Programozási nyelvek I.
8. gyakorlat

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

Segítség az 1. feladathoz (1)

- Egyszerű taszk objektum specifikációja:
```
task Név;
```
- Törzse:
```
task body Név is
   Változódeklarációk
begin
   Utasítások
end Név;
```
- A taszk nem önálló fordítási egység, hanem ugyanabba a fordítási egységbe kell beágyazni a specifikációt és a törzset

Segítség az 1. feladathoz (2)

- Ha a törzs túl nagy, akkor elhelyezhető külön állományban is (de ekkor sem önálló fordítási egység)
- Ekkor a taszk törzse helyén:
```
task body Név is separate;
```
- Törzset tartalmazó állomány neve:
```
Fordítási egység neve-Taszk neve.adb
```
- Törzset tartalmazó állomány első sora:
```
separate ( Fordítási egység neve );
```
- Taszk párhuzamosan fut a főprogrammal

1. feladat

Írj egy üres programot, amelyben a főprogrammal párhuzamosan egy üres taszk is lefut. A program neve legyen `Semmi_T`!

2. feladat

Írj programot, amelyben két taszk (`Jancsi` és `Julcsa`) egymással párhuzamosan felsorolja az „összes” pozitív számot, a főprogram pedig üres. A taszkok minden egyes szám elé írják ki a nevüket is, az azonosíthatóság érdekében. A program neve legyen `Jancsi_Julcsa`! Figyeld meg a futást!

Segítség az 3. feladathoz (1)

- Taszk típusnál specifikációban: `task` helyett `task type`
- Taszk objektumok ezután valamilyen taszk típusú változók
- Belépési pontok

`task` vagy `task type` *Taszk* `is`

```
   entry Név ( Paraméterek );
   entry Név ( Paraméterek );
   ...
end Taszk;
```

Segítség az 3. feladathoz (2)

- Paraméterek hasonlóak az eljárásoknál megismertekhez
- Randevú: taszk törzsében:

```
Utasítások
accept Név ( Paraméterek ) do
   Utasítások
end Név;
Utasítások
```

- Minden belépési ponthoz pontosan egy randevú tartozik, nevük is megegyezik

2004. november 18. Programozási nyelvek I. - 8. gyakorlat 9

Segítség az 3. feladathoz (3)

- Randevú akkor jön létre, amikor a főprogram vagy egy másik taszk hívja valamelyik belépési pontot:

```
Taszknév.Pontnév ( Paraméterek );
```

- Híváskor a hívó taszk felfüggesztődik, amíg a randevún belüli utasítások végre nem hajtódnak
- A randevú mérete mindig a lehető legkisebb legyen!

2004. november 18. Programozási nyelvek I. - 8. gyakorlat 10

3. feladat

Írj az előzőhöz hasonló programot úgy, hogy a két taszk egy közös típushoz tartozik, és nevüket egy belépési ponton keresztül kapják meg. Mivel a nevek hosszát előre nem tudjuk, a taszk lokális változója karakterláncra mutató típusú legyen, és névadáskor dinamikusan kerüljön lefogalalásra a szükséges terület. A két taszk ismét `Jancsi` és `Julcsa` legyen, az egész programé pedig `Jancsi_Julcsa2`!

2004. november 18. Programozási nyelvek I. - 8. gyakorlat 11

Segítség a 4. feladathoz

- Taszk típus diszkriminánssal: specifikáció:
  `task type Név ( Diszkrimináns ) ...`
- Taszk diszkriminánsa hasonló egy rekord diszkriminánsához, típusa csak diszkrét típus vagy mutató típus lehet
- Időhöz kötött várakoztatás:
  `delay` *`Idő`*`;`
- Az *`Idő`* `Duration` típusú (skalár típus)

4. feladat

Módosítsd az előző programot úgy, hogy a taszk neve a diszkriminánsa legyen, alapértelmezett értékkel. Továbbá, ha az első parancssori argumentumban meg van adva egy idő, akkor ennyit várjon a taszk két szám kiírása között. (Feltehetjük, hogy ha van parancssori argumentum, akkor az idő.) A program neve legyen `Jancsi_Julcsa3`!