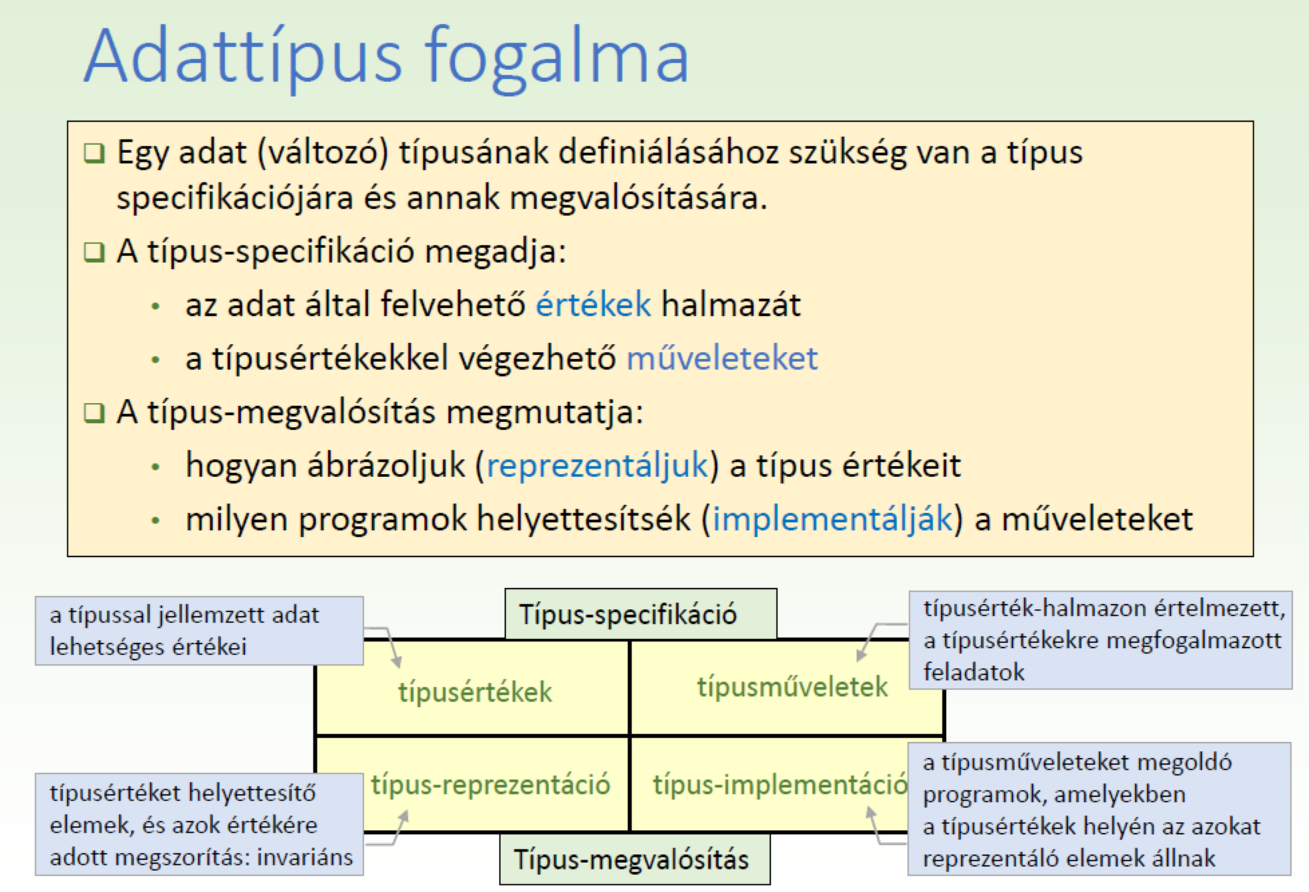
OEP

2. táblás gyakorlat
adattípus definiálás
egyszerű típusok

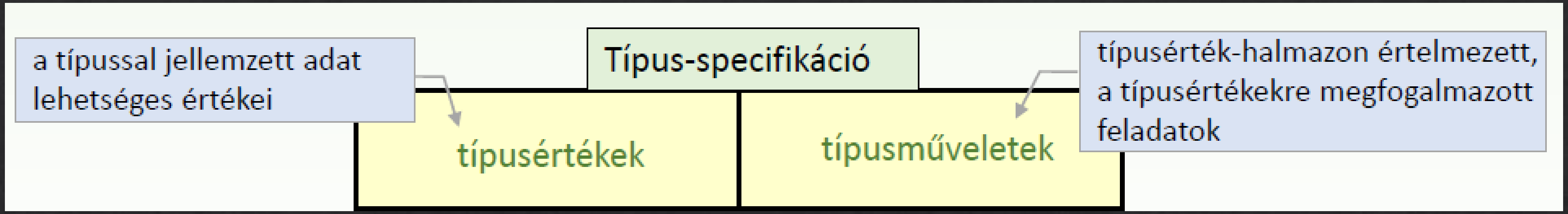
Tartalom

Adattípus fogalma (1. előadás)

