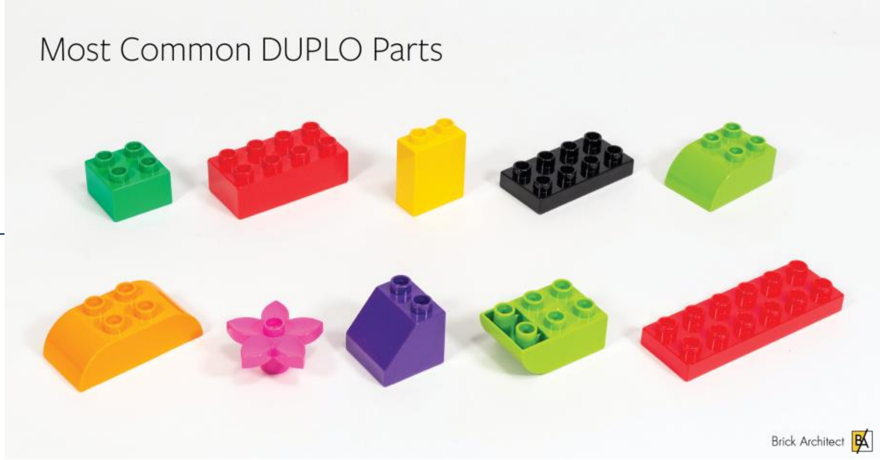
ELTE | IK

PROGRAMOZÁS

Programozási minták

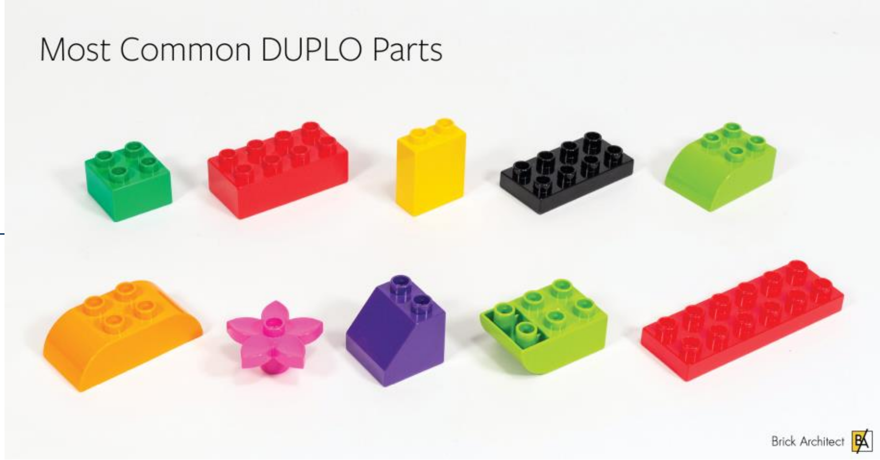
Horváth Győző, Horváth Gyula, Szlávi Péter

Programozási minták



1. Összegzés
   1. Általános összegzés
2. Megszámolás
3. Maximumkiválasztás
   a. Minimumkiválasztás
4. Feltételes maximumkeresés
5. Keresés
6. Eldöntés
   a. Mind eldöntés
7. Kiválasztás
8. Másolás
9. Kiválogatás

Programozási minták



1. Általános összegzés
    a. Megszámolás
    b. (Feltételes összegzés)
    c. Másolás
    d. Kiválogatás
2. Maximumkiválasztás
    a. Minimumkiválasztás
3. Feltételes maximumkeresés
4. Keresés
    a. Eldöntés
    b. Mind eldöntés
5. Kiválasztás

Összegzés
sablon

Feladat

Adott az egész számok egy [e..u] intervalluma és egy f:[e..u]→H függvény. A H halmaz elemein értelmezett az összeadás művelet. Határozzuk meg az f függvény [e..u] intervallumon felvett értékeinek az összegét, azaz a $\sum_{i=e}^{u} f(i)$ kifejezés értékét! (e>u esetén ennek az értéke definíció szerint a nulla elem)

