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`!

1. feladat megoldása

semmi.adb:

```
procedure Semmi_T is
   task T;

   task body T is
   begin
      null;
   end T;
begin
   null;
end 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!

2. feladat megoldása

jancsi_julcsa.adb:

```
...
   task Jancsi;
...
   task body Jancsi is
   begin
      for I in Positive'Range loop
         Put_Line ( "Jancsi: " &
            Positive'Image ( I ));
      end loop;
   end Jancsi;
...
```


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

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!

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 lefoglalásra a szükséges terület. A két taszk ismét `Jancsi` és `Julcsa` legyen, az egész programé pedig `Jancsi_Julcsa2`!

3. feladat megoldása (1)

jancsi_julcsa2.adb:

```
...
task type Kiiro is
   entry Init ( S : in String := "" );
end Kiiro;

task body Kiiro is
   type String_Mut is access String;
   Nev : String_Mut;
begin
   accept Init ( S : in String := "" ) do
(folyt. köv.)
```

3. feladat megoldása (2)

jancsi_julcsa2.adb:

```
(folyt.)
      Nev := new String' ( S );
   end Init;
   for I in Positive'Range loop
      Put_Line ( Nev.all & ": " & ...
   ...
   A, B : Kiiro;
begin
   A.Init ( "Jancsi" );
   B.Init ( "Julcsa" );
end Jancsi_Julcsa2;
```


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`!

4. feladat megoldása (1)

jancsi_julcsa3.adb:

```
...
type String_Mut is access String;
task type Kiiro ( Nev : String_Mut :=
                    new String' ( "" ));

task body Kiiro is
begin
   for I in Positive'Range loop
      Put_Line ( Nev.all & ": " & ...
      if Argument_Count > 0 then
(folyt. köv.)
```


4. feladat megoldása (2)

jancsi_julcsa3.adb:

```
(folyt.)
            delay Duration'Value (
                     Argument ( 1 ));
         end if;
   ...
   Jancsi : Kiiro (
               new String' ( "Jancsi" ));
   Julcsa : Kiiro (
               new String' ( "Julcsa" ));
...
```