

Programozási nyelvek I. 3. gyakorlat

Balogh Ádám
bas@elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

2. házi feladat megoldása

```
maximum.adb:
...
procedure Maximum is
  function Max ( A, B : Positive ) ...
  function Max ( A, B, C : Positive ) ...
  ...
begin
  A := Max ( 4, 6 );
  B := Max ( 4, 6, 12 );
  ...
end Maximum;
```

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

2

3. házi feladat megoldása

```
szoroz.adb:
with Text_IO, Ada.Command_Line;
...
begin
  if Argument_Count < 2 then
    Put_Line ( "Kevés argumentum!" );
  else
    Put_Line ( Natural'Image (
      Natural'Value ( Argument ( 1 ) ) *
      Natural'Value ( Argument ( 2 ) ) ));
  end if;
end Szoroz;
```

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

3

Segítség az 1. feladathoz

- Altípus definiálása:
 subtype *Név* is *Konstrukció*;
- Csak az eredeti típust módosító konstrukciók használhatók
- Egyirányú kompatibilitás az eredeti típussal
- Diszkrét típus megszorítása intervallumra
 Típus range *Eleje* .. *Vége*

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

4

1. feladat

Módosítsd az előző gyakorlaton írt *Naptar* programot úgy, hogy a *Positive* típus helyett annak egy megszorított altípusát használja, amely csak az 1 és 7 közötti számokat tartalmazza. Az új program neve legyen *Naptar2*!

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

5

1. feladat megoldása

```
naptar2.adb:
...
subtype Nap_Tipus is
  Positive range 1 .. 7;
Nap : Nap_Tipus;
...
Nap := Positive'Value (
  Szam ( 1 .. Szam_Hossza ) );
vagy
Nap := Nap_Tipus'Value (
  Szam ( 1 .. Szam_Hossza ) );
...
```

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

6

Segítség a 2. feladathoz

- Származtatott típus definiálása:
`type Név is new Konstrukció;`
- Csak az eredeti típust módosító konstrukciók használhatók, ugyanazok, mint altípus esetén
- Teljesen inkompatibilis az eredeti típussal

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

7

2. feladat

Módosítsd az előbb megírt `Naptar2` programot úgy, hogy a `Positive` típus helyett egy belőle származtatott típust használjon, amely csak az 1 és 7 közötti számokat tartalmazza. Az új program neve legyen `Naptar3`!

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

8

2. feladat megoldása

```
naptar3.adb:
...
type Nap_Tipus is new
    Positive range 1 .. 7;

    Nap : Nap_Tipus;
...
    Nap := Nap_Tipus'Value (
        Szam ( 1 .. Szam_Hossza ));
...
```

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

9

Segítség a 3. feladathoz

- Felsorolási típus:
`(Azonosító, Azonosító, ...)`
- A diszkrét típusok minden attribútuma alkalmazható rá

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

10

3. feladat

Módosítsd az előbb megírt `Naptar3` programot úgy, `Positive`-ből származtatott típus helyett felsorolási használjon, amely a napok neveit tartalmazza. Az új program neve legyen `Naptar4`!

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

11

3. feladat megoldása

```
naptar4.adb:
...
type Nap_Tipus is ( Hetfo, Kedd,
    Szerda, Csutortok, Pentek,
    Szombat, Vasarnap );
...
case Nap is
    when Hetfo => ...
    when Kedd => ...
    ...
end case;
...
```

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

12

Segítség a 4. feladathoz

- Vektor típus konstrukciója:

```
type Név is array ( Eleje .. Vége )
  of Elemtípus;
```
- Teljes vektor megadása (pl. mint kezdőérték):

```
( Kifejezés, Kifejezés, ... )
```
- Vektor egy eleme: `Vektor (Index)`
- Egész számok típusa: `Integer`

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

13

4. feladat

Írj programot, ami megkeresi egy 1-től 10-ig indexelt tömb elemeinek maximumát! A maximum helyét (indexét) a főprogramba ágyazott `Max` függvény adja vissza! A (konstans) tömb elemei legyenek 2, 6, 5, -9, 1, 56, 86, 23, -71 és 4. A program neve legyen `Max_Ker`!

