OEP 1. gyakorlat

1. feladat

- Adottak az x és y nem negatív egész számok. Számítsuk ki a szorzatukat, úgy, hogy csak összeadást használhatunk.
- Feladat állapottere (bemeneti és kimeneti változók és típusaik)
- A=(x:ℕ, y:ℕ, z:ℕ)
- Előfeltétel (mely változók rendelkeznek kezdőértékkel, valamint a kezdőértékre vonatkozó elvárások)
- Ef=($x=x_0 \wedge y=y_0$)   másképp írva: Ef=($x=x' \wedge y=y'$)
- Utófeltétel (utalunk a programozási tételre is)
- Uf=($z = \sum_{i=1}^{x_0} y_0$)   vagy   Uf=(Ef $\wedge\ z = \sum_{i=1}^{x} y$)

Összegzés programozási tétel

1. Összegzés

Feladat: Adott egy f:[$m..n$]→H függvény. A H halmaz elemein értelmezett egy asszociatív, baloldali nulla elemmel rendelkező művelet (nevezzük ezt összeadásnak és jelölje a +). Határozzuk meg a függvény intervallumon felvett értékeinek összegét!

Specifikáció:

A = (m:ℤ, n:ℤ , s:H)
Ef = ($m=m' \wedge n=n'$)
Uf = ($Ef \wedge s = \sum_{i=m..n} f(i)$)

Algoritmus:

$s := 0$	
$i = m .. n$	i:ℤ
$s := s+f(i)$	

Visszavezetés az összegzés tételre

- A=(x:ℕ, y:ℕ, z:ℕ)
- Ef=($x=x_0 \wedge y=y_0$)
- Uf=(Ef $\wedge\ z = \sum_{i=1}^{x} y$)

1. Összegzés

Feladat: Adott egy f:[$m..n$]→H függvény. A H halmaz elemein értelmezett egy asszociatív, baloldali nulla elemmel rendelkező művelet (nevezzük ezt összeadásnak és jelölje a +). Határozzuk meg a függvény intervallumon felvett értékeinek összegét!

Specifikáció:

A = (m:ℤ, n:ℤ , s:H)
Ef = ($m=m' \wedge n=n'$)
Uf = ($Ef \wedge s = \sum_{i=m..n} f(i)$)

Algoritmus:

$s := 0$
$i = m .. n$ i:ℤ
$s := s+f(i)$

i=m..n	~	i=1..x
s	~	z
f(i)	~	y
H,+,0	~	N,+,0

Visszavezetéssel kapott megoldó algoritmus

- A=(x:ℕ, y:ℕ, z:ℕ)
- Ef=($x=x_0 \wedge y=y_0$)
- Uf=(Ef $\wedge\ z = \sum_{i=1}^{x} y$)

1. Összegzés

Feladat: Adott egy f:[$m..n$]→H függvény. A H halmaz elemein értelmezett egy asszociatív, baloldali nulla elemmel rendelkező művelet (nevezzük ezt összeadásnak és jelölje a +). Határozzuk meg a függvény intervallumon felvett értékeinek összegét!

Specifikáció:

A = (m:ℤ, n:ℤ , s:H)
Ef = ($m=m' \wedge n=n'$)
Uf = ($Ef \wedge s = \sum_{i=m..n} f(i)$)

Algoritmus:

$s := 0$
$i = m .. n$ i:ℤ
$s := s+f(i)$

i=m..n	~	i=1..x
s	~	z
f(i)	~	y
H,+,0	~	N,+,0

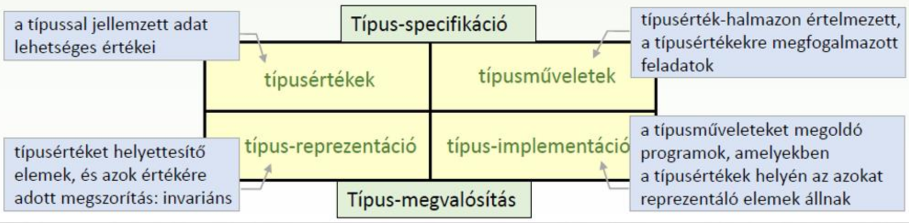
z := 0
i = 1..x   i:ℤ
z := z + y

Adattípus fogalma (1. előadás)

