

A black and white aerial photograph of Budapest, Hungary, showing the Danube River and the city skyline. The image is used as a background for the title slide.

Programozás

1. előadás

Tartalom

- A problémamegoldás lépései – a programkészítés folyamata
- A specifikáció
- Az algoritmus
- Algoritmikus nyelvek – struktogram
- Adatokkal kapcsolatos fogalmak
- Adattípusok, elemi adattípusok
- Elemi feladatok – elágazások
- Rekordok
- Kódolás – elágazások



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



A specifikáció fogalma

Tulajdonságai:

1. „Egyértelmű”, pontos, teljes
2. Tömör (\leftarrow formalizált)
3. Érthető, szemléletes (fogalmak)

A három szempont sokszor ellentmond egymásnak.

Specifikációs eszközök:

1. Szöveges leírás
2. Matematikai megadás



Az algoritmus fogalma

Elemi tevékenységek:

értékkadás, beolvasás, kiírás.

Az algoritmusok **összeállítási módjai**:

- 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)



Példa: háromszög

(specifikáció)

Feladat:

3 szám lehet-e egy derékszögű háromszög 3 oldala?

Specifikáció:

➤ **Bemenet:** $x, y, z \in \mathbb{R}$

\mathbb{R} = Valós számok **halmaza**

➤ **Kimenet:** $\text{lehet} \in \mathbb{L}$

\mathbb{L} = Logikai értékek **halmaza**

➤ **Előfeltétel:** $x > 0$ és $y > 0$ és $z > 0$

➤ **Utófeltétel:** $\text{lehet} = (x^2 + y^2 = z^2)$

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



Példa: háromszög

(specifikáció)

Feladat:

3 szám lehet-e egy derékszögű háromszög 3 oldala?

Specifikáció₂:

- **Bemenet:** $x, y, z \in \mathbb{R}$
- **Kimenet:** $\text{lehet} \in \mathbb{L}$
- **Előfeltétel:** $0 < x \leq y \leq z$
- **Utófeltétel:** $\text{lehet} = (x^2 + y^2 = z^2)$

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



Háromszög: $\mathbb{R}^3 \rightarrow \mathbb{L}$

Háromszög(x, y, z):=lehet és

$$\text{lehet} = (x^2 + y^2 = z^2)$$

Specifikáció = függvény:

Független változók

➤ **Bemenet:** $x, y, z \in \mathbb{R}$

a függvény értelmezési tartománya: $\mathbb{R} \times \mathbb{R} \times \mathbb{R} = \mathbb{R}^3$ (amelynek egyes komponenseire lehet hivatkozni a specifikációban x -szel, y -nal, z -vel)

➤ **Kimenet:** $\text{lehet} \in \mathbb{L}$

Függő változó

a függvény értékkészlete: \mathbb{L} (amelyre hivatkozhatunk a specifikációban lehet-tel)

➤ **Előfeltétel:** $x > 0$ és $y > 0$ és $z > 0$

a függvény értelmezési tartományának (\mathbb{R}^3) szűkítése (\mathbb{R}_+^3)

➤ **Utófeltétel:** $\text{lehet} = (x^2 + y^2 = z^2)$

mi igaz a végeredményre: a „kiszámítási szabály”

Háromszög: $\mathbb{R}^3 \rightarrow \mathbb{L}$

Háromszög(x, y, z):= $(x^2 + y^2 = z^2)$



Specifikáció és megvalósítás



Specifikáció és megvalósítás:

A feladat specifikációja valós világbeli objektumokhoz rendel valamilyen valós világbeli eredményt. Emiatt valós világbeli dolgokkal (pontosabban azok absztrakciójával, pl. valós számok halmaza) foglalkozik.

A feladat számítógépes megoldása emiatt több részből áll

- a valós világbeli objektumokat leíró adatokat be kell juttatni a számítógépbe, annak memóriájában tárolni kell – ezek lesznek a megoldásbeli változók, amelyek típusa a számítógépes világ által elfogadott/megvalósított típusokból állhat (azaz pl. a számítógépes valós számok halmaza a matematika valós számhalmazának egy véges része lehet) – a specifikációban szereplő neveket (egyelőre) azonos nevű memóriabeli változókkal azonosítjuk;



Specifikáció és megvalósítás

- a memóriában megjelenő változókból valamilyen függvénnyel kiszámítjuk az eredményt, amit szintén a memóriában tároljuk – ezek neve (egyelőre) szintén megegyezik a specifikációban szereplő elnevezésekkel;
- végül az eredményt tartalmazó változók értékeit valahogyan kijuttatjuk a külvilágba.

Megjegyzés: lehetnek olyan változók is (látni fogjuk), amelyek a specifikációban nem jelennek meg.