- Típus-specifikáció
  - Típusértékek
  - Típusműveletek
- Típus megvalósítás
  - Típus-reprezentció
  - Típus-implementáció

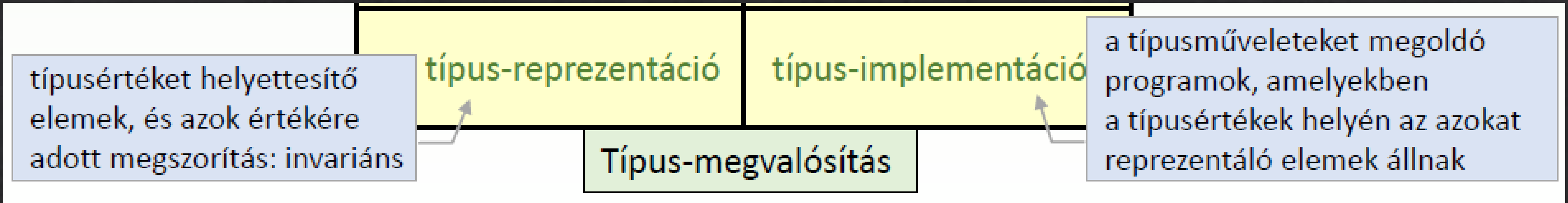


1. feladat: racionális számok típusa specifikáció



$\mathbb{Q}$	• összeadás/kivonás ($\boldsymbol{a}$: $\mathbb{Q}$, $\boldsymbol{b}$: $\mathbb{Q}$, $\boldsymbol{c}$: $\mathbb{Q}$) $\mathbf{c} \coloneqq \boldsymbol{a} \pm \boldsymbol{b}$ • szorzás/osztás ($\boldsymbol{a}$: $\mathbb{Q}$, $\boldsymbol{b}$: $\mathbb{Q}$, $\boldsymbol{c}$: $\mathbb{Q}$) $\boldsymbol{c} \coloneqq \boldsymbol{a} * \boldsymbol{b}$ $\boldsymbol{c} \coloneqq \dfrac{\boldsymbol{a}}{\boldsymbol{b}}$

1. feladat: racionális számok típusa
típus reprezentáció



- Ötlet: ábrázoljuk két egész szám hányadosával $\frac{x}{y}$, x,y: $\mathbb{Z}$
- Bármely két egészre igaz, hogy racionális számot ábrázol?
- Nullával nem lehet osztani $\Rightarrow$ y$\neq$ 0 (típus invariáns tulajdonság)

Típus-reprezentáció

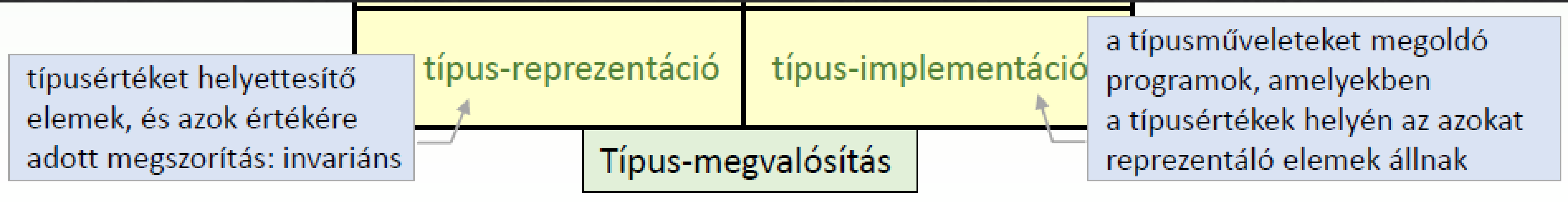
$x, y : \mathbb{Z}$

$//\frac{x}{y}$

Invariáns: $y \neq 0$

Kvíz: szorzás művelet?

1. feladat: racionális számok típusa
típus implementáció



- Összeadás/kivonás:
- Szorzás:
- Osztás: (b.x≠0):

Típus-implementáció

Összeadás/kivonás

$$c.x, c.y \coloneqq (a.x * b.y \pm a.y * b.x), a.y * b.y$$

Szorzás

$$c.x, c.y \coloneqq a.x * b.x, a.y * b.y$$

Osztás (b.x $\neq$ 0)

$$c.x, c.y \coloneqq a.x * b.y, a.y * b.x$$

Racionális szám típus implementálás, UML ábra

rNum
- x: int - y: int {y≠0}

//konstruktorok
+rNum(z:int,w:int) {w≠0}
//getterek
+getNumerator():int {query}
+getDenominator():int {query}
//setterek
+setNumerator(z:int)
+setDenominator(w:int) {w≠0}
+setrNum(z:int,w:int) {w≠0}
//műveletek
+add(a:rNum,b:rNum):rNum
+sub(a:rNum,b:rNum):rNum
+mul(a:rNum,b:rNum):rNum
+div(a:rNum,b:rNum):rNum {b.x≠0}