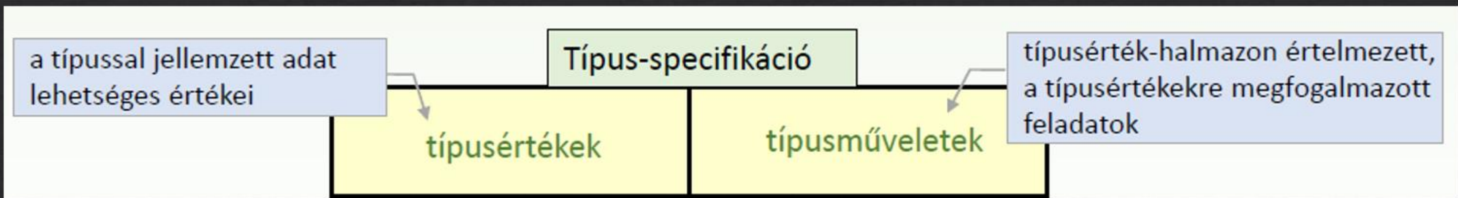
- Típus-specifikáció
  - Típusértékek
  - Típusműveletek
- Típus megvalósítás
  - Típus-reprezentáció
  - Típus-implementáció

Adattípus fogalma

- Egy adat (változó) típusának definiálásához szükség van a típus specifikációjára és annak megvalósítására.
- A típus-specifikáció megadja:
  - az adat által felvehető értékek halmazát
  - a típusértékekkel végezhető műveleteket
- A típus-megvalósítás megmutatja:
  - hogyan ábrázoljuk (reprezentáljuk) a típus értékeit
  - milyen programok helyettesítsék (implementálják) a műveleteket

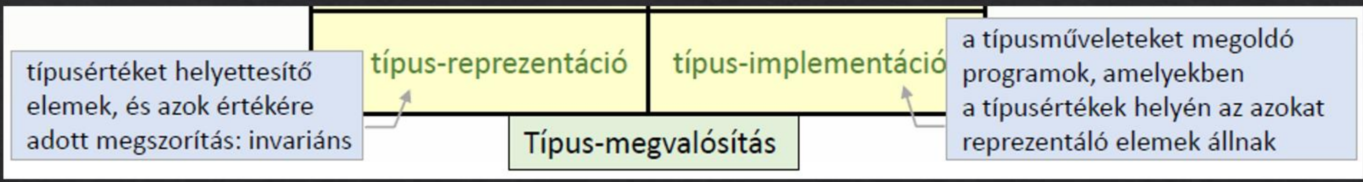


2. feladat: racionális számok típusa specifikáció



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

2. feladat: racionális számok típusa
típus reprezentáció



- Ötlet: ábrázoljuk két egész szám hányadosával
  $\frac{x}{y}$, x,y: ℤ
- Nullával nem lehet osztani ⇒ y≠ 0
  (típus invariáns tulajdonság)

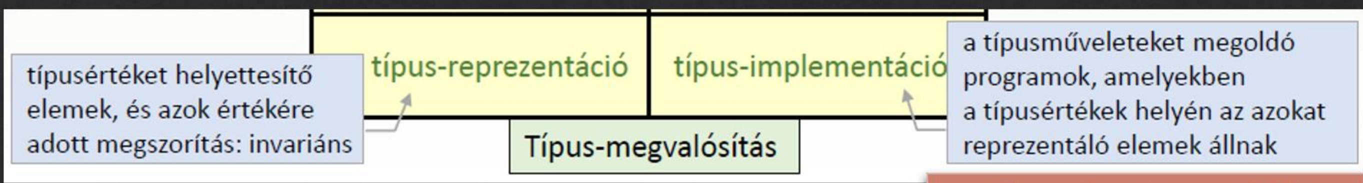
Típus-reprezentáció

$x, y : \mathbb{Z}$

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

Invariáns: $y \neq 0$

2. feladat: racionális számok típusa
típus implementáció



- Összeadás/kivonás:

$$\frac{a.x}{a.y} \pm \frac{b.x}{b.y} = \frac{a.x * b.y \pm a.y * b.x}{a.y * b.y}$$

- Szorzás:

$$\frac{a.x}{a.y} * \frac{b.x}{b.y} = \frac{a.x * b.x}{a.y * b.y}$$

- Osztás: (b.x≠0):

$$\frac{\frac{a.x}{a.y}}{\frac{b.x}{b.y}} = \frac{a.x}{a.y} * \frac{b.y}{b.x} = \frac{a.x * b.y}{a.y * b.x}$$