Ebből alakul ki a klasszikus programok három fő lépése (=3 algoritmus szekvenciája):

- az adatok beolvasása;
- az eredmény kiszámítása;
- az eredmény kiírása.



Példa: háromszög (algorithmus)

Algorithmus₁:

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

Valós: Valós számok **típusa**
Logikai: Logikai értékek **típusa**

Specifikáció:

- **Bemenet:** $x, y, z \in \mathbb{R}$
- **Kimenet:** $\text{lehet} \in \mathbb{L}$
- **Előfeltétel:** $x > 0$ és $y > 0$ és $z > 0$
- **Utófeltétel:** $\text{lehet} = (x^2 + y^2 = z^2)$

Be: $x, y, z \rightarrow [x > 0 \text{ és } y > 0 \text{ és } z > 0]$

$\text{lehet} := (x^2 + y^2 = z^2)$

Ki: lehet

Változó

x, y, z : **Valós**

lehet: **Logikai**

A **deklarációt**, az „elemi” utasításokat egy-egy „dobozba” írjuk. Később a be- és kimenetet nem algoritmizáljuk!



Példa: háromszög (algorithmus)

Algorithmus₂:

A programunk 4 fő részből áll: az adatok **deklarálása**, **beolvasása**, az eredmény ki**számítása**, az eredmény **kiírása**:

Specifikáció:

- **Bemenet:** $x, y, z \in \mathbb{R}$
- **Kimenet:** $\text{lehet} \in \mathbb{L}$
- **Előfeltétel:** $x > 0$ és $y > 0$ és $z > 0$
- **Utófeltétel:** $\text{lehet} = (x^2 + y^2 = z^2)$

Be: x, y, z [$x > 0$ és $y > 0$ és $z > 0$]	
I \	/ N ($x^2 + y^2 = z^2$)
lehet := Igaz	lehet := Hamis
Ki: lehet	

Változó

x, y, z : **Valós**

lehet: **Logikai**

A **deklarációt**, az „elemi” utasításokat egy-egy „dobozba” írjuk. Később a be- és kimenetet nem algoritmizáljuk!



Példa: háromszög (algorithmus)

Egy másik **algorithmus** a lényegi részre:

Segéd változók deklarálása
„széljegyzetként”

Változó

xx,yy,zz:Valós

xx:=x²

yy:=y²

zz:=z²

lehet:=(xx+yy=zz)

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



Példa: másodfokú egyenlet (specifikáció)



Feladat:

Adjuk meg a másodfokú egyenlet egy megoldását!

Az egyenlet: $ax^2+bx+c=0$.

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*



Példa: másodfokú egyenlet (specifikáció)

Specifikáció₁:

- Bemenet: $a, b, c \in \mathbb{R}$
- Kimenet: $x \in \mathbb{R}$
- Előfeltétel: –
- Utófeltétel₁: $ax^2 + bx + c = 0$

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

Megoldóképlet:
$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4 * a * c}}{2 * a}$$

Feladat:

Adjuk meg a másodfokú egyenlet egy megoldását! Az egyenlet: $ax^2 + bx + c = 0$

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*



Példa: másodfokú egyenlet (specifikáció)

Specifikáció₂:

- Bemenet: $a, b, c \in \mathbb{R}$
- Kimenet: $x \in \mathbb{R}$
- Előfeltétel: $a \neq 0$
- Utófeltétel₂: $x = \frac{-b + \sqrt{b^2 - 4 * a * c}}{2 * a}$

Nyitott kérdések:

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



Példa: másodfokú egyenlet (specifikáció)

Kimenetbővítés:

- Kimenet: $x \in \mathbb{R}$, $van \in \mathbb{L}$
- Utófeltétel: $van = (b^2 - 4 * a * c \geq 0)$ és

$$van \rightarrow x = \frac{-b + \sqrt{b^2 - 4 * a * c}}{2 * a}$$

A „feladat-függvény”
értékkészlete: $\mathbb{R} \times \mathbb{L}$

Nyitott kérdés:

- *Egy megoldás van? – hf.*



Példa: másodfokú egyenlet (algoritmus)

Algoritmus:

Specifikáció₂:

- Bemenet: $a, b, c \in \mathbb{R}$
- Előfeltétel: $a \neq 0$
- Kimenet: $x \in \mathbb{R}$, $\text{van} \in \mathbb{L}$
- Utófeltétel: $\text{van} = (b^2 - 4 * a * c \geq 0)$ és

$$\text{van} \rightarrow x = \frac{-b + \sqrt{b^2 - 4 * a * c}}{2 * a}$$

Be: a, b, c [$a \neq 0$]

d: $b^2 - 4 * a * c$

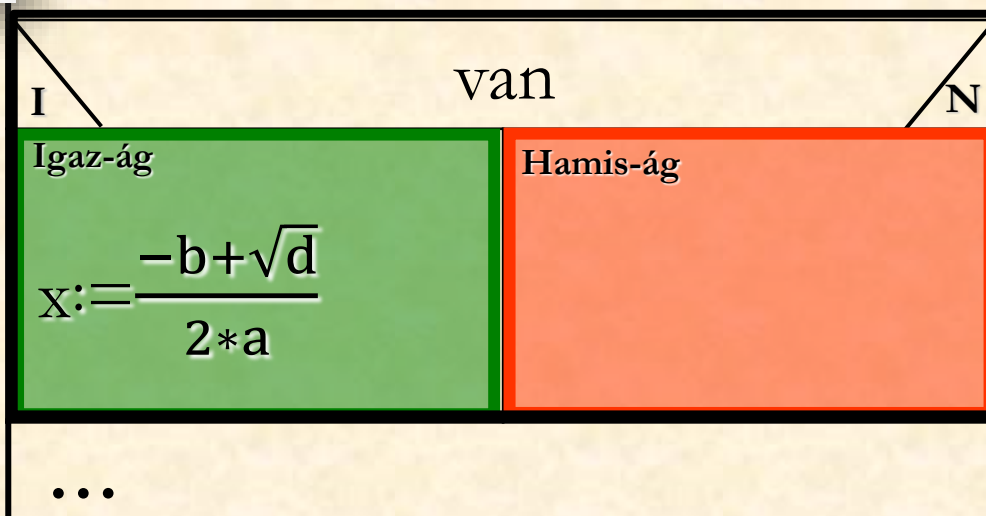
van: $d \geq 0$

Változó

a, b, c, x : **Valós**

van : **Logikai**

d : **Valós**



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



Példa: másodfokú egyenlet (algoritmus)

Algoritmus másképpen:

Program MásodfokúEgyenlet:

Változó

a, b, c, x : **Valós**

van : **Logikai**

d : **Valós**

Be: a, b, c [$a \neq 0$]

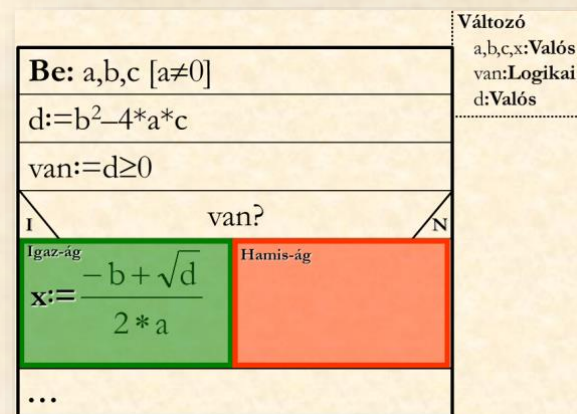
$d := b^2 - 4 * a * c$

$\text{van} := d \geq 0$

Ha van **akkor**

...

Program vége.



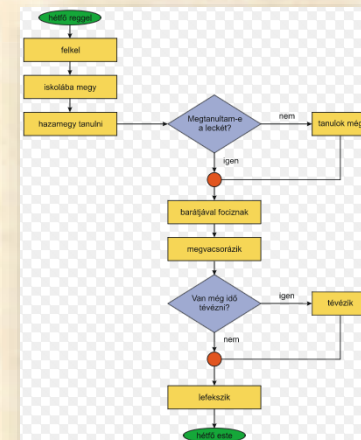
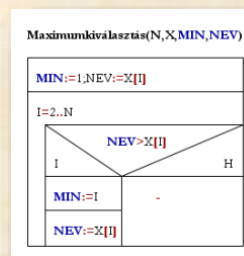
Algoritmusleíró nyelvek

➤ Szöveges leírás

- Mondatokkal leírás
- Mondatszerű elemekkel – **pszeudokód**

➤ Rajzos leírás

- Folyamatábra
- Struktogram



Struktogram (és pseudokód)

➤ Szekvencia:



Utasítás1
Utasítás2

➤ Elágazások:



Ha Feltétel **akkor**
Igaz-ág utasításai
különben
Hamis-ág utasításai
Elágazás vége