Specifikáció

```
Be: e∈Z, u∈Z
Ki: s∈H
Ef: -
Uf: s=SZUMMA(i=e..u, f(i))
```

Algoritmus

```
s:=0
i=e..u
    s:=s+f(i)
```

Általános összegzés sablon

Művelet neve	Operátor	Nul
összeadás	+	0
szorzás	*	1
szövegösszefűzés	+	""

Művelet neve	Operátor	Nullelem
únió	U	Ø
logikai és	és	igaz
logikai vagy	vagy	hamis
tömbhöz fűzés	hozzáfűz	üres tömb

Feladat

Adott az egész számok egy [e..u] intervalluma és egy f:[e..u]→H függvény. A H halmaz elemein értelmezett egy asszociatív, baloldali nulla elemmel rendelkező művelet, amit most összeadásnak nevezünk és +-szal jelöljük. Határozzuk meg az f függvény [e..u] intervallumon felvett értékeinek az összegét, azaz a $\sum_{i=e}^{u} f(i)$ kifejezés értékét! (e>u esetén ennek az értéke definíció szerint a nulla elem)

Specifikáció

```
Be: e∈Z, u∈Z
Ki: s∈H
Ef: -
Uf: s=SZUMMA(i=e..u,f(i),0,+)
```

Algoritmus

s:=0
i=e..u
s:=s+f(i)

Megszámolás
sablon

i	T(i)	érték
e	IGAZ	1
e+1	HAMIS	0
...	HAMIS	0
u	IGAZ	1
	db=	2

Feladat

Adott az egész számok egy [e..u] intervalluma és egy T:[e..u]→Logikai feltétel. Határozzuk meg, hogy az [e..u] intervallumon a T feltétel hányszor veszi fel az igaz értéket!

Specifikáció

Be: e∈Z, u∈Z

Ki: db∈N

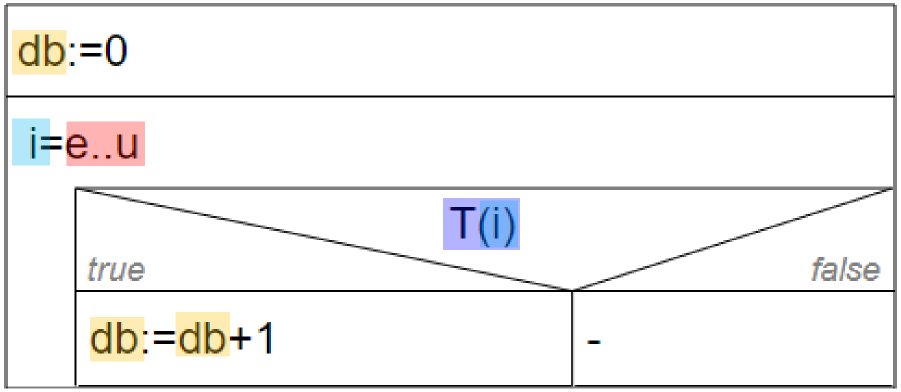
Ef: -

Uf: db=**SZUMMA**(i=e..u, 1, T(i))

Rövidítve:

Uf: db=**DARAB**(i=e..u, T(i))

Algoritmus



Maximumkiválasztás
sablon

Feladat

Adott az egész számok egy [e..u] intervalluma és egy f:[e..u]→H függvény. A H halmaz elemein értelmezett egy teljes rendezési reláció. Határozzuk meg, hogy az f függvény hol veszi fel az [e..u] nem üres intervallumon a legnagyobb értéket, és mondjuk meg, mekkora ez a maximális érték!