Típus-implementáció

Összeadás/kivonás

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

Szorzás

$c.x, c.y := a.x * b.x, a.y * b.y$

Osztás ($b.x \neq 0$)

$c.x, c.y := a.x * b.y, a.y * b.x$

Racionális szám típus implementálás, UML ábra

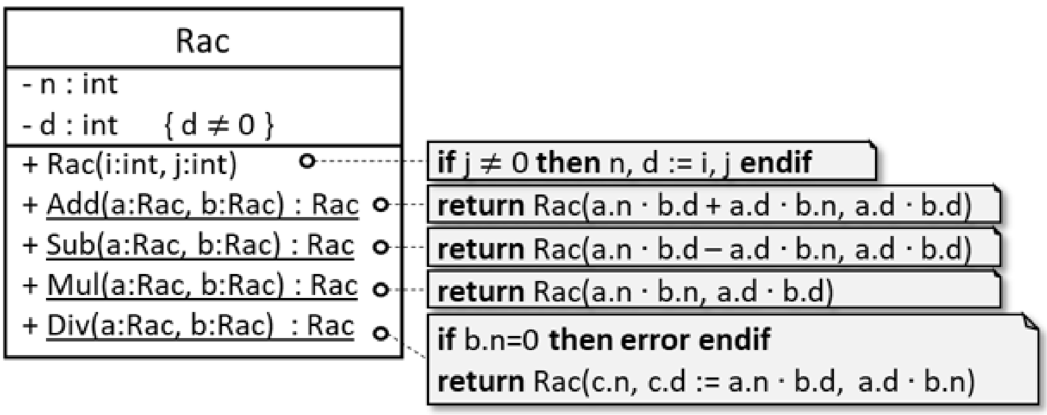
$\mathbb{Q}$	c := a$\pm$b (a:$\mathbb{Q}$, b:$\mathbb{Q}$, c:$\mathbb{Q}$)
	c := a*b (a:$\mathbb{Q}$, b:$\mathbb{Q}$, c:$\mathbb{Q}$)
	c := a/b (b$\neq$0) (a:$\mathbb{Q}$, b:$\mathbb{Q}$, c:$\mathbb{Q}$)
x,y: $\mathbb{Z}$ (Inv: y$\neq$0) // $\frac{x}{y}$	c.x, c.y := a.x*b.y $\pm$ a.y*b.x, a.y*b.y
	c.x,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

Osztálydiagram:

n := x, d := y, a.n := a.x, b.n := b.x, a.d := a.y, b.d := b.y

A műveleteket feladatoknak is tekinthetjük, amelyeket elő- utófeltételes specifikációval is megfogalmazhatunk mind a típusspecifikáció, mind a típusmegvalósítás szintjén:

A = (a:$\mathbb{Q}$, b:$\mathbb{Q}$, c:$\mathbb{Q}$)		*A* = (a:(n: $\mathbb{Z}$, d: $\mathbb{Z}$), b:(n: $\mathbb{Z}$, d: $\mathbb{Z}$), c:(n: $\mathbb{Z}$, d: $\mathbb{Z}$))
Ef = (a=a’ $\wedge$ b=b’)	ill.	*Ef* = (a=a’ $\wedge$ b=b’)
Uf = (*Ef* $\wedge$ b$\neq$0 $\rightarrow$ c=a / b)		*Uf* = (*Ef* $\wedge$ b.n$\neq$0 $\rightarrow$ c.n, c.d = a.n · b.d, a.d · b.n)



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 ± b.x, a.y ± 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

