

```
1:
2: template <class Tag>
3: class Accessor {
4: public:
5:     static typename Tag::type value;
6: };
7:
8: template <class Tag>
9: typename Tag::type Accessor<Tag>::value;
10:
11: template <class Tag, typename Tag::type p>
12: class Initializer {
13: public:
14:     Initializer()
15:     {
16:         Accessor<Tag>::value = p;
17:     }
18:     static Initializer instance;
19: };
20:
21: template <class Tag, typename Tag::type p>
22: Initializer<Tag, p> Initializer<Tag, p>::instance;
23:
24:
25:
26: class Data_mem1 {
27: public:
28:     typedef int Data::* type;
29: };
30:
31: class Data_mem2 {
32: public:
33:     typedef int Data::* type;
34: };
35:
36: template class Initializer<Data_mem1, &Data::_mem1>;
37: template class Initializer<Data_mem2, &Data::_mem2>;
```