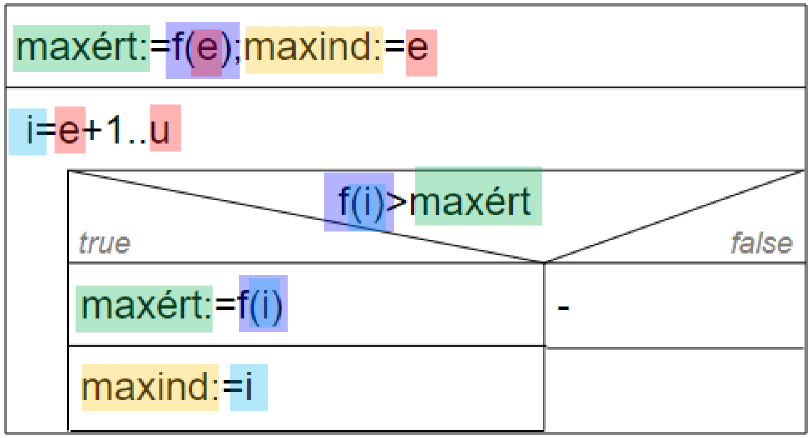
Specifikáció

```
Be: e∈Z, u∈Z
Ki: maxind∈Z, maxért∈H
Ef: e<=u
Uf: maxind∈[e..u] és
    ∀i∈[e..u]:(f(maxind)>=f(i)) és
    maxért=f(maxind)
```

Rövidítve:

```
Uf: (maxind,maxért)=MAX(i=e..u,f(i))
```

Algoritmus



Minimumkiválasztás
sablon

Feladat

Adott az egész számok egy [e..u] intervalluma és egy f:[e..u]→H függvény. A H halmaz elemein értelmezett egy teljes rendezési reláció. Határozzuk meg, hogy az f függvény hol veszi fel az [e..u] nem üres intervallumon a legkisebb értéket, és mondjuk meg, mekkora ez a minimális érték!

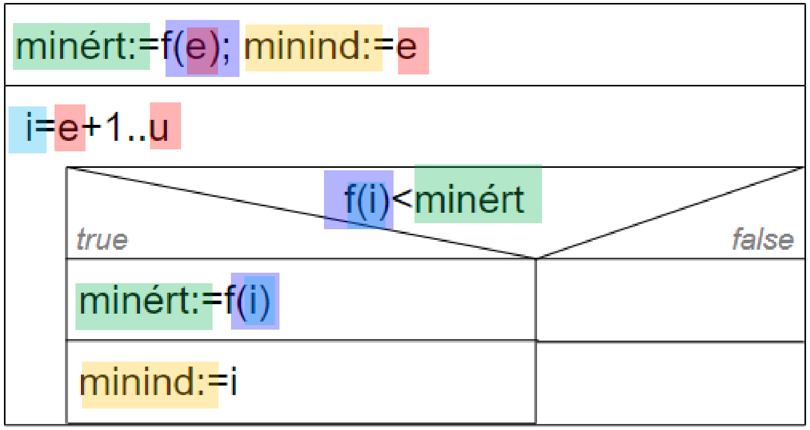
Specifikáció

```
Be: e∈Z, u∈Z
Ki: minind∈Z, minért∈H
Ef: e<=u
Uf: minind∈[e..u] és
    ∀i∈[e..u]:(f(minind)<=f(i)) és
    minért=f(minind)
```

