föl hirtelen, a „semmiből”.]

**Példa: másodfokú egyenlet (specifikáció)**

**Feladat**:

*Adjuk meg a másodfokú egyenlet egy megoldását! Az egyenlet:* ax2+bx+c=0

**Kérdések**:

* + Mitől függ a megoldás? – *bemenet*
  + Mi a megoldás? – *kimenet*
  + Mit jelent: „megoldásnak lenni”? – *utófeltétel*
  + Mindig/Mikor *van* megoldás? – *előfeltétel*
  + Biztos *egy* megoldás van? – *kimenet/utófeltétel*

**Specifikáció1:**

* Bemenet: a,b,c∈R
* Kimenet: x∈R
* Előfeltétel: –
* Utófeltétel1: ax2+bx+c=0

Megjegyzés: az uf. nem ad algoritmizálható információt. Nem baj, sőt tipikus, de … próbálkozzunk még!

Megoldóképlet: 

**Specifikáció2**:

* Bemenet: a,b,c∈R
* Kimenet: x∈R
* Előfeltétel: a≠0
* Utófeltétel2: 

**Nyitott kérdések:**

* *Mindig/Mikor van megoldás?*
* *Egy megoldás van?*

**Kimenet bővítés:**

* Kimenet: x∈R,van∈L [A „feladat-függvény” értékkészlete: R×L]
* Utófeltétel:   
  van=(b2–4\*a\*c≥0) és  
  van

**Nyitott kérdés:**

* *Egy megoldás van? – hf .*

**Példa: másodfokú egyenlet (algoritmus)**

**Algoritmus**:

A **feltételes** utasítás „3-dobozos” struktúra.