$\mathbb{Q}$	c := a±b (a:$\mathbb{Q}$, b:$\mathbb{Q}$, c:$\mathbb{Q}$)
	c := a*b (a:$\mathbb{Q}$, b:$\mathbb{Q}$, c:$\mathbb{Q}$)
	c := a/b (b≠0) (a:$\mathbb{Q}$, b:$\mathbb{Q}$, c:$\mathbb{Q}$)
x,y: $\mathbb{Z}$ (Inv: y≠0) // $\frac{x}{y}$	c.x, c.y := a.x*b.y ± a.y*b.x, a.y*b.y
	c.x, c.y := a.x * b.x, a.y * b.y
	c.x, c.y := a.x * b.y, a.y * b.x (b.x≠0)

Racionális szám típus implementálás UML ábra

rNum
- x: int - y: int {y≠0}

//konstruktorok
+rNum(z:int,w:int) {w≠0}
//getterek
+getNumerator():int {query}
+getDenominator():int {query}
//setterek
+setNumerator(z:int)
+setDenominator(w:int) {w≠0}
+setrNum(z:int,w:int) {w≠0}
//műveletek
+operator+(a:rNum,b:rNum):rNum
+operator-(a:rNum,b:rNum):rNum
+operator*(a:rNum,b:rNum):rNum
+operator/(a:rNum,b:rNum):rNum {b.x≠0}
 	
-simplify():void

$\mathbb{Q}$	c := a±b (a:$\mathbb{Q}$, b:$\mathbb{Q}$, c:$\mathbb{Q}$)
	c := a*b (a:$\mathbb{Q}$, b:$\mathbb{Q}$, c:$\mathbb{Q}$)
	c := a/b (b≠0) (a:$\mathbb{Q}$, b:$\mathbb{Q}$, c:$\mathbb{Q}$)
x,y: $\mathbb{Z}$ (Inv: y≠0) // $\frac{x}{y}$	c.x, c.y := a.x*b.y ± a.y*b.x, a.y*b.y
	c.x, c.y := a.x * b.x, a.y * b.y
	c.x, c.y := a.x * b.y, a.y * b.x (b.x≠0)

Érdemes kiegészíteni az osztályt egy egyszerűsítést végző eljárással (művelet elvégzése után elvégzi a lehetséges egyszerűsítést). Ez kívülről nem látható.

Komplex számok típusa

- Valósítsuk meg a komplex számok típusát! Ábrázoljuk a komplex számokat az algebrai alakjukkal (x+iy)! Implementáljuk a négy alapműveletet!

$\mathbb{C}$	c := a±b (a: $\mathbb{C}$, b:$\mathbb{C}$, c:$\mathbb{C}$) c := a*b (a:$\mathbb{C}$, b:$\mathbb{C}$, c:$\mathbb{C}$) c := a/b (a:$\mathbb{C}$, b:$\mathbb{C}$, c:$\mathbb{C}$) b≠0
x,y: $\mathbb{R}$ // x+iy	?

Műveletek komplex számokkal

- Összeadás/kivonás:
  (-3-2i)+(4+3i)=-3-2i+4+3i=1+i
  (3+2i)-(-4-3i)=3+2i+4+3i=7+5i

$$c.x,\ c.y := a.x \pm b.x,\ a.y \pm b.y$$

- Szorzás:
  $(3+2i)*(5-3i)=15-9i+10i-6i^2=15+6+(10-9)i=21+i$

$$c.x,\ c.y := a.x * b.x - a.y * b.y,\ a.x * b.y + a.y * b.x$$

- Osztás:
  (3+2i) / (5-3i) = ((3+2i)*(5+3i)) / ((5-3i)*(5+3i)) = (9+19i) / (25+9) = 9/34 + (19/34)i

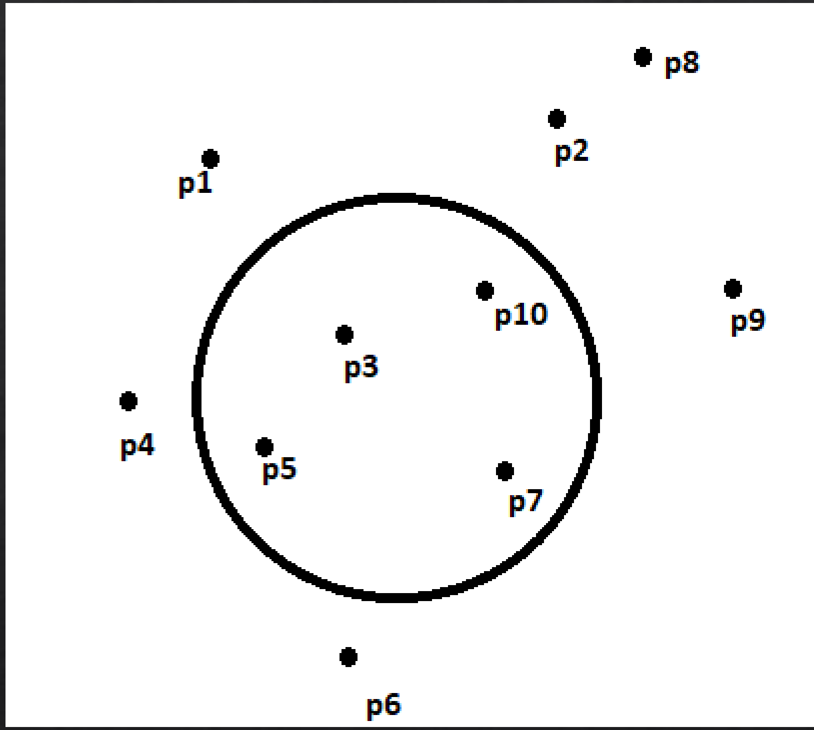
$$c.x,\ c.y := (a.x * b.x + a.y * b.y)/(b.x^2 + b.y^2),$$
$$(a.y * b.x - a.x * b.y)/(b.x^2 + b.y^2)$$
$$b.x \neq 0 \vee b.y \neq 0$$

