

## 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

### 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

---

---

---

---

---

---

---

---

## 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

---

---

---

---

---

---

---

---

## 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

---

---

---

---

---

---

---