Rövidítve:

```
Uf: (minind,minért)=MIN(i=e..u,f(i))
```

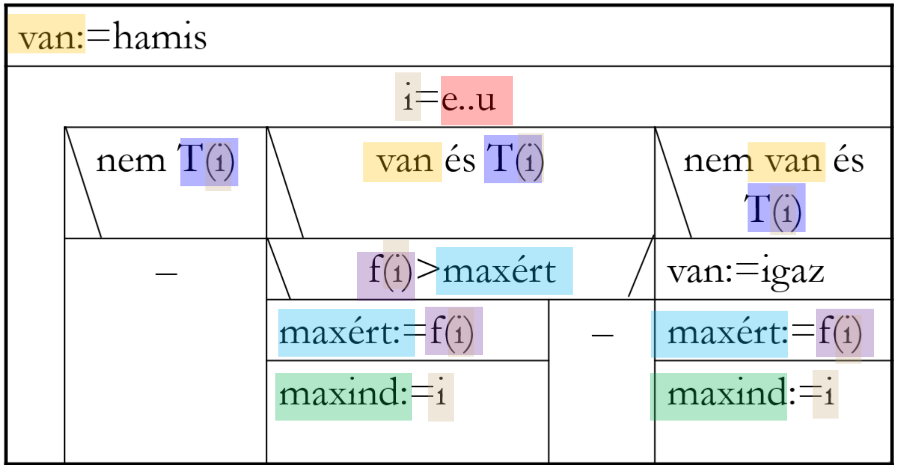
Algoritmus



Feltételes maximumkeresés
sablon

Feladat

Adott az egész számok egy [e..u] intervalluma, egy f:[e..u]→H függvény és egy T:[e..u]→Logikai feltétel. A H halmaz elemein értelmezett egy teljes rendezési reláció. Határozzuk meg, hogy az [e..u] intervallum T feltételt kielégítő elemei közül az f függvény hol veszi fel a legnagyobb értéket, és mondjuk meg, mekkora ez az érték!



Specifikáció és algoritmus:

```
Be: e∈Z, u∈Z
Ki: van∈L, maxind∈Z, maxért∈H
Ef: -
Uf: van = ∃i∈[e..u]:(T(i)) és
    van -> (maxind∈[e..u] és
            maxért=f(maxind) és T(maxind) és
            ∀i∈[e..u]:(T(maxind) -> maxért>=f(i)))
```

Rövidítve:

```
Uf: (van,maxind,maxért)=MAX(i=e..u,f(i),T(i))
```

Keresés
sablon

Feladat

Adott az egész számok egy [e..u] intervalluma és egy T:[e..u]→Logikai feltétel. Határozzuk meg az [e..u] intervallumban balról az első olyan számot, ha van, amely kielégíti a T feltételt!

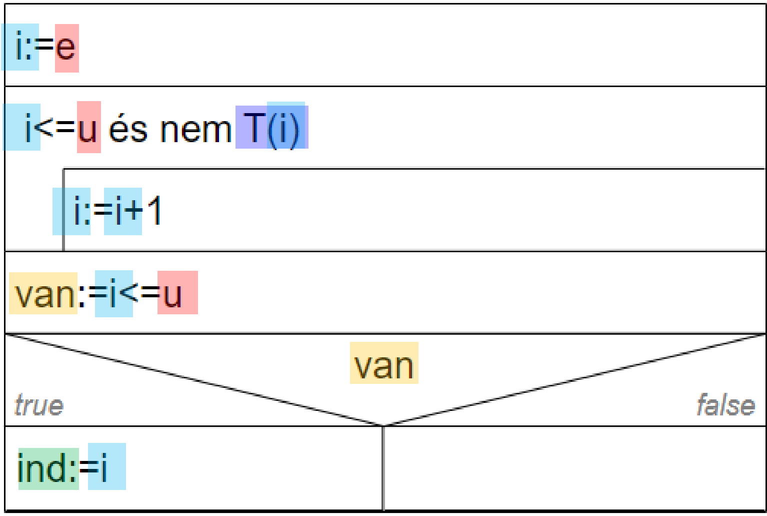
Specifikáció

```
Be: e∈Z, u∈Z
Ki: van∈L, ind∈Z
Ef: -
Uf: van=∃i∈[e..u]:(T(i)) és
    van->(ind∈[e..u] és T(ind) és
          ∀i∈[e..ind-1]:(nem T(i)))
```

Rövidítve:

```
Uf: (van,ind)=KERES(i=e..u,T(i))
```

Algoritmus



Keresés
sablon

Feladat

Adott az egész számok egy [e..u] intervalluma és egy T:[e..u]→Logikai feltétel. Határozzuk meg az [e..u] intervallumban balról az első olyan számot, ha van, amely kielégíti a T feltételt!