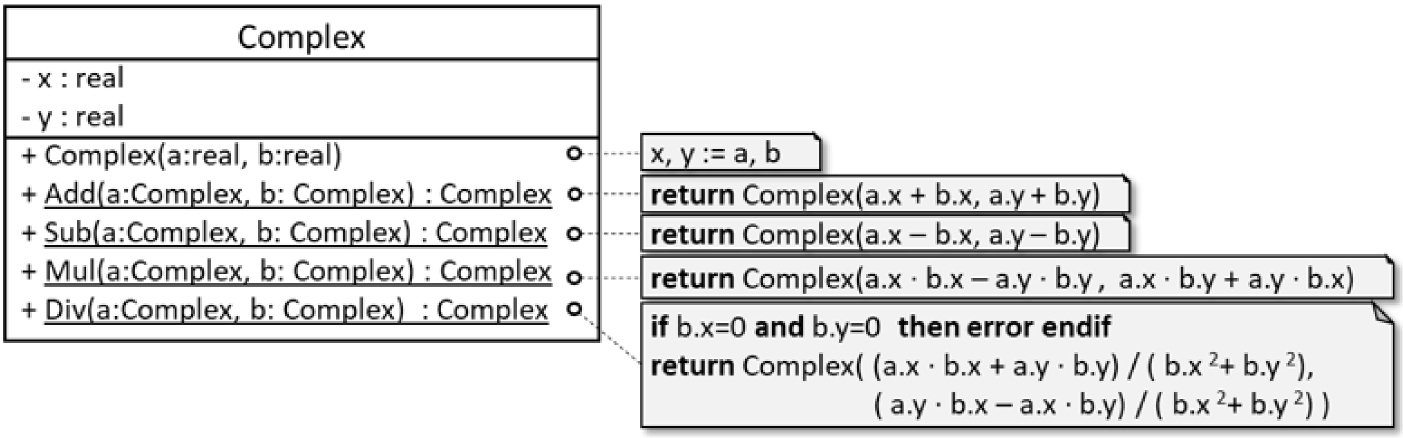
- Osztás:
  (3+2i) / (5-3i) = ((3+2i)*(5+3i)) / ((5-3i)*(5+3i)) = (9+19i) / (25+9) = 9/34 + (19/34)i

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≠0 ∨ b.y≠0

Komplex számok típusa

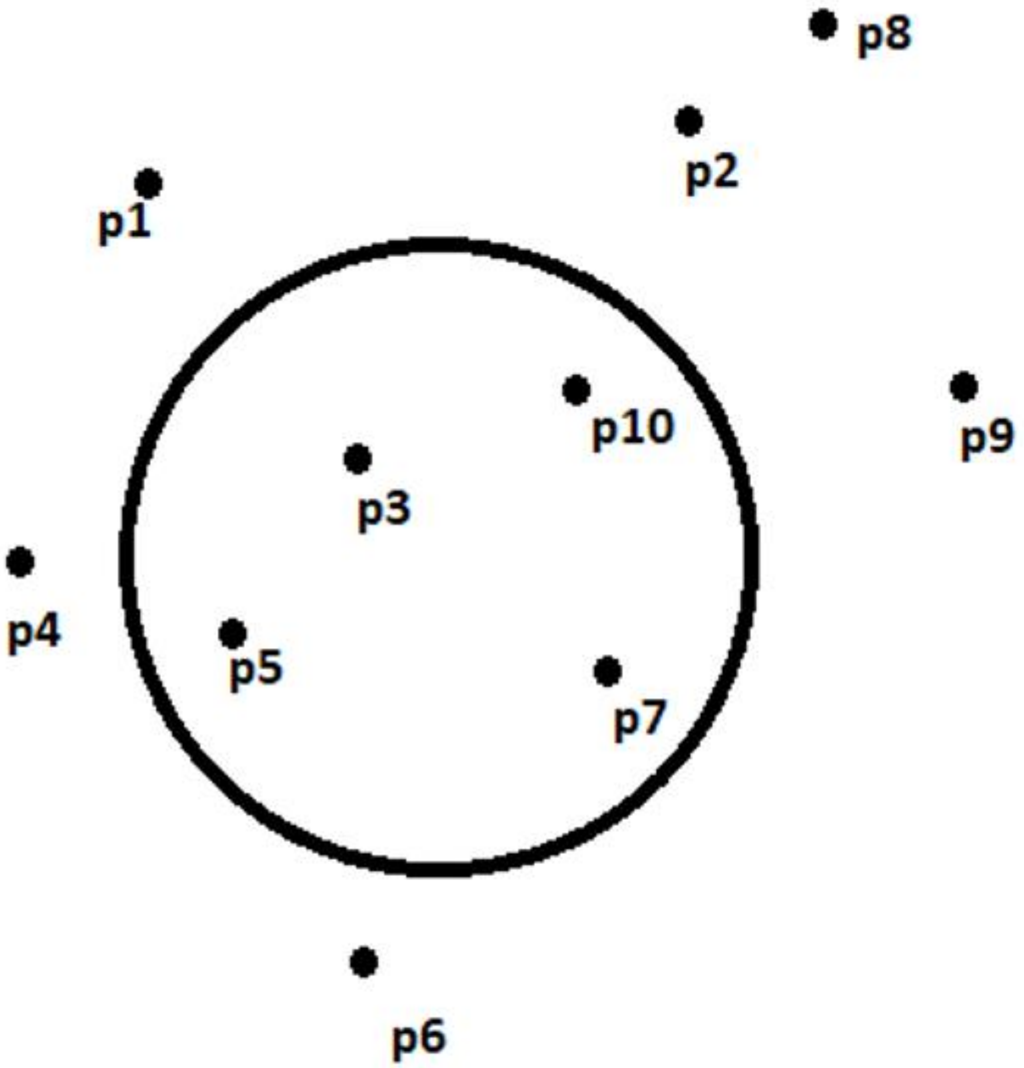
$\mathbb{C}$	c := a±b (a: $\mathbb{C}$, b:$\mathbb{C}$, c:$\mathbb{C}$) c := a*b (a:$\mathbb{C}$, b:$\mathbb{C}$, c:$\mathbb{C}$) c := a/b (a:$\mathbb{C}$, b:$\mathbb{C}$, c:$\mathbb{C}$) b≠0
x,y: $\mathbb{R}$ // x+iy	c.x, c.y := a.x ± b.x, a.y ± b.y c.x, c.y := a.x * b.x – a.y * b.y , a.x * b.y + a.y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

