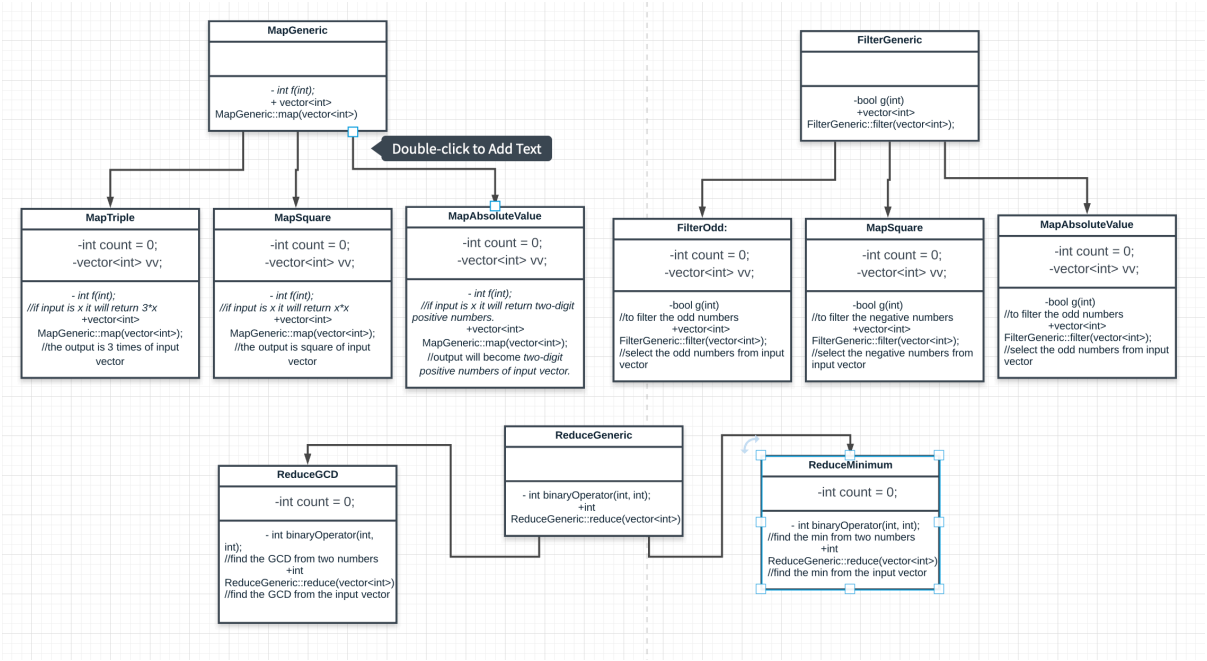
Design

Yian Xie
A1702241



Description:

MapGeneric: has a private method *int f(int)* that specifies the operation we want to map onto a list. This method is overridden later in the derived classes to deliver specific map operations. The function *f* can be declared to be pure virtual. There also have a public method vector<int> MapGeneric::map(vector<int>)

- int f(int);

+ vector<int> MapGeneric::map(vector<int>)

MapTriple: this class inherits MapGeneric class. There have two methods:

*- int f(int);//if input is x it will return 3*x*

+ vector<int> MapGeneric::map(vector<int>);//the output is 3 times of input vector

MapSquare: this class inherits MapGeneric class. There have two methods:

*- int f(int);//if input is x it will return x*x*

+ vector<int> MapGeneric::map(vector<int>);//the output is square of input vector

MapAbsoluteValue: this class inherits MapGeneric class. There have two methods:

- int f(int);//if input is x it will return two-digit positive numbers.

+ vector<int> MapGeneric::map(vector<int>);//output will become *two-digit positive numbers of input vector.*

FilterGeneric: has a private method bool g(int) that specifies the operation we want to map onto a list. This method is overridden later in the derived classes to deliver specific map operations.

-bool g(int)

+ vector<int> FilterGeneric::filter(vector<int>);

FilterOdd: this class inherits FilterGeneric, there have two class:
-bool g(int)//to filter the odd numbers
+ vector<int> FilterGeneric::filter(vector<int>);//select the odd numbers from input vector
FilterNonPositive: this class inherits FilterGeneric, there have two class:
-bool g(int)//to filter the negative numbers
+ vector<int> FilterGeneric::filter(vector<int>);//select the negative numbers from input vector
FilterForTwoDigitPositive: this class inherits FilterGeneric, there have two class:
-bool g(int)//to filter the odd numbers
+ vector<int> FilterGeneric::filter(vector<int>);//select the odd numbers from input vector

ReduceGeneric: has a private method int binaryOperator(int, int) that specifies the operator. This method is overridden later in the derived classes to deliver specific map operations.
- int binaryOperator(int, int);
+int ReduceGeneric::reduce(vector<int>)
ReduceMinimum: this class is inherits ReduceGeneric. There have two classes:
- int binaryOperator(int, int);//find the min from two numbers
+int ReduceGeneric::reduce(vector<int>)//find the min from the input vector

ReduceGCD: this class is inherits ReduceGeneric. There have two classes:
- int binaryOperator(int, int);//find the GCD from two numbers
+int ReduceGeneric::reduce(vector<int>)//find the GCD from the input vector

Testing:

Testing for MapGeneric, MapTriple, MapSquare and MapAbsoluteValue classes:

Input: 1, 3, 5, 9, 10
Expect output:
3 9 15 27 30
1 9 25 81 100
25 81
Output:
3 9 15 27 30
1 9 25 81 100
25 81

Input: 6, -11, -15, 50
Expect output:
18 33 45 150
36 121 225 2500
36
Output:
18 33 45 150
36 121 225 2500
36

Testing for FilterGeneric, FilterOdd, FilterNonPositive and FilterForTwoDigitPositive:

Input: 6, -11, 53, -16, 73, 128, 105, 104, -71
Expect output:
-11 53 73 105 -71
-11 -16 -71
53 73
Output:
-11 53 73 105 -71
-11 -16 -71
53 73

Input: -5, -24, -123, -81, 200, 157, -72, 24
Expect output:
-5 -123 -81 157
-5 -24 -123 -81 -72
24
Output:
-5 -123 -81 157
-5 -24 -123 -81 -72
24

Testing for ReduceGeneric, ReduceMinimum and ReduceGCD:

Input: 4 16 18
Expect output: 4 2
Output: 4 2

Input: 15 25 5 225
Expect output: 5 5
Output: 5 5