

2D1387 Programsystemkonstruktion med C++

Lösningsförslag, tentamen 17 april 2001

Obs: Dessa lösningar är just lösningsförslag. Rättningen kan se något annorlunda ut.

Uppgift 1

A B A B B (på fem rader)

`delete ap` ger ingen utskrift (eftersom klassen A saknar virtuell destruktör) och den sista utskriften (B) kommer från destruktör av B b.

Uppgift 2

a) `int get_elem(const int *, const int *, bool (*)(int, int))`

b) Exempel på program:

```
template<class T, class F>
T get_elem(T *begin, T *end, F comp)
{
    T ret = *begin;
    for(T *i = begin + 1; i != end; i++)
        if(!comp(ret, *i))
            ret = *i;
    return ret;
}
```

Uppgift 3

Följande fyra defekter ger kompilatorfel:

- `C *pc = new B`; En s.k. *down-cast* är aldrig tillåten, dvs försök till att tilldela en basklasspekare (`B *`) till en pekare till nedärvd klass (`C *`).
- `A *const pa2 = pa1`; Den nya typen (`A *const`) är mindre restriktiv än typen på det tilldelade objektet (`const A *`), t.ex. skulle man kunna ändra i det konstanta objektet som `pa1` pekar på genom `pa2`.
- `pa2->bar()`; Funktionen `bar` finns inte i klassen A.
- `delete pa2`; Länkningsfel eftersom destruktorn i klassen A saknar definition.

Uppgift 4

a) De olika lagringstyperna:

stacken Automatiska objekt hamnar här och allokeras vid början av det block där de definierats och avallokeras vid slutet av blocket.

heapen Objekt allokerade med `new` och `new[]`. Allokering resp. avallokering sker genom anrop till `new` resp. `delete`.

statiska objekt Allokeras på en fast minnesadress vid första tillfället de används (dvs då exekveringen passerar definitionen) och inte tidigare.

globala objekt Allokeras vid programmets början på en fast minnesadress. Avallokering sker vid programslut. Avallokeringen sker i omvänd ordning jämfört med allokeringen.

b) Exempel på rudimentär trådklass:

```
class Thread
{
    ~Thread()    { }
public:
    void run()   { destroy(); }
    void destroy() { delete this; }
};
```

Privat destruktör förbjuder den automatiska destruktionsen om objektet allokeras på stacken. Thread-objekt allokeras med `new` och avallokeras genom anrop till `destroy`.