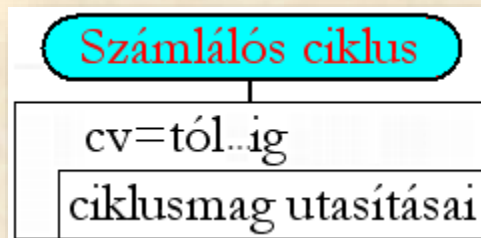
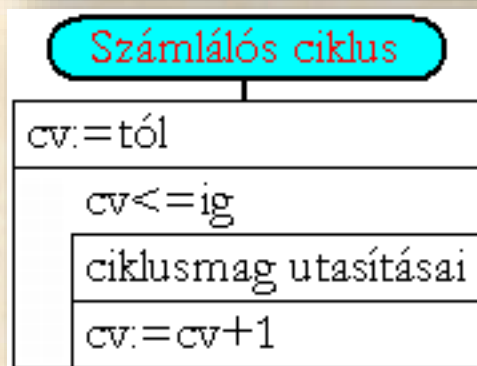
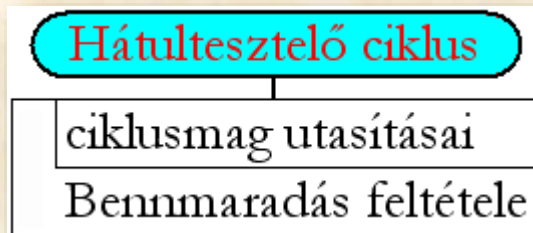
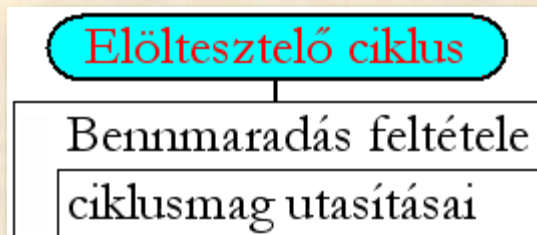
Elágazás
Feltétel1 **esetén** Utasítások1
Feltétel2 **esetén** Utasítások2
...
egyéb esetekben Utasítások
Elágazás vége



Struktogram

(és pseudokód)

➤ Ciklusok:



Ciklus amíg Feltétel
 ciklusmag utasításai
Ciklus vége

Ciklus
 ciklusmag utasításai
amíg Feltétel
Ciklus vége

Ciklus cv=tól-tól ig-ig
 ciklusmag utasításai
Ciklus vége

➤ Struktogramszerkesztés:

- Táblázatkezelővel/szövegszerkesztővel
- Célprogramokkal



Adatokkal kapcsolatos fogalmak

➤ Konstans

Az az adat, amely a műveletvégzés során nem változtathatja meg értékét, mindvégig ugyanabban az „állapotban” marad.

➤ Változó

Az ilyen adatféleségnek lényegéhez tartozik a „változékonyság”, más szóval: vonatkozhatnak rá olyan műveletek is, amelyek új értékkel látják el. Tudományosan fogalmazva: végrehajtás során megváltozhat az állapothalmaza.



Adatokkal kapcsolatos fogalmak

➤ Változók fajtái céljuk szerint

- bemeneti változó: bemenetkor kap értéket
- eredmény: kiszámítás tartozik hozzá
- részeredmény: kiszámítás tartozik hozzá, belőle további kiszámítások indulnak
- ... *(lesznek még továbbiak)*



Adatokkal kapcsolatos fogalmak

➤ Értékadás

Az az utasítás, amely révén a pillanatnyi állapotból egy meghatározott állapotba kerül a változó. (Nyilvánvaló, hogy konstans adatra nem vonatkozhat értékadás, az egy, kezdő-értéket meghatározón kívül.)

➤ Típus

Olyan „megállapodás” (absztrakt kategória), amely adatok egy lehetséges körét jelöli ki az által, hogy rögzíti azok állapothalmazát és az elvégezhető műveletek készletét.



Az adatjellemzők összefoglalása

Azonosító

Az a jelsorozatok, amelyekkel hivatkozhatunk a tartalmára, amely által módosíthatjuk tartalmát.

Kezdőérték

A születéskor hozzárendelt érték.

Konstansoknál nyilvánvaló, hogy deklarációban kapja; változóknál akár deklarációban, akár futáskor szerezt értéket magának.



A típus

Összetettség (strukturáltság) szempontjából beszélhetünk

- strukturálatlan (vagy skalár, elemi) típusról, ha (az adott szinten) szerkezetet nem tulajdonítunk neki; vagy
- strukturált (más szóval: összetett) típusról, ha (elemibb) összetevőkre bontjuk.



Elemi típusok

Egész típus

- Értékhalma: $-2^{31} \dots +2^{31} - 1$
(Min'Egész..Max'Egész)
- Műveletek: $+$, $-$, $*$, **Div** (egészosztás), **Mod** (osztási maradék), $-$ (unáris mínusz), $^$ (pozitív egész kitevős hatványozás)
- Relációk: $=$, $<$, \leq , \neq , \geq , $>$
- Ábrázolás: kettes komplement kódú
- Változatai: méret és előjel szerint sokfélék

Példaként: 4-bájtos
ábrázolást feltételezve.

A beolvasáson, a kiíráson
és értékadáson túliakkal
foglalkozunk csak.

