

## Programozási nyelvek I. 7. gyakorlat

Balogh Ádám  
bas@elte.hu

Eötvös Loránd Tudományegyetem  
Informatikai Kar

### Segítség az 1. feladathoz

- Adott típusra mutató típus: `access Típus`
- Semmire sem mutató: `null`
- Mutatott objektum: `Mutató.all`
- Rekordnál `Mutató.all.Mező` helyett `Mutató.Mező` írható
- Objektum létrehozása dinamikusan:  
`Mutató := new Típus' ( Érték )`

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

2

### 1. feladat

Valósítsd meg a vermet láncolt listás ábrázolással! A műveletek legyenek ugyanazok, mint a vektorban ábrázolt veremnél, kivéve a `Tele` műveletet! Az új csomag neve legyen `Verem_LGE`! A kipróbáló programot a `Verem_DGE_Proba` módosításával hozd létre!

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

3

### 1. feladat megoldása (1)

```
verem_lge.ads:
...
private
  type Doboz;
  type Doboz_Mutato is access Doboz;
  type Doboz is
    record
      Adat : Elem;
      Kovetkezo : Doboz.Mut := 0;
    end record;
(folyt.köv.)
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

4

### 1. feladat megoldása (2)

```
verem_lge.ads:
(folyt.)
type Verem is
  record
    Teto : Doboz_Mutato := null;
  end record;
...
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

5

### 1. feladat megoldása (3)

```
verem_lge.adb:
...
procedure Beteves ( V : in out Verem;
  E : in Elem ) is
begin
  V.Teto := new Doboz' ( E, V.Teto );
end Beteves;
(folyt.köv.)
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

6

## 1. feladat megoldása (4)

```
verem_lge.adb:
(folyt.)
  procedure Kidobas ( V : in out Verem )
                                is
  begin
    V.Teto := V.Teto.Kovetkezo;
  exception
    when Constraint_Error =>
      raise Ures_Verem;
  end Kidobas;
(folyt.köv.)
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

7

## 1. feladat megoldása (5)

```
verem_lge.adb:
(folyt.)
  function Teto ( V : Verem ) return Elem
                                is
  begin
    return V.Teto.Adat;
  exception
    when Constraint_Error =>
      raise Ures_Verem;
  end Teto;
(folyt.köv.)
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

8

## 1. feladat megoldása (6)

```
verem_lge.adb:
(folyt.)
  function Ures ( V : Verem )
                                return Boolean is
  begin
    return ( V.Teto = null );
  end Ures;
...
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

9

## Segítség a 2. feladathoz (1)

- Egész számok be- és kivitele:  
Text\_IO.Integer\_IO sablon csomag
- Put-nak és Get-nek van ebben a csomagban is megfelelő típusú változata
- Állományleíró típus: File\_Type
- Új szöveges állomány létrehozása:  

```
procedure Create (
  File : in out File_Type;
  Mode : in File_Mode := Out_File;
  Name : in String := "" );
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

10

## Segítség a 2. feladathoz (2)

- Szöveges állomány létrehozása:  

```
procedure Open (
  File : in out File_Type;
  Mode : in File_Mode;
  Name : in String );
```
- Megnyitott állomány lezárása:  

```
procedure Close (
  File : in out File_Type );
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

11

## Segítség a 2. feladathoz (3)

- Írás és olvasás: Get, Put, Get\_Line, Put\_Line, New\_Line, Skip\_Line; első paraméterük az állomány leírója
- Állomány végének lekérdezése:  

```
function End_Of_File (
  F : File_Type ) return Boolean;
```
- Sor vége hasonlóan End\_Of\_Line
- Mindkettő működik billentyűzetre is, ekkor nincs paramétere

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

12

## 2. feladat

- Írj programot, amely létrehoz egy szöveges állományt, és a billentyűzetről olvasott számokat abba beleírja. Egy másik program olvassa ki az adatokat az állományból, és írja azokat a képernyőre. Ügyelj rá mindkét programnál, hogy egy sorba ugyanannyi szám kerüljön, mint amennyi eredetileg volt. A programok neve legyen *ír* és *olvas*!

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

13

## 2. feladat megoldása (1)

```
ir.ads:
...
package Int_IO is
    new Integer_IO ( Integer );
...
Create ( F, Name => "szamok.txt" );
while not End_Of_File loop
    while not End_Of_Line loop
        Get ( I );
        Put ( F, I );
    end loop;
end loop;
(folyt.köv.)
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

14

## 2. feladat megoldása (2)

```
ir.ads:
(folyt.)
end loop;
Skip_Line;
New_Line ( F );
end loop;
Close ( F );
end Ir;
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

15

## 2. feladat megoldása (3)

```
olvas.ads:
...
package Int_IO is
    new Integer_IO ( Integer );
...
Open ( F, In_File, "szamok.txt" );
while not End_Of_File ( F ) loop
    while not End_Of_Line ( F ) loop
        Get ( F, I );
        Put ( I );
    end loop;
end loop;
(folyt.köv.)
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

16

## 2. feladat megoldása (4)

```
olvas.ads:
(folyt.)
end loop;
Skip_Line ( F );
New_Line;
end loop;
Close ( F );
end Olvas;
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

17

## Segítség a 3. feladathoz

- Bináris állományok kezelése sorosan: Sequential\_IO sablon csomag
- Új létrehozása, létező megnyitása ugyanúgy, mint szöveges állományoknál
- Adatok olvasása:
 

```
procedure Read ( File : in File_Type;
                  Item : out Típus );
```
- Adatok írása:
 

```
procedure Write ( File : in File_Type;
                  Item : out Típus );
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

18

### 3. feladat

- Írj programot, amely létrehoz egy bináris állományt, és a billentyűzetről olvasott számokat abba beleírja. Egy másik program olvassa ki az adatokat az állományból, és írja azokat a képernyőre. A kiírásnál a program ötösével írja a számokat egy sorba! A programok neve legyen `Ír2` és `Olvas2`!

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

19

### 3. feladat megoldása (1)

```
ir.ads:
...
package Int_Text_IO is
    new Integer_IO ( Integer );
package Int_Bin_IO is
    new Integer_IO ( Integer );
...
F : Int_Bin_IO.File_Type;
...
Create ( F, Name => "szamok.dat" );
(folyt.köv.)
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

20

### 3. feladat megoldása (2)

```
ir.ads:
(folyt.)
while not End_Of_File loop
    while not End_Of_Line loop
        Get ( I );
        Write ( I, F );
    end loop;
    Skip_Line;
end loop;
Close ( F );
end Ir2;
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

21

### 3. feladat megoldása (3)

```
olvas.ads:
...
package Int_Text_IO is
    new Integer_IO ( Integer );
package Int_Bin_IO is
    new Integer_IO ( Integer );
...
F : Int_Bin_IO.File_Type;
...
Open ( F, In_File, "szamok.dat" );
(folyt.köv.)
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

22

### 3. feladat megoldása (4)

```
ir.ads:
(folyt.)
while not End_Of_File loop
    Get ( I );
    Write ( I, F );
    J := J + 1;
    if J = 5 then
        New_Line;
        J := 0;
    end if;
...
...
...
(folyt.köv.)
```

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

23

### Házi feladatok

- Készítsd el a két „végleges” `Sor` típust megvalósító sablon csomagot is, a `Sor_DGE`-t és a `Sor_LGE`-t!
- Írj programot, amely nem sorosan, hanem közvetlen eléréssel kezel egy állományt! Segítség: könyv 10.2.2. fejezet illetve A.5. függelék.

2004. november 11.

Programozási nyelvek I. - 7. gyakorlat

24