Komplex számok típusa

$\mathbb{C}$	c := a±b (a: $\mathbb{C}$, b:$\mathbb{C}$, c:$\mathbb{C}$) c := a*b (a:$\mathbb{C}$, b:$\mathbb{C}$, c:$\mathbb{C}$) c := a/b (a:$\mathbb{C}$, b:$\mathbb{C}$, c:$\mathbb{C}$) b≠0
x,y: $\mathbb{R}$ // x+iy	c.x, c.y := a.x ± b.x, a.y ± b.y c.x, c.y := a.x * b.x – a.y * b.y , a.x * b.y + a.y * b.x c.x, c.y := (a.x * b.x + a.y * b.y)/(b.x^2+ b.y^2), (a.y * b.x – a.x * b.y)/(b.x^2+ b.y^2) b.x≠0 ∨ b.y≠0

Pont és kör

- Feladat:
  Adott egy síkbeli pontokat tartalmazó tömb, és egy kör. A tömbnek van-e olyan eleme, amely a körben helyezkedik el? Ha igen, adjuk meg az első ilyen pontot!
- Ötlet:
  Készítsünk a feladathoz egy pont és egy kör típust!



Pont típus

- Valósítsuk meg a pont típus, mely a síkbeli koordináta-rendszer pontjainak a kezelésére lesz alkalmas!
- Implementáljuk két pont távolságát kiszámoló metódust!

Pont	d:= távolság(p, q) (p, q :Pont, d:$\mathbb{R}$)
(x, y) :$\mathbb{R}^2$	d:= $\sqrt{(p.x - q.x)^2 + (p.y - q.y)^2}$

Point
+ x : real + y : real
+ distance(Point) : real

Kör típus

- Valósítsunk meg egy kör típust, amely használja a pont típust!
- Implementáljuk annak eldöntését, hogy egy adott pont rajta van-e a körlemezen.
- Kvíz: hogyan reprezentáljuk a kört?

Kör	$l := \text{tartalmaz}(k, p) \quad (k:\text{Kör}, p:\text{Pont}, l:\mathbb{L})$
c : Pont, $r : \mathbb{R}$ Inv: $r \geq 0$	$l := \text{távolság}(k.c, p) \leq k.r$

Circle
- c : Point - r: real $\{ r \geq 0 \}$
+ contains(p : Point) : bool <<getter>> + getc() : Point + getr() : real

A feladat megoldása

- Specifikáció:
- $A = (x{:}Point^{n},\ k{:}Circle,\ l{:}\mathbb{L},\ ind : \mathbb{Z})$
- $Ef = (\ x=x' \wedge k=k'\ )$
- $Uf = (\ Ef\ \wedge (l,\ ind) = SEARCH_{i=1..n}\ k.contains(x[i]))$
- Tétel: lineáris keresés

6. Keresés (szekvenciális vagy lineáris keresés)

Feladat: Adott egy *felt*:$[m..n]\to\mathbb{L}$ feltétel. Határozzuk meg az intervallum első olyan elemét, amelyre teljesül a feltétel! (Ez az ún. pesszimista lineáris keresés.)

Specifikáció:

$A\ = (m{:}\mathbb{Z},\ n{:}\mathbb{Z},\ l{:}\mathbb{L},\ ind{:}\mathbb{Z})$

$Ef = (\ m=m' \wedge n=n'\ )$

$Uf = (Ef \wedge l,\ ind =$
$SEARCH_{i=m..n} felt(i))$

Algoritmus:

$l,\ i := hamis,\ m$	$i{:}\mathbb{Z}$
$\neg l \wedge i \leq n$	
$l,\ ind := felt(i),\ i$	
$i := i+1$	

- $A = (x:Point^n,\ k:Circle,\ l:\mathbb{L},\ ind:\mathbb{Z})$
- $Ef = (x=x' \wedge k=k')$
- $Uf = (Ef \wedge (l, ind) = SEARCH_{i=1..n}\ k.contains(x[i]))$
- Visszavezetés:

$m..n \sim 1..n$

$felt(i) \sim k.contains(x[i])$

Algoritmus:

i, l := 1, false	i:ℤ
¬l ∧ i ≤ n	
l, ind := k.contains(x[i]), i	
i := i+1	

6. Keresés (szekvenciális vagy lineáris keresés)

Feladat: Adott egy $felt:[m..n] \rightarrow \mathbb{L}$ feltétel. Határozzuk meg az intervallum első olyan elemét, amelyre teljesül a feltétel! (Ez az ún. pesszimista lineáris keresés.)