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

14

4. feladat megoldása (1)

```
max_ker.adb:
...
type Int_Vektor is array ( 1 .. 10 ) of
  Integer;

IV : constant Int_Vektor := ( 2, 6, 5,
  - 9, 1, 56, 86, 23, - 71, 4 );

function Max_Hely ( V : Int_Vektor )
  return Positive is;
  MH : Positive := 1;
(folyt. köv.)
```

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

15

4. feladat megoldása (2)

```
max_ker.adb:
(folyt.)
begin
  for I in 1 .. 10 loop
    if V ( MH ) < V ( I ) then
      MH := I;
    end if;
  end loop;
  return MH;
end Max_Hely;
...
```

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

16

Segítség az 5. feladathoz

- Vektor típus konstrukciója egy típus teljes intervallumára:

```
type Név is array ( Indextípus )
  of Elemtípus;
```
- Teljes vektor megadása indexszel jelölt formában (pl. mint kezdőérték):

```
( Indexhalmaz => Kifejezés,
  Indexhalmaz => Kifejezés, ... )
```
- Indexhalmaz: mint érték-halmaz a `case`-nél

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

17

5. feladat

Alakítsd át az előző programot úgy, hogy a tömb indexeinek típusa eleve csak az 1 és 10 közötti számokat tartalmazza (altípus vagy származtatott típus)! A tömb elemei legyenek 2, 2, 0, 9, 9, 9, 4, 13, 4, 2, amit indexszel jelölt formában adj meg (még hozzá úgy, hogy egy érték csak egyszer szerepeljen)! Az új program neve legyen `Max_Ker2`!

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

18

5. feladat megoldása

```
max_ker2.adb:
...
type Index is new Integer range
                    1 .. 10;
type Int_Vektor is array ( 1 .. 10 ) of
    Integer;

IV : constant Int_Vektor := ( 3 => 0,
    4 .. 6 => 9, 8 => 13,
    7 | 9 => 4, 1 | 2 | 10 => 2 );
...
2004. szeptember 30.      Programozási nyelvek I. - 3. gyakorlat      19
```

Segítség a 6. feladathoz (1)

- Tömb típus indexeinek intervallumát nyitva lehet hagyni, csak a típusát kell rögzíteni:
`type Név is array (Indextípus range <>) of Elemtípus;`
- Konkrét intervallum rögzítése változó deklarációjakor:
`Változó : Tömbtípus (Intervallum);`

2004. szeptember 30. Programozási nyelvek I. - 3. gyakorlat 20

Segítség a 6. feladathoz (2)

- Tömb indextípusának `First` és `Last` attribútumai a tömbnek is attribútumai
- `X'First .. X'Last = X'Range`
- Teljes tömb megadásakor vegyes forma is használható (mint eljárás vagy függvényhívásnál paraméterátadásra)
- Itt is lehet `others`, mint a `case` ágainál

2004. szeptember 30. Programozási nyelvek I. - 3. gyakorlat 21

6. feladat

Módosítsd az első `Max_Ker` programot úgy, hogy a tömb indexeinek intervallumát csak a konstans deklarációjakor rögzíted! A tömb elemei most 2, 6, 4, 4, 4, 4, 4, 4 és 4 legyenek, amit vegyes formában adj meg, és használd az `others`-et is! Az új program neve legyen `Max_Ker3`!

2004. szeptember 30. Programozási nyelvek I. - 3. gyakorlat 22

6. feladat megoldása (1)

```
max_ker3.adb:
...
type Int_Vektor is array
    ( Integer range <> ) of Integer;

IV : constant Int_Vektor ( 1 .. 10 ) :=
    ( 2, 6, others => 4 );

function Max_Hely ( V : Int_Vektor )
    return Integer is
    MH : Integer := V'First;
(folyt. köv.)
2004. szeptember 30.      Programozási nyelvek I. - 3. gyakorlat      23
```

6. feladat megoldása (2)

```
max_ker3.adb:
(folyt.)
begin
    for I in V'Range loop
        if V ( MH ) < V ( I ) then
            MH := I;
        end if;
    end loop;
    return MH;
end Max_Hely;
...
2004. szeptember 30.      Programozási nyelvek I. - 3. gyakorlat      24
```

Segítség a 7. feladathoz

- Nyitott index-intervallumú tömb típusú változók intervallumát a kezdőértékadás is rögzítheti (ha nincs benne `others`)
- Attribútumok segítségével lekérdezhető az ilyen módon automatikusan meghatározott intervallum

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

25

7. feladat

Írd át a maximumkeresést (`Max_Ker`) úgy, hogy a tömb index-intervalluma nyitott legyen, mint az előző feladatnál, azonban most csak a változó kezdőértéke rögzítse! Az új program neve legyen `Max_Ker4`!

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

26

7. feladat megoldása

```
max_ker4.adb:
...
  type Int_Vektor is array
    ( Integer range <> ) of Integer;

  IV : constant Int_Vektor := ( 2, 6, 5,
    - 9, 1, 56, 86, 23, - 71, 4 );
...
```

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

27

Házi feladatok

- A legutolsó program `Max_Hely` függvényének segítségével írd egy rendező programot! A rendezés is a főprogramnak egy külön eljárása legyen! A program neve legyen `Rendezes`!
- Módosítsd az előző programot úgy, hogy ciklus helyett rekurziót használjon a rendezéshez! A program neve legyen `Rendezes2`!

2004. szeptember 30.

Programozási nyelvek I. - 3. gyakorlat

28