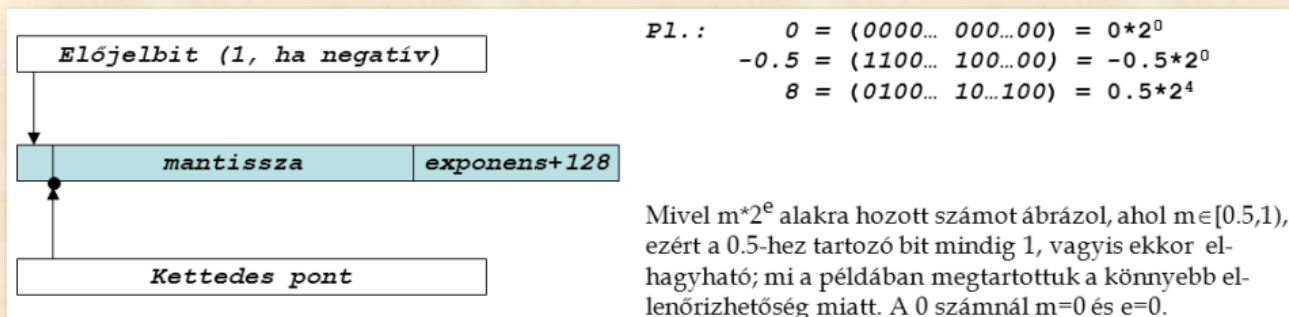
Pl. 3-bites 2-es komplement kódú egész számok:
 $+0=0|00_2$, $+1=0|01_2$, $+2=0|10_2$, $+3=0|11_2$,
 $-1=1|11_2$, $-2=1|10_2$, $-3=1|01_2$, $-4=1|00_2$,
Vegye észre a „szabályszerűségeket”!



Elemi típusok

Valós típus

- Értékhalmoz: ?????..????
(Min'Valós..Max'Valós nem definiáltak, vagy reprezentáció-függőek)
- Műveletek: +, -, *, /, ^, - (unáris mínusz)
- Relációk: =, <, ≤, ≠, ≥, >
- Ábrázolás: lebegőpontos ábrázolás (pontosabb lenne, ha e típust racionálisnak neveznénk, mert csak racionális számot képes ábrázolni)



Elemi típusok

Logikai típus

- Értékhalma: Hamis..Igaz
(Min'Logikai..Max'Logikai: Hamis, illetve Igaz)
- Műveletek: nem, és, vagy (a szokásos logikai műveletek)
- Relációk: $=$, $<$, \leq , \neq , \geq , $>$
- Ábrázolás: $0B = \text{Hamis}$, $-1B = \text{Igaz}$
(esetleg: $1B = \text{Igaz}$)... ahol $xB = x$ érték „bináris egészként” ábrázolva
- Megjegyzés: a rendezésnek nem nagy a gyakorlati jelentősége.



Elemi típusok

Karakter típus

- Értékhalma: 0..255 - kódú jelek – ASCII
(Min'Karakter..Max'Karakter: a 0, illetve a 255 kódú karakter)
- Műveletek: karakter-specifikus nincs
(esetleg a Kód:Karakter→Egész függvény, és inverze a Karakter:Egész→Karakter függvény, amelyek a belső ábrázolással hozza kapcsolatba)
- Relációk: =, <, ≤, ≠, ≥, >
(a belső ábrázolásuk alapján → nem ABC-sorrend!)



Feladatok elágazásra: vércsoport – 1



Feladat:

Egy ember vércsoportját (Rh negatív vagy pozitív) egy génpár határozza meg. Mindkét gén lehet „+” vagy „-” típusú. A „++” és a „+-” típusúak az „Rh pozitívok”, a „--” típusúak pedig az „Rh negatívok”.

Írj programot, amely megadja egy ember vércsoportját a génpárja ismeretében!



Egy ember vércsoportját (Rh negatív vagy pozitív) egy génpár határozza meg. Mindkét gén lehet „+” vagy „-” típusú. A „++” és a „+-” típusúak az „Rh pozitívok”, a „--” típusúak pedig „Rh negatívok”.