Specifikáció:

$A = (m:\mathbb{Z},\ n:\mathbb{Z},\ l:\mathbb{L},\ ind:\mathbb{Z})$

$Ef = (m=m' \wedge n=n')$

$Uf = (Ef \wedge l, ind = SEARCH_{i=m..n} felt(i))$

Algoritmus:

l, i := hamis, m	i:ℤ
¬l ∧ i≤n	
l, ind := felt(i), i	
i := i+1	

Feltételes maximum keresés

- Azon pontok közül, amelyek nem a körlemezre esnek, melyik van a legközelebb a körlemezhez?
- Specifikáció:
- A = (x:Point ^{n}, k:Circle, l:$\mathbb{L}$, min:R, ind : $\mathbb{Z}$)
- Ef = (x=x' $\wedge$ k=k')
- Uf = (Ef $\wedge$ (l, min, ind) = $\text{Min}_{i=1..n}$ x[i].distance(k.getc())
  $\neg$ k.contains(x[i])
- Tétel: feltételes maximum (minimum) keresés

Feltételes maximum keresés programozási tétel

4. *Feltételes maximumkeresés*

Feladat: Adott egy $f:[m..n] \to H$ függvény és egy $felt:[m..n] \to \mathbb{L}$ feltétel. A H halmaz elemein értelmezett egy teljes rendezési reláció. Határozzuk meg, melyik a függvény legnagyobb értéke azok között, amelyeket olyan intervallumbeli elemhez rendel, amelyek kielégíti a feltételt! Adjuk meg az egyik olyan intervallumbeli elemet, amelyre a feltétel teljesül és ahol a függvény ezt a maximális értéket felveszi!

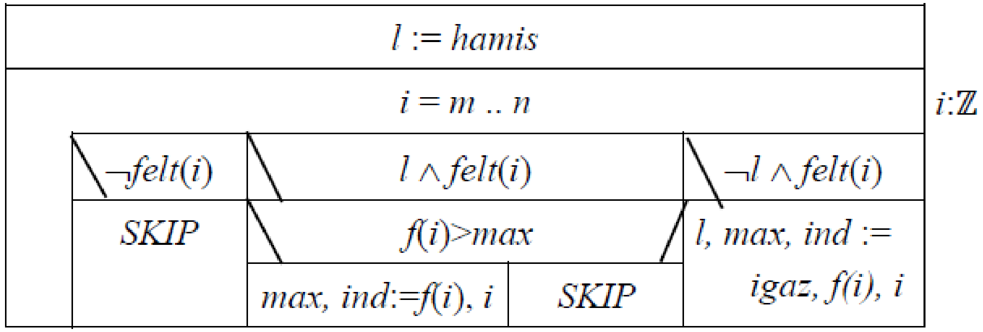
Specifikáció:

$A = (m:\mathbb{Z},\ n:\mathbb{Z},\ l:\mathbb{L},\ ind:\mathbb{Z},\ max:H)$

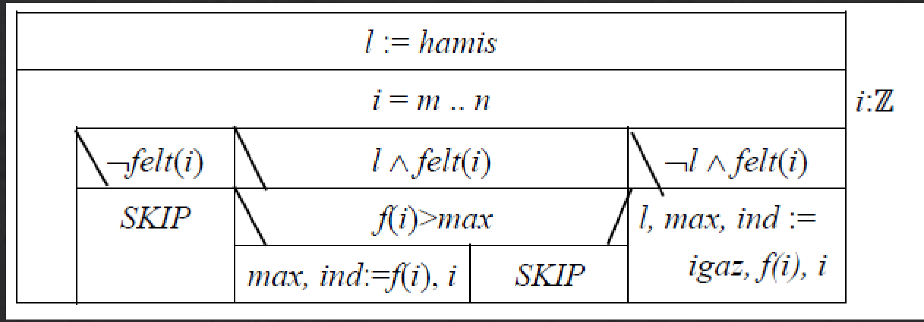
$Ef = (\ m=m' \wedge n=n'\ )$

$Uf = (\ Ef \wedge (l,max,ind) = \underset{felt(i)}{MAX_{i=m..n}} f(i)$

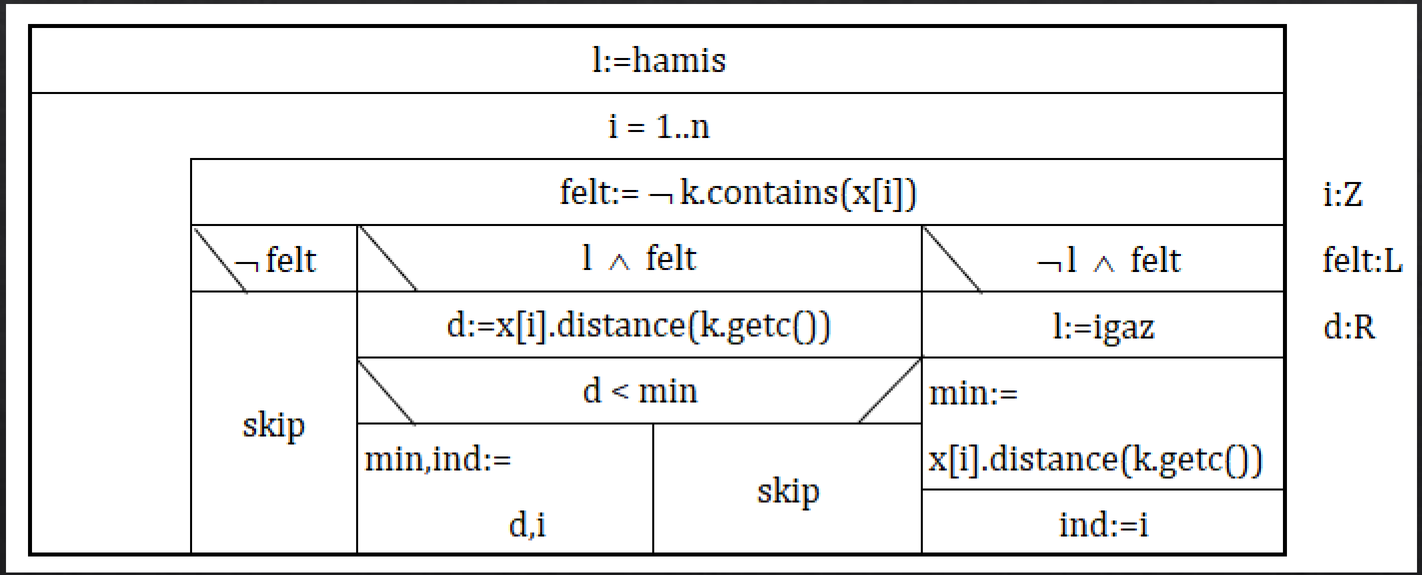
Algoritmus:



- $A = (x{:}Point^{n}, k{:}Circle, l{:}\mathbb{L}, min{:}R, ind : \mathbb{Z})$
- $Ef = (x=x' \wedge k=k')$
- $Uf = (Ef \wedge (l, min, ind) = Min_{i=1..n}\ x[i].distance(k.getc())$
  $\neg\ k.contains(x[i]))$



- Visszavezetés:
  m..n ~ 1..n
  f(i) ~ x[i].distance(k.getc())
  felt(i) ~ ¬ k.contains(x[i])
  H,> ~ R,<



Síkvektorok

- Valósítsuk meg a síkvektorok típusát!
- Implementáljuk két vektor összegének, egy vektor nyújtásának, és két vektor skaláris szorzatának műveleteit!
- Feladat:
  Adott síkvektorok összege merőleges-e egy adott síkvektorra (skaláris szorzatuk nulla-e).
- Kvíz: melyik nem lenne jó reprezentáció?
- Síkvektor reprezentáció:
  „origóból az (x_1, x_2) koordinátájú pontba mutató vektorok”

Két vektor összege rajzban a paralelogramma-szabály szerint képezhető

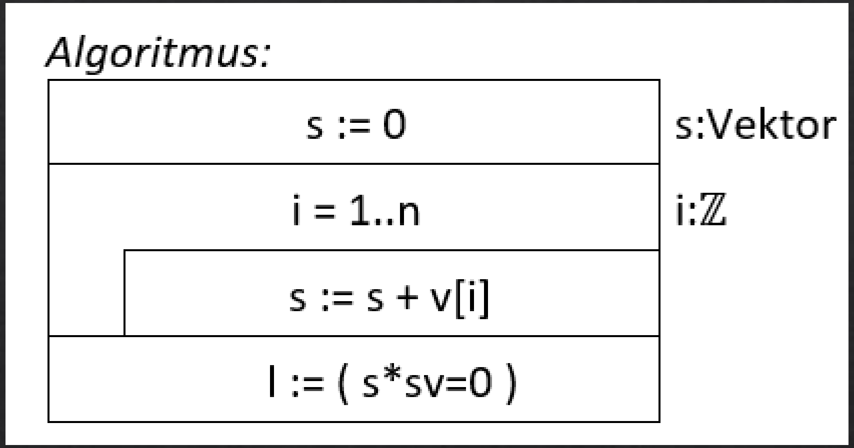
Síkvektorok típusa

Vektor	c := a+b	(a, b, c : Vektor)
	c := k*a	(a, c : Vektor, k:ℝ)
	s := a*b	(a, b: Vektor, s:ℝ)

x, y:ℝ	c.x, c.y := a.x+b.x, a.y+b.y
	c.x, c.y := k*a.x, k*a.y
	c.x, c.y:= a.x*b.x + a.y*b.y