Osztálydiagram:



Pont és kör

Adott síkbeli pontok közül hány esik rá egy adott kör lemezére?



Specifikáció:

$A = (x{:}\mathrm{Pont}^{n},\ k{:}\mathrm{K\ddot{o}r},\ db{:}\mathbb{N})$

$Ef = (\ x=x' \wedge k=k'\ )$

$Uf = (\ Ef \wedge db = \sum_{\substack{i=1..n \\ x[i]\in k}} 1\ )$

Számlálás

i = m .. n ~ i = 1 .. n
felt(i) ~ x[i]∈k

Algoritmus:

i:$\mathbb{Z}$

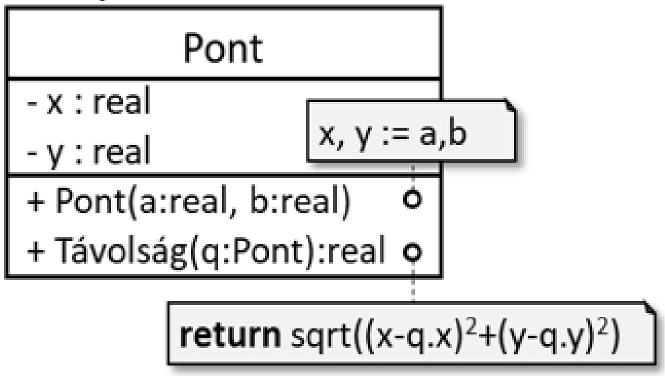
db := 0		
i = 1 .. n		
	x[i]∈k	
	db := db + 1	–

Síkbeli pont típusa. Ábrázoljuk a pontokat a koordinátáikkal.

Típusdefiníció:

Pont	$d := \lvert\overline{p, q}\rvert \quad (p, q : \text{Pont}, d : \mathbb{R})$
$x, y : \mathbb{R}$	$d := \sqrt{(p.x - q.x)^2 + (p.y - q.y)^2}$

Osztály:

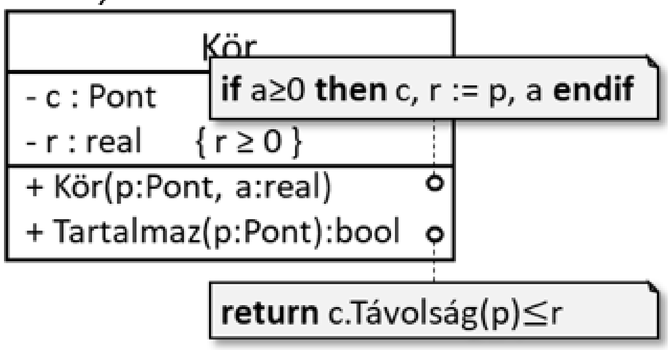


Kör típusa. Ábrázoljuk a köröket a középpontjukkal és a sugarukkal.

Típusdefiníció:

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

Osztály:



Megj: A tervezés során inkább a felülről-lefelé „irányt” követjük, de objektum-orientált kódolás az alulról felfelé építkezést szereti.