

## Programozási nyelvek I. 4. gyakorlat

Balogh Ádám  
bas@elte.hu

Eötvös Loránd Tudományegyetem  
Informatikai Kar

### 1. házi feladat megoldása (1)

```
rendezes.adb:
...
procedure Rendezes is
  function Max_Hely ( V : Int_Vektor )
    return Index is ...
  procedure Cserel ( A, B :
    in out Integer ) is ...
  procedure Rendez ( V :
    in out Int_Vektor ) is
    M : Integer;
  begin
    (folyt. köv.)
```

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

2

### 1. házi feladat megoldása (2)

```
rendezes.adb:
(folyt.)
  for I in reverse V'Range loop
    M := Max_Hely ( V'First .. I );
    Cserel ( V ( I ), V ( M ) );
  end loop;

begin
  Rendez ( IV );
  ...
end Rendezes;
```

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

3

### 2. házi feladat megoldása

```
rendezes2.adb:
...
procedure Rendez ( V : in out Int_Vektor ) is
  M : Index;
begin
  if V'Length > 1 then
    M := Max_Hely ( V );
    Cserel ( V ( V'Last ), V ( M ) );
    Rendez ( V ( V'First .. V'Last - 1 ) );
  end if;
end Rendez;
...
```

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

4

### Segítség az 1. feladathoz (1)

- Csomag specifikációja:  
package Név is  
  Deklarációk  
end Név;
- Állomány neve: *név.ads*
- Csomag implementációja:  
package body Név is  
  Definíciók  
end Név;
- Állomány neve: *név.adb*

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

5

### Segítség az 1. feladathoz (2)

- Operátorok átlapolása: függvény neve  
"operátor" legyen, pl. "+"
- Lebegőpontos számok típusa:  
Float digits *Pontosság*
- Többdimenziós tömb:  
array ( *Intervallum*, *Intervallum*,  
  ... ) of *Típus*
- Többdimenziós tömb attribútumai:  
*Attribútum* ( *Dimenzió* )

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

6

## 1. feladat

Írj egy `Matrixok` nevű csomagot, amely a mátrixok típusát valósítja meg! A mátrixra legyen értelmezve az összeadás és a szorzás művelete, a szokásos műveleti jelekkel. A műveletek megvalósításánál feltételezhetjük, hogy az operandus mátrixok méretei megfelelőek a művelet elvégzéséhez. A főprogramot (`Matrixok_Proba`) töltsd le a gyakorlat honlapjáról!

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

7

## 1. feladat megoldása (1)

```
matrixok.ads:
package Matrixok is
  type Index is new Integer;
  type Ertek is new Float digits 5;
  type Matrix is array ( Index range <>,
    Index range <> ) of Ertek;
  function "+" ( A, B : Matrix )
    return Matrix is ...
  function "*" ( A, B : Matrix )
    return Matrix is ...
...

```

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

8

## 1. feladat megoldása (2)

```
matrixok.adb:
package body Matrixok is
  function "+" ( A, B : Matrix )
    return Matrix is
    ...
  end "+";

  function "*" ( A, B : Matrix )
    return Matrix is
    ...
  end "*";
end Matrixok;

```

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

9

## Segítség a 2. feladathoz

- Rekord típus:
 

```
type Név is record
  Mezők
end record;
```
- Mezők: mint változódeklarációk
- Átlátszatlan típus: `private`
- Átlátszatlan típus kifejtése: csomag specifikációjának privát részében, amit a `private` kulcsszó vezet be

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

10

## 2. feladat

Valósítsd meg a pozitív racionális számok típusát a `Poz_Rac` csomagban, átlátszatlan típusként! A típus reprezentációja egy számláló és nevező mezőket tartalmazó rekord legyen. Műveletek: két pozitív osztása, két pozitív racionális osztása, pozitív racionális osztása pozitívval, valamint számláló és nevező lekérdezése. A főprogramot (`Poz_Rac_Proba`) töltsd le a gyakorlat honlapjáról!