Elágazásra: vércsoport – 1

Specifikáció:

- Bemenet: $x, y \in K$
- Kimenet: $v \in S$
- Előfeltétel: $x, y \in \{ "+", "-" \}$
- Utófeltétel: $(x = "+" \text{ vagy } y = "+") \text{ és } v = "Rh+" \text{ vagy } (x = "-" \text{ és } y = "-") \text{ és } v = "Rh-"$

K = Karakterek halmaza

S = Karakter-sorozatok
(szövegek) halmaza

Algoritmus:

Innentől kezdve el-
hagyjuk a változók
deklarálását, a
beolvasást és a
kiírást

I	x = "+" vagy y = "+"		N
	v := "Rh+"	v := "Rh-"	



Feladatok elágazásra: vércsoport – 1

Specifikáció:

- Bemenet: $x, y \in K$
- Kimenet: $v \in S$
- Előfeltétel: $x, y \in \{ "+", "-"\}$
- Utófeltétel: $(x = "+" \text{ vagy } y = "+") \rightarrow v = \text{"Rh+"}$ és $(x = "-" \text{ és } y = "-") \rightarrow v = \text{"Rh-"}$

Egy ember vércsoportját (Rh negatív vagy pozitív) egy génpár határozza meg. Mindkét gén lehet „+” vagy „-” típusú. A „++” és a „+-” típusúak az „Rh pozitívok”, a „--” típusúak pedig „Rh negatívok”.

Algoritmus:

I	x="+" vagy y="+"		N
v:="Rh+"		v:="Rh-"	

Elolvasandó a jegyzet!

Feladatok elágazásra: vércsoport – 2



Feladat:

*Egy ember vércsoportját (A , B , AB vagy O) egy génpár határozza meg. Mindkét gén lehet ***a***, ***b*** vagy ***O*** típusú.*

A vércsoport meghatározása: $A = \{aa, aO, Oa\}$; $B = \{bb, bO, Ob\}$; $AB = \{ab, ba\}$; $O = \{OO\}$.

Írj programot, amely megadja egy ember vércsoportját a génpárja ismeretében!



Feladatok elágazásra: vércsoport – 2



Specifikáció:

- Bemenet: $x, y \in K$
- Kimenet: $v \in S$
- Előfeltétel: $x, y \in \{ "a", "b", "0" \}$
- Utófeltétel:

$(x = "a" \text{ és } y \neq "b" \text{ vagy } x \neq "b" \text{ és } y = "a")$

$\text{és } v = "A" \text{ vagy}$

$(x = "b" \text{ és } y \neq "a" \text{ vagy } x \neq "a" \text{ és } y = "b")$

$\text{és } v = "B" \text{ vagy}$

$(x = "a" \text{ és } y = "b" \text{ vagy } x = "b" \text{ és } y = "a")$

$\text{és } v = "AB" \text{ vagy}$

$x = "0" \text{ és } y = "0" \text{ és } v = "0"$

Egy ember vércsoportját (A, B, AB vagy 0) egy gén-pár határozza meg. Mindkét gén lehet a, b vagy 0 típusú.

A vércsoport meghatározása: $A = \{aa, a0, 0a\}$;
 $B = \{bb, b0, 0b\}$; $AB = \{ab, ba\}$; $0 = \{00\}$.



Feladatok elágazásra: vércsoport – 2



Algoritmus₁:

Kétirányú
elágazások
egymásba
ágyazásával.

I \ $x="a"$ és $y \neq "b"$ vagy $x \neq "b"$ és $y="a"$		N
v:="A"	I \ $x="b"$ és $y \neq "a"$ vagy $x \neq "a"$ és $y="b"$	N
	v:="B"	N
		N
	I \ $x="a"$ és $y="b"$ vagy $x="b"$ és $y="a"$	N
	v:="AB"	v:="0"

Specifikáció:

- Bemenet: $x, y \in K$
- Kimenet: $v \in S$
- Előfeltétel: $x, y \in \{"a", "b", "0"\}$
- Utófeltétel:
 - $(x="a" \text{ és } y \neq "b" \text{ vagy } x \neq "b" \text{ és } y="a")$
és $v="A"$ **vagy**
 - $(x="b" \text{ és } y \neq "a" \text{ vagy } x \neq "a" \text{ és } y="b")$
és $v="B"$ **vagy**
 - $(x="a" \text{ és } y="b" \text{ vagy } x="b" \text{ és } y="a")$
és $v="AB"$ **vagy**
 - $x="0" \text{ és } y="0" \text{ és } v="0"$



Feladatok elágazásra: vércsoport – 2



Algoritmus₂:

Sokirányú
elágazással.

$x = "a"$ és $y \neq "b"$ vagy $x \neq "b"$ és $y = "a"$	$x = "b"$ és $y \neq "a"$ vagy $x \neq "a"$ és $y = "b"$	$x = "a"$ és $y = "b"$ vagy $x = "b"$ és $y = "a"$	$x = "0"$ és $y = "0"$
$v := "A"$	$v := "B"$	$v := "AB"$	$v := "0"$

Specifikáció:

- Bemenet: $x, y \in K$
- Kimenet: $v \in S$
- Előfeltétel: $x, y \in \{ "a", "b", "0" \}$
- Utófeltétel:
 - $(x = "a" \text{ és } y \neq "b" \text{ vagy } x \neq "b" \text{ és } y = "a")$
és $v = "A"$ **vagy**
 - $(x = "b" \text{ és } y \neq "a" \text{ vagy } x \neq "a" \text{ és } y = "b")$
és $v = "B"$ **vagy**
 - $(x = "a" \text{ és } y = "b" \text{ vagy } x = "b" \text{ és } y = "a")$
és $v = "AB"$ **vagy**
 - $x = "0" \text{ és } y = "0" \text{ és } v = "0"$



Feladatok elágazásra: vércsoport –

Lokális változók
deklarálása

Algoritmus₃:

Segédváltozók
bevezetésével.

vana:= x="a" vagy y="a"			
vanb:= x="b" vagy y="b"			
I	vana		N
I	vanb	N	I
			N
v:="AB"		v:="A"	v:="B"
		v:="0"	

Változó
vana,
vanb:Logikai

Specifikáció:

- Bemenet: $x, y \in K$
- Kimenet: $v \in S$
- Előfeltétel: $x, y \in \{ "a", "b", "0" \}$
- Utófeltétel:
 - $(x="a" \text{ és } y \neq "b" \text{ vagy } x \neq "b" \text{ és } y="a")$
és $v="A"$ **vagy**
 - $(x="b" \text{ és } y \neq "a" \text{ vagy } x \neq "a" \text{ és } y="b")$
és $v="B"$ **vagy**
 - $(x="a" \text{ és } y="b" \text{ vagy } x="b" \text{ és } y="a")$
és $v="AB"$ **vagy**
 - $x="0" \text{ és } y="0" \text{ és } v="0"$



Rekordok/Struktúrák

Feladat:

Adjuk meg, hogy egy P síkbeli pont melyik síknegyedbe esik!

Megoldás felé – specifikáció:

A síkbeli pontokat ($\mathbf{R} \times \mathbf{R}$, \times :direktszorzás művelet) x - és y -koordinátájukkal adjuk meg.

Ezt a specifikációban így írjuk le:

$P \in \text{Pont}$, $\text{Pont} = \mathbf{X} \times \mathbf{Y}$, $\mathbf{X}, \mathbf{Y} = \mathbf{R}$

A **nevük**kel hivatkozhatunk rájuk a specifikációban:

$P = (P.\mathbf{x}, P.\mathbf{y})$.

Tehát: $P.\mathbf{x} \in \mathbf{X}$, $P.\mathbf{y} \in \mathbf{Y}$



Rekordok/Struktúrák

Feladat:

Adjuk meg, hogy egy P síkbeli pont melyik síknegyedbe esik!

Specifikáció:

- Bemenet: $P \in \text{Pont}$, $\text{Pont} = X \times Y$, $X, Y = \mathbb{R}$
- Kimenet: $SN \in \mathbb{N}$
- Előfeltétel: –
- Utófeltétel: $P.x \geq 0$ és $P.y \geq 0 \rightarrow SN=1$ és
 $P.x < 0$ és $P.y \geq 0 \rightarrow SN=2$ és
 $P.x < 0$ és $P.y < 0 \rightarrow SN=3$ és
 $P.x \geq 0$ és $P.y < 0 \rightarrow SN=4$

Új halmaz definíciója



Rekordok/Struktúrák

Specifikáció \rightarrow algoritmus_{adatleírás}:

➤ $P \in \text{Pont}$, $\text{Pont} = X \times Y$, $X, Y = R$

Típus $\text{TPont} = \text{Rekord}(x, y: \text{Valós})$

Változó $P: \text{TPont}$

Tehát a **TPont** egy új **adattípus**.

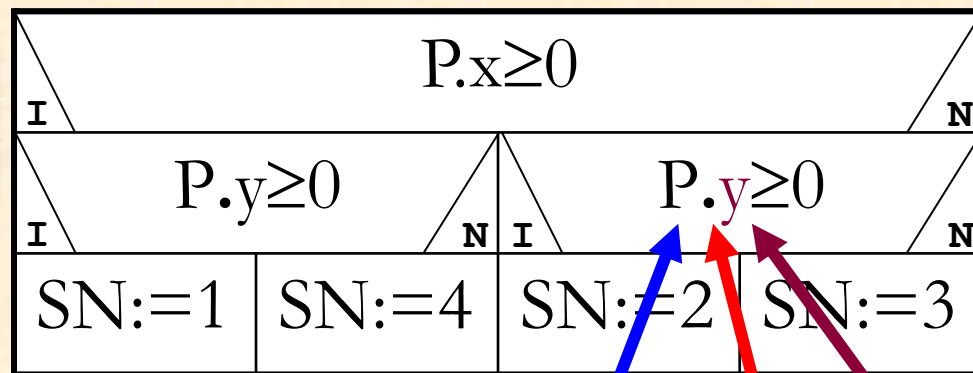
➤ A rekordok **összetett** adatszerkezetek, a részeikre „**nevük**” által meghatározott **szelektorok**kal hivatkozunk:
 $P = (P.x, P.y)$.



