Változók hatálya

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2023 szeptember 19.

Változók hatálya (scope)

Megismerkedtünk már több alapvető nyelvi elemmel, amely blokkokra tagolja a programot:

- `if-else` blokk
- ciklusok

Ezekben a blokkokban definiálhatunk lokálisan változókat

- ezek a változók azonban a program többi részében nem láthatóak
- egy adott blokkban definiált változók hatályát/láthatóságát a kapcsos zárójelek jelölik ki

Változók hatálya és a kapcsos zárójelek

```
int main()
{
int a = 5;

for ( int i = 0; i < 6; ++i)
{ int a = i + 1;
  int c = i + 2;
cout << "lokalis a=" << a << endl;
cout << "lokalis c=" << c << endl;
cout << "\n" << endl;
}
cout << "main-ben a=" << a << endl;
//cout << "main-ben c=" << c << endl;

    return 0;
}
```

- a main-ben definiáltunk egy a változót

Változók hatálya és a kapcsos zárójelek

```
int main()
{
int a = 5;

for ( int i = 0; i < 6; ++i)
{ int a = i + 1;
  int c = i + 2;
cout << "lokalis a=" << a << endl;
cout << "lokalis c=" << c << endl;
cout << "\n" << endl;
}
cout << "main-ben a=" << a << endl;
//cout << "main-ben c=" << c << endl;

    return 0;
}
```

- a main-ben definiáltunk egy a változót
- a for ciklus {} blokkján belül új változók definiálhatóak
- ezeknek ugyanaz a nevük is lehet, mint a main-ben definiált változónak
- a fordító figyelmeztethet az a miatt: `...declaration shadows local variable...`

Változók hatálya és a kapcsos zárójelek

```
1  int main()
2  {
3  int a = 5;
4
5  for ( int i = 0; i < 6; ++i)
6  { int a = i + 1;
7    int c = i + 2;
8  cout << "lokalis a=" << a << endl;
9  cout << "lokalis c=" << c << endl;
10 cout << "\n" << endl;
11 }
12 cout << "main-ben a=" << a << endl;
13 //cout << "main-ben c=" << c << endl;
14
15     return 0;
16 }
```

Fontos észrevenni:

- a `for {...}`-ban definiált `a` változó lokálisan felüldefiniálja a `main`-ben definiált `a`-t
- a `c` változó a `for` ciklus blokkján kívül nem látszik
- ezért a 13. sor fordítási hibát okozna

Változók hatálya és a kapcsos zárójelek

Az program futásának eredménye:

```
lokalis a=1
lokalis c=2

lokalis a=2
lokalis c=3

lokalis a=3
lokalis c=4

lokalis a=4
lokalis c=5

lokalis a=5
lokalis c=6

lokalis a=6
lokalis c=7

main-ben a=5
```

- a `for` ciklus blokkjában definiált `a` értéke nem írta felül a `main`-ben definiált `a` értékét

Ajánlások

Minden változót a lehető legszűkebb { } –blokkban (scope-ban) használjunk

Kerüljük az ún “shadowing”-ot, vagyis a változónevek legyenek egyediek