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

11

## 2. feladat megoldása

```
poz_rac.ads:
package Poz_Rac is
  type Pozrac is private;

  function "/" ( A, B : Positive )
    return Pozrac is ...
...
private
  type Pozrac is record
    Szamlalo, Nevezo : Positive;
  end record;
end Poz_Rac;

```

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

12

### Segítség a 3. feladathoz

- Átlátszatlan típusokra a csomagban definiált műveleteken kívül értelmezve van az értékadás és az összehasonlítás is
- Korlátozott átlátszatlan típus (*limited private*): még ez a két művelet sincs értelmezve rá

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

13

### 3. feladat

Valósítsd meg a verem típust korlátozott átlátszatlan típusként a *Verem\_V* csomagban. A műveletek legyenek: elem behelyezése, legfelső elem lekérdezése illetve törlése, valamint annak vizsgálata, hogy üres-e illetve tele van-e a verem. A főprogramot (*Verem\_V\_Proba*) töltsd le a gyakorlat honlapjáról!

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

14

### 3. feladat megoldása (1)

```
verem_v.ads:
package Verem_V is
  Max_Meret : constant Positive := 100;

  subtype Elem is Integer;

  type Verem is limited private;
  ...
  (folyt.köv.)
```

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

15

### 3. feladat megoldása (2)

```
verem_v.ads:
(folyt.)
private
  type Vektor is array ( 1 .. Max_Meret )
    of Elem;

  type Verem is record
    Mutato : Integer := 0;
    Tartalom : Vektor;
  end record;
end Verem_V;
```

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

16

### Segítség a 4. feladathoz (1)

- Diszkriminánsos rekord típus:
 

```
type Név ( Diszkrimináns ) is record
  Mezők
end record;
```
- A *Diszkrimináns* szintaxisa olyan, mint egy függvény paraméteréé
- Mezőkben paraméterként használhatjuk a diszkriminánsot
- Átlátszatlan típusnál is szerepelnie kell a diszkriminánsnak, nemcsak a kifejtésénél

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

17

### Segítség a 4. feladathoz (2)

- Diszkrimináns a rekord egyik mezője, de értéke csak a teljes rekordnak való értékadáskor változtatható
- Diszkrimináns alapértelmezett értéke: ha a diszkriminánsot tömb méretére használjuk, és meghagyjuk az alapértelmezett értéket, akkor az indextípus által megengedett maximális méretű terület foglalódik le, de csak az alapértelmezett értéknyit lehet használni, amíg az egész új értéket nem kap

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

18

## 4. feladat

Módosítsd a vermet úgy, hogy méretét diszkrimináns adja meg! A diszkrimináns alapértelmezett értéke 100 legyen, és 1000-nél többet ne is lehessen megadni, ennyi foglalódjon le, ha az alapértelmezett értéket használjuk. A csomag neve legyen Verem\_D!

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

19

## 4. feladat megoldása (1)

```
verem_d.ads:
package Verem_D is
  subtype Meret is Natural range
                                1 .. 1000;
  subtype Elem is Integer;

  type Verem ( Max_Meret : Meret ) is
    limited private;
  ...
(folyt.köv.)
```

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

20

## 4. feladat megoldása (2)

```
verem_d.ads:
(folyt.)
private
  type Vektor is array
    ( Natural range <> ) of Elem;

  type Verem ( Max_Meret : Natural )
    is record
    Mutato : Integer := 0;
    Tartalom : Vektor ( 1 .. Max_Meret );
  end record;
end Verem_D;
```

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

21

## Házi feladatok

- Egészítsd ki a pozitív racionális típust új műveletekkel: összeadás, kivonás, szorzás, valamint az összes művelet legyen végrehajtható pozitív racionális és pozitív között (pozitív racionális eredményt adva)!
- Valósítsd meg a sor típust a (továbbfejlesztett) verem típushoz hasonlóan!

2004. október 7.

Programozási nyelvek I. - 4. gyakorlat

22