Specifikáció

```
Be: e∈Z, u∈Z
Ki: van∈L, ind∈Z
Ef: -
Uf: van=∃i∈[e..u]:(T(i)) és
    van->(ind∈[e..u] és T(ind) és
          ∀i∈[e..ind-1]:(nem T(i)))
```

Rövidítve:

```
Uf: (van,ind)=KERES(i=e..u,T(i))
```

Algoritmus

```
ind:=e
ind<=u és nem T(ind)
    ind:=ind+1
van:=ind<=u
```

Keresés
sablon

Feladat

Adott az egész számok egy [e..u] intervalluma és egy T:[e..u]→Logikai feltétel. Határozzuk meg az [e..u] intervallumban balról az első olyan számot, ha van, amely kielégíti a T feltételt!

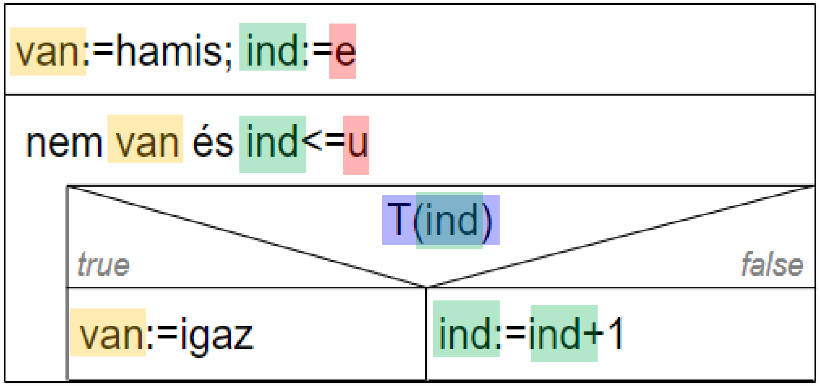
Specifikáció

```
Be: e∈Z, u∈Z
Ki: van∈L, ind∈Z
Ef: -
Uf: van=∃i∈[e..u]:(T(i)) és
    van->(ind∈[e..u] és T(ind) és
          ∀i∈[e..ind-1]:(nem T(i)))
```

Rövidítve:

```
Uf: (van,ind)=KERES(i=e..u,T(i))
```

Algoritmus



Eldöntés
sablon

Feladat

Adott az egész számok egy [e..u] intervalluma és egy T:[e..u]→Logikai feltétel. Határozzuk meg, hogy van-e az [e..u] intervallumnak olyan eleme, amely kielégíti a T feltételt!

Specifikáció

Be: e∈Z, u∈Z

Ki: van∈L

Ef: -

Uf: van=∃i∈[e..u]:(T(i))

Rövidítve:

Uf: van=**VAN**(i=e..u,T(i))

Algoritmus

i:=e
i<=u és nem T(i)
i:=i+1
van:=i<=u

Eldöntés
sablon

Feladat

Adott az egész számok egy [e..u] intervalluma és egy T:[e..u]→Logikai feltétel. Határozzuk meg, hogy van-e az [e..u] intervallumnak olyan eleme, amely kielégíti a T feltételt!

