

Programozási nyelvek I. 6. gyakorlat

Balogh Ádám
bas@elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

Segítség az 1. feladathoz

- Blokk utasítás:

```
[declare
  Deklarációk, definíciók]
begin
  Utasítások
end;
```
- Tipikus használati területei:
 - Deklaráció vagy definíció valamely paraméterét programkód határozza meg
 - Kivételkezelés (ld. később)

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

2

1. feladat

Egészítsd ki a `Verem_DG_Proba` programot úgy, hogy a verem méretét a felhasználó adhassa meg billentyűzetről! Az új program neve legyen `Verem_DG_Proba2`!

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

3

1. feladat megoldása

```
verem_dg_proba2.adb:
...
Get_Line ( Melyseg, Melyseg_Hossz );
declare
  V : Verem ( Meret'Value (
    Melyseg ( 1 .. Melyseg_Hossz )) );
begin
  ...
end;
...
```

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

4

Segítség a 2. feladathoz (1)

- Kivételek kezelése: begin-end blokkban:


```
begin
  Utasítások
exception
  when Kivétel => Utasítások
  when Kivétel => Utasítások
  ...
end;
```
- Általános határsértési kivétel:


```
Constraint_Error
```

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

5

Segítség a 2. feladathoz (2)

- Kezeletlen kivétel tovább terjed az utasításblokkot tartalmazó blokk felé
- Alprogram fő utasításblokkjától a hívó felé terjed tovább, és a hívás helyén jelenik meg
- Ha a főprogram fő utasításblokkjában nincs lekezelve, akkor a program hibaüzenettel leáll

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

6

2. feladat

Egészítsd ki a `Verem_DG_Proba2` programot úgy, hogy kezelje azt a kivételt, amely akkor keletkezik, amikor üres veremből próbálunk kivenni, illetve tele verembe próbálunk betenni! Az új program neve legyen `Verem_DG_Proba3`!

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

7

2. feladat megoldása (1)

```
verem_dg_proba2.adb:
...
begin
    Put_Line ( "A verem teteje: " &
               Integer'Image ( Teto ( V ) ) );
exception
    when Constraint_Error =>
        Put_Line ( "A verem üres!" );
end;
(folyt. köv.)
```

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

8

2. feladat megoldása (2)

```
verem_dg_proba2.adb:
(folyt.)
begin
    Betevés ( V, I );
    Put_Line ( "Betéve: " &
               Integer'Image ( I ) );
exception
    when Constraint_Error =>
        Put_Line ( "A verem tele van!" );
end;
...
```

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

9

Segítség a 3. feladathoz

- Saját kivétel deklarálása:
Név : exception;
- Kivétel kiváltása:
raise Név;
- Csomagoknak illik saját kivételeket kiváltaniuk a beépítettek helyett

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

10

3. feladat

Módosítsd a `Verem_DG` csomagot úgy, hogy `Constraint_Error` helyett `Üres_Verem` illetve `Tele_Verem` kivételeket váltson ki! Az új csomag neve legyen `Verem_DGE_Proba`! Természetesen írd át a kipróbáló programot is!

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

11

3. feladat megoldása (1)

```
verem_dge.ads:
...
Tele_Verem, Üres_Verem : exception;
...
verem_dge.adb:
...
exception
    when Constraint_Error =>
        raise Üres_Verem;
...
```

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

12

3. feladat megoldása (2)

```
verem_dge_proba.adb:
...
begin
  Put_Line ( "A verem teteje: " &
             Integer'Image ( Teto ( V ) ) );
exception
  when Üres_Verem =>
    Put_Line ( "A verem üres!" );
end;
(folyt. köv.)
```

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

13

3. feladat megoldása (3)

```
verem_dg_proba2.adb:
(folyt.)
begin
  Betevés ( V, I );
  Put_Line ( "Betéve: " &
             Integer'Image ( I ) );
exception
  when Tele_Verem =>
    Put_Line ( "A verem tele van!" );
end;
...
```

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

14

Segítség a 4. feladathoz

- Kivételek felhasználhatók vezérlésre is, bizonyos elágazások helyett
- Programozástechnikai szempontból nem célszerű erre a célra használni a kivételeket

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

15

4. feladat

Módosítsd a rekurzív függvény segítségével működő `Faktor2` programot úgy, hogy a rekurzív függvényben elágazás helyett kivételkezelés szerepeljen! Az új program neve legyen `Faktor3`, a rekurzív függvényt pedig `Fakt_E`-nek hívják!

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

16

4. feladat megoldása

```
fakt_e.adb:
function Fakt_E ( Szam : Natural )
  return Natural is
begin
  return Number * Fakt_E ( Number - 1 );
exception
  when Constraint_Error => return 1;
end Fakt_E;
```

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

17

Házi feladatok

- Próbáld meg kiírni az 1, 10, 100, ... végtelen sorozat tagjait úgy, hogy a változó, ami a sor aktuális elemét tartalmazza az 1 és 10000 közötti típusérték-halmazból való. Mikor ad hibaüzenetet? És ha kikapcsolod a határok ellenőrzését? (Ld. pragma Suppress, könyv 17.6)

2004. október 21.

Programozási nyelvek I. - 6. gyakorlat

18