

```
1 #include "Interface.h"
2
3 using namespace System;
4 using namespace System::ComponentModel::Composition;
5 using namespace System::Collections::Generic;
6
7 namespace SimpleCalculator {
8
9     [Export(typeof(ICalculator))]
10     ref class Calculator : ICalculator {
11     private:
12         [ImportMany]
13         IEnumerable<System::Lazy<IOperation^, IOperationData^>>^ operations_;
14
15     public:
16         // Symbolが一致するIOperationを見つけ、実行する
17         virtual int Calculate(int left, System::String^ operation, int right) {
18             for each (Lazy<IOperation^, IOperationData^>^ item in operations_) {
19                 if (item->Metadata->Symbol == operation)
20                     return item->Value->Operate(left, right);
21             }
22             throw gcnew NotSupportedException(operation);
23         }
24
25         // Symbolの列挙を返す
26         virtual IEnumerable<System::String^>^ Symbols() {
27             auto result = gcnew List<String^>();
28             for each (Lazy<IOperation^, IOperationData^>^ item in operations_) {
29                 result->Add(item->Metadata->Symbol);
30             }
31             return result;
32         }
33     };
34 }
35
36
37
```