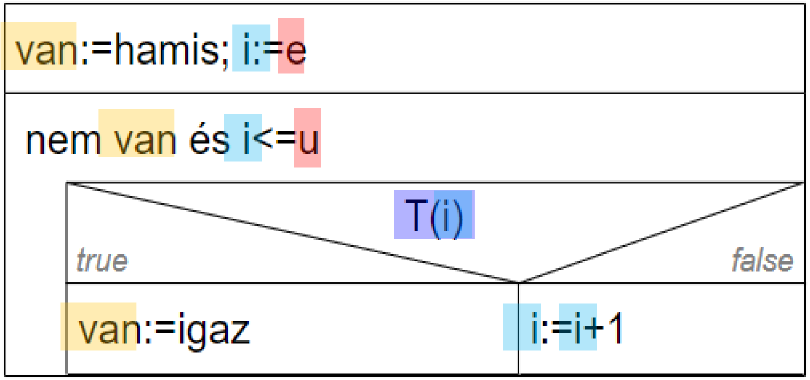
Specifikáció

```
Be: e∈Z, u∈Z
Ki: van∈L
Ef: -
Uf: van=∃i∈[e..u]:(T(i))
```

Rövidítve:

```
Uf: van=VAN(i=e..u,T(i))
```

Algoritmus



Mind eldöntés (vagy optimista eldöntés) sablon

Feladat

Adott az egész számok egy [e..u] intervalluma és egy T:[e..u]→Logikai feltétel. Határozzuk meg, hogy az [e..u] intervallumnak mindegyik eleme olyan-e, amely kielégíti a T feltételt!

Specifikáció

```
Be: e∈Z, u∈Z
Ki: mind∈L
Ef: -
Uf: mind=∀i∈[e..u]:(T(i))
```

Rövidítve:

```
Uf: mind=MIND(i=e..u,T(i))
```

Algoritmus

```
i:=e
i<=u és T(i)
    i:=i+1
mind:=i>u
```

Kiválasztás
sablon

Feladat

Adott egy e egész szám és egy e-től jobbra értelmezett T:Egész→Logikai feltétel. Határozzuk meg az e-től jobbra eső első olyan számot, amely kielégíti a T feltételt, ha tudjuk, hogy ilyen szám biztosan van!

Specifikáció

```
Be: e∈Z
Ki: ind∈Z
Ef: ∃i∈[e..∞]:(T(i))
Uf: ind>=e és T(ind) és
    ∀i∈[e..ind-1]:(nem T(i)))
```

Rövidítve:

```
Uf: ind=KIVÁLASZT(i>=e,T(i))
```

Algoritmus

```
i:=e
nem T(i)
    i:=i+1
ind:=i
```

```
ind:=e
nem T(ind)
    ind:=ind+1
```

Másolás
sablon

i		y
e	→	f(e)
e+1	→	f(e+1)
…	→	…
u	→	f(u)

Feladat

Adott az egész számok egy [e..u] intervalluma és egy f:[e..u]→H függvény. Rendeljük az [e..u] intervallum minden értékéhez az f függvény hozzá tartozó értékét!

Specifikáció

```
Be: e∈Z, u∈Z
Ki: y∈H[1..u-e+1]
Ef: -
Uf: ∀i∈[e..u]:(y[i-e+1]=f(i))
```

Rövidítve:

Uf: y=**MÁSOL**(i=e..u, f(i))

Algoritmus

```
i=e..u
    y[i-e+1]:=f(i)
```

Kiválogatás
sablon

Feladat

Adott az egész számok egy [e..u] intervalluma, egy ezen értelmezett T:[e..u]→Logikai feltétel és egy f:[e..u]→H függvény. Határozzuk meg az f függvény az [e..u] intervallum azon értékeinél felvett értékeit, amelyekre a T feltétel teljesül!

Specifikáció

```
Be: e∈Z, u∈Z
Ki: db∈N, y∈H[1..db]
Ef: -
Uf: db=DARAB(i=e..u,T(i)) és
    ∀i∈[1..db]:(
      ∃j∈[e..u]:T(j) és y[i]=f(j))
 és y⊆(f(e),f(e+1),...,f(u))
```

Rövidítve:

```
Uf: (db,y)=KIVÁLOGAT(i=e..u,T(i),f(i))
```

Algoritmus