Programozási nyelvek I.
3. gyakorlat

Balogh Ádám
bas@elte.hu
Eötvös Loránd Tudományegyetem
Informatikai Kar

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`*

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`!

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

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`!

Segítség a 3. feladathoz

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

3. feladat

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

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`

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`!

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):

```
( Indexhelmaz => Kifejezés,
  Indexhelmaz => Kifejezés, ... )
```

- Indexhalmaz: mint értékhalmaz a `case`-nél

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 (méghozzá úgy, hogy egy érték csak egyszer szerepeljen)! Az új program neve legyen `Max_Ker2`!

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 );
```

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

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, 4 és 4 legyenek, amit vegyes formában adj meg, és használd az `others`-et is! Az új program neve legyen `Max_Ker3`!

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

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`!

Házi feladatok

- A legutolsó program `Max_Hely` függvényének segítségével írj 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`!