Rekordok/Struktúrák

Specifikáció \rightarrow algoritmus_{tevékenység}:

➤ Utófeltétel: $P.x \geq 0$ és $P.y \geq 0 \rightarrow SN=1$ és
 $P.x < 0$ és $P.y \geq 0 \rightarrow SN=2$ és
 $P.x < 0$ és $P.y < 0 \rightarrow SN=3$ és
 $P.x \geq 0$ és $P.y < 0 \rightarrow SN=4$



→ → →
 adatazonosító mezőazonosító



Típusdefiniálás

Rekordok/Struktúrák

Típusdefiníció



Specifikáció → algoritmus → kód:

➤ Típus

TPont=Rekord(x,y:Egész)

C# típusdefiníció:

```
struct TPont  
{  
    public int x,y;  
}
```

típusazonosító

hozzáférési jog

mezőtípus

mezőazonosítók



Típusdeklarálás

Rekordok/Struktúrák

Típusdeklaráció



Specifikáció → algoritmus → kód:

➤ Változó

P:TPont

C# típusdeklaráció:

TPont P ← adatazonosító

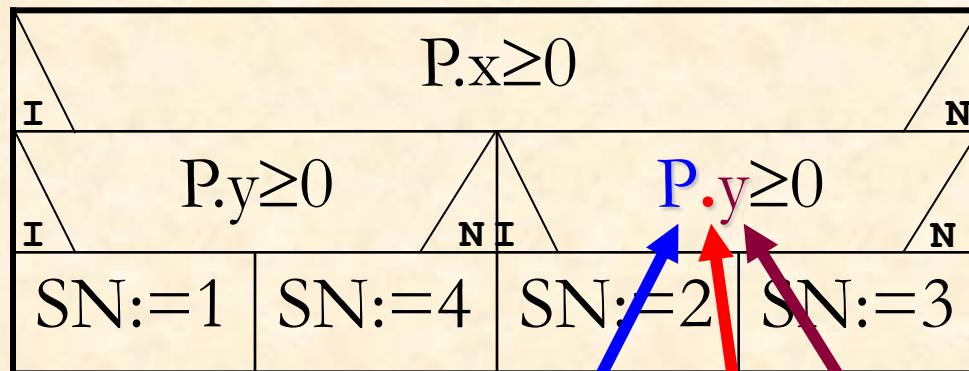
típusazonosító



Rekordok/Struktúrák

Algoritmus → kód:

➤ Utófeltétel: $P.x \geq 0$ és $P.y \geq 0 \rightarrow SN=1$ és
 $P.x < 0$ és $P.y \geq 0 \rightarrow SN=2$ és
 $P.x < 0$ és $P.y < 0 \rightarrow SN=3$ és
 $P.x \geq 0$ és $P.y < 0 \rightarrow SN=4$



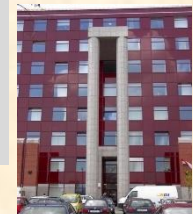
adatazonosító mezőazonosító

C# hivatkozás:

```

if (P.x >= 0) {
    if (P.y >= 0) SN=1;
    else SN=4;
} else {
    if (P.y >= 0) SN=2;
    else SN=3;
}
    
```

adatazonosító mezőazonosító



Kódolás (C#)



Kód:

kétirányú

elágazás

sokirányú
(általános)

```
if (Felt)
```

```
{
```

```
    UtI
```

```
}
```

```
else
```

```
{
```

```
    elhagyható
```

```
    UtH
```

```
}
```

```
if (Felt1) {
```

```
    Ut1
```

```
}
```

```
else if (...) {
```

```
    ...
```

```
}
```

```
else if (FeltN) {
```

```
    UtN
```

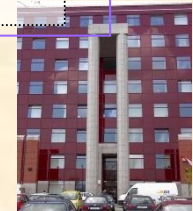
```
}
```

```
else {
```

```
    Ut
```

```
    elhagyható
```

```
}
```



Kódolás (C#)



Kód:

sokirányú
elágazás (speciális)

```
switch (kif)
{
    case érték1: Ut1; break;
    case ... : ... ; break;
    case értékN: UtN; break;
    default elhagyható; break;
}
```



Visszatekintés

- A problémamegoldás lépései – a programkészítés folyamata
- A specifikáció
- Az algoritmus
- Algoritmikus nyelvek – struktogram
- Adatokkal kapcsolatos fogalmak
- Adattípusok, elemi adattípusok
- Elemi feladatok – elágazások
- Rekordok
- Kódolás – elágazások