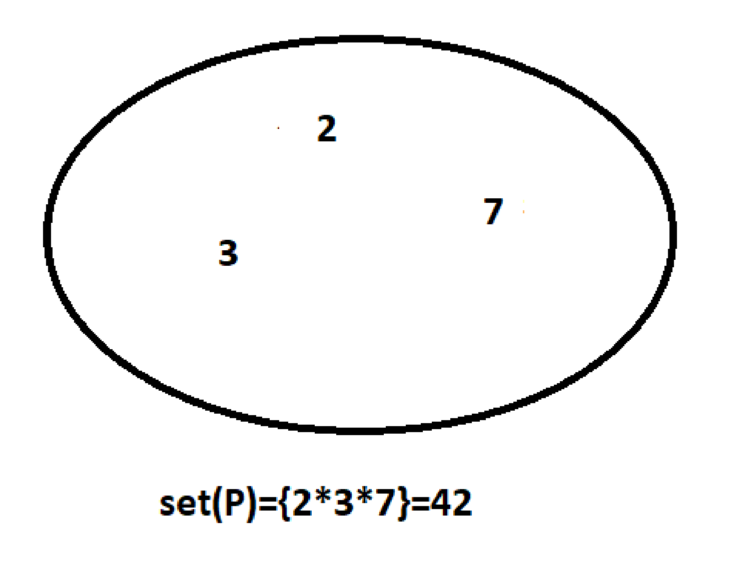
Feladat megoldása

- Specifikáció:
- $A = (v:Vektor^{n}, sv:Vektor, l: \mathbb{L})$
- $Ef = (v=v' \wedge sv=sv')$
- $Uf = (Ef \wedge l = ((\Sigma_{i=1..n} v[i]) * sv = 0.0))$
- Tétel: összegzés
- Visszavezetés:
  $m..n \sim 1..n$
  $f(i) \sim v[i]$
  $s \sim s$
  $H,+,0 \sim Vektor,+,0$ //(0,0) = nullvektor



Prímek halmaza

- Valósítsuk meg a prímek halmazának típusát a következőképpen: A prímek halmazát egy természetes számmal tudjuk reprezentálni úgy, hogy a szám a halmazban szereplő prímek szorzata.
- Implementáljuk a következő műveleteket:
  - benne van-e,
  - betesz,
  - kivesz,
  - üres,
  - üres-e,
  - minimum,
  - elemszám.



Prímek halmaza

set($\mathbb{P}$) prímek véges halmaza	l := h.benne_van-e (p)	(h : set($\mathbb{P}$), p : $\mathbb{P}$, l : $\mathbb{L}$)
	h.betesz(p) // ha p-t h már tartalmazza, akkor hatástalan	(h : set($\mathbb{P}$), p : $\mathbb{P}$)
	h.kivesz(p) // ha p-t h nem tartalmazza, akkor hatástalan	(h : set($\mathbb{P}$), p : $\mathbb{P}$)
	h.üres()	(h : set($\mathbb{P}$))
	l := h.üres-e()	(h : set($\mathbb{P}$), l : $\mathbb{L}$)
	p := h.min() // hiba, ha p üres halmaz	(h : set($\mathbb{P}$), p : Prím)
	c := h.elemszám()	(h : set($\mathbb{P}$), c : $\mathbb{N}$)

Kvíz: hatékony implementálás?

- Reprezentáció:

$n : \mathbb{N}$

ahol az n szám prímtényezős felbontásában a prímek egyszeresen fordulnak elő

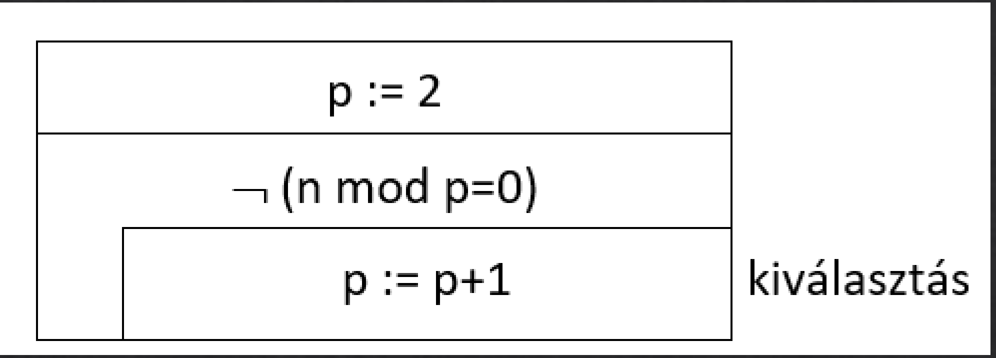
- Műveletek:

l := h.benne_van-e (p)	($h : set(\mathbb{P}), p : \mathbb{P}, l : \mathbb{L}$)	l := (n mod p=0)
h.betesz(p) // ha p-t h már tartalmazza, akkor hatástalan	($h : set(\mathbb{P}), p : \mathbb{P}$)	$n := n \cdot p$ ha $n \bmod p \neq 0$
h.kivesz(p) // ha p-t h nem tartalmazza, akkor hatástalan	($h : set(\mathbb{P}), p : \mathbb{P}$)	$n := n/p$ ha $n \bmod p = 0$
h.üres()	($h : set(\mathbb{P})$)	n := 1
l := h.üres-e()	($h : set(\mathbb{P}), l : \mathbb{L}$)	l := (n=1)

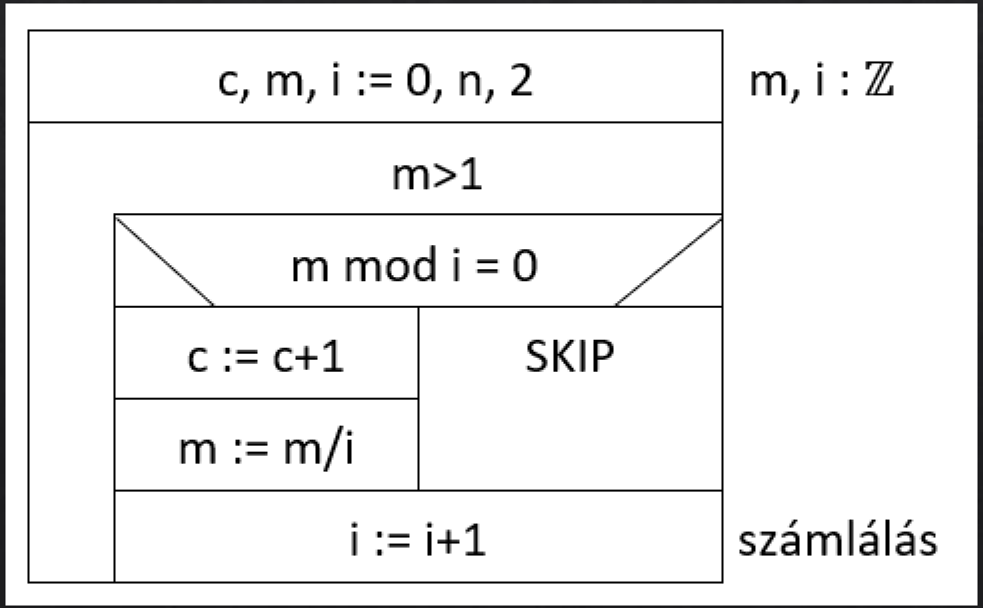
- Műveletek:

$p := h.\mathrm{min}()$ $\qquad\qquad$ $(h : \mathrm{set}(\mathbb{P}),\ p : \mathrm{Prím})$

// hiba, ha p üres halmaz



$c := h.\mathrm{elemszám}()$ $\qquad\qquad$ $(h : \mathrm{set}(\mathbb{P}),\ c : \mathbb{N})$



Másodfokú polinom

- Feladat:
  Adott egy valós számokat tartalmazó tömb, és egy másodfokú polinom. A tömb mely elemére ad a másodfokú polinom maximális helyettesítési értéket?
- Hozzunk létre a másodfokú polinomok típusát.
- Implementáljuk a következő műveleteket:
  - Helyettesítési értékének kiszámítása
  - Polinomok összeadása, szorzása számmal

Másodfokú polinom típusa

másodfokú polinomok (MP)	h := érték(p, x) (p : MP, x : $\mathbb{R}$, h : $\mathbb{R}$)
	p3 := p1$\pm$p2 (p1, p2, p3:MP)
	p2 := p1*q (p1, p2:MP, q: $\mathbb{R}$)
a,b,c : $\mathbb{R}$ // ax^2+bx+c	h := p1.a*x*x+p1.b*x+p1.c
	p3.a, p3.b, p3.c := p1.a $\pm$ p2.a, p1.b $\pm$ p2.b, p1.c $\pm$ p2.c
	p3.a, p3.b, p3.c := p3.a*q, p3.b*q, p3.c*q

Polinom (MP)
- a : real
- b : real
- c : real
+ value(x : real) : real
+ add(p1 : MP, p2 : MP) : MP
+ mul(p1 : MP, q : real) : MP

A feladat megoldása

- Specifikáció:
- A = (p : MP, v : $\mathbb{R}^n$, ind : $\mathbb{Z}$, max : $\mathbb{R}$)
- Ef = (p=p’ $\wedge$ v=v’)
- Uf = (Ef $\wedge$ (max, ind) = $\mathrm{MAX}_{i=1..n}$ p.value(v[i]))
- Visszavezetés: maximum kiválasztás
  m..n ~ 1..n
  f(i) ~ p.value(v[i])
  H,> ~ R,>