

WYKŁAD: Programowanie obiektowe



dr Artur Bartoszewski
Katedra Informatyki
UTH Radom

Pola statyczne klas

Składniki statyczne klas

Pola statyczne – stosujemy, gdy wszystkie egzemplarze obiektów danej klasy powinny współdzielić tę samą daną.

Pole statyczne jest w pamięci tworzone jednokrotnie i jest wspólne dla wszystkich egzemplarzy obiektów danej klasy.

Co więcej: istnieje nawet wtedy, gdy jeszcze nie zdefiniowaliśmy ani jednego egzemplarza obiektu tej klasy.

```
class klasa{  
    public :  
    int x ;  
    static int skladnik ;  
};
```

Składniki statyczne klas

- ✓ Deklaracja pola statycznego w ciele klasy nie jest jego definicją.
- ✓ Definicję musimy umieścić w takim miejscu programu, aby miała zakres pliku. Czyli tak, jak definicję zmiennej globalnej.
- ✓ Definicja taka może zawierać inicjalizację.

```
int klasa::skladnik = 6;
```

- ✓ Pole statyczne może być także typu ***private***.
- ✓ Inicjalizacja pola statycznego możliwa jest nawet jeśli jest ono typu ***private***. Po inicjalizacji prywatne pole statyczne nie może być czytane ani zapisywane z poza klasy.

Składniki statyczne klas

Do składnika statycznego można odwołać się na trzy sposoby:

1. Za pomocą nazwy klasy i operatora zakresu „ :: ”

`klasa::składnik`

2. Jeśli istnieją już jakieś egzemplarze obiektów klasy, to możemy posłużyć się operatorem „ . ”

`obiekt.składnik`

3. Jeśli mamy wskaźnik do obiektu stosujemy operator „->”

`*wsk = &obiekt;`

`wsk->składnik;`

Składniki statyczne klas

```
1      #include <iostream>
2      class Klasa
3      {
4          static int wspolne;
5      public:
6          void metoda(){
7              std::cout << wspolne << std::endl;
8              ++wspolne;
9          }
10     };
11
12     int Klasa::wspolne = 0;
13
14     int main()
15     {
16         Klasa a, b, c;
17     }
```

Przykład:

Klasa automatycznie nadająca
kolejne numery id nowo
tworzonym obiektom

```
3  class osoba
4  {
5  public:
6      int Id;
7      string imie;
8      static int ile;
9      osoba() {ile++; Id=ile; getline(cin, imie);}
10     void wypisz() {cout<<Id<<" : "<<imie<<endl;}
11     ~osoba() {cout<<"obiekt "<<Id<<" usunieto"<<endl;}
12 };
13 int osoba::ile=0;
14 const int n=3;
15 int main()
16 {
17     osoba * lista[n];
18     for (int i=0; i<n;i++) lista[i]= new osoba();
19     for (int i=0; i<n;i++) lista[i]->wypisz();
20     for (int i=0; i<n;i++) delete lista[i];
21     return 0;
22 }
```

Literatura:

W prezentacji wykorzystano przykłady i fragmenty:

- Grębosz J. : ***Symfonia C++, Programowanie w języku C++ orientowane obiektowo***, Wydawnictwo Edition 2000.
- Jakubczyk K.: *Turbo Pascal i Borland C++ Przykłady*, Helion.

Warto zajrzeć także do:

- Sokół R. : ***Microsoft Visual Studio 2012 Programowanie w Ci C++***, Helion.
- Kernighan B. W., Ritchie D. M.: ***język ANSI C***, Wydawnictwo Naukowo Techniczne.

Dla bardziej zaawansowanych:

- Grębosz J. : ***Pasja C++***, Wydawnictwo Edition 2000.
- Meyers S.: ***język C++ bardziej efektywnie***, Wydawnictwo Naukowo Techniczne