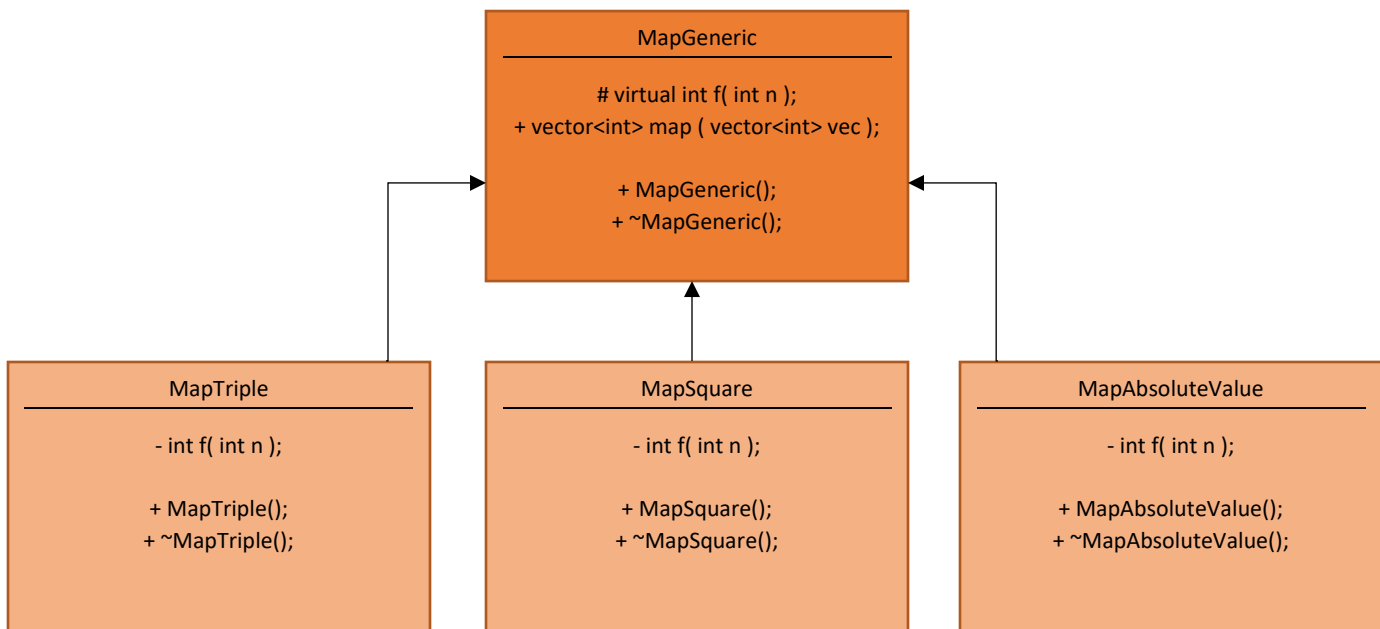


Class Diagram

Map



MapGeneric

virtual int f(int n) : Specifies the operation we want to map onto a list. Method is overridden later in derived classes to deliver specific map operations. It is declared as a pure virtual function.

vector <int> map

(vector<int> vec) : Takes a vector as input and returns the resulting vector after mapping.

MapTriple

int f(int n) : Takes in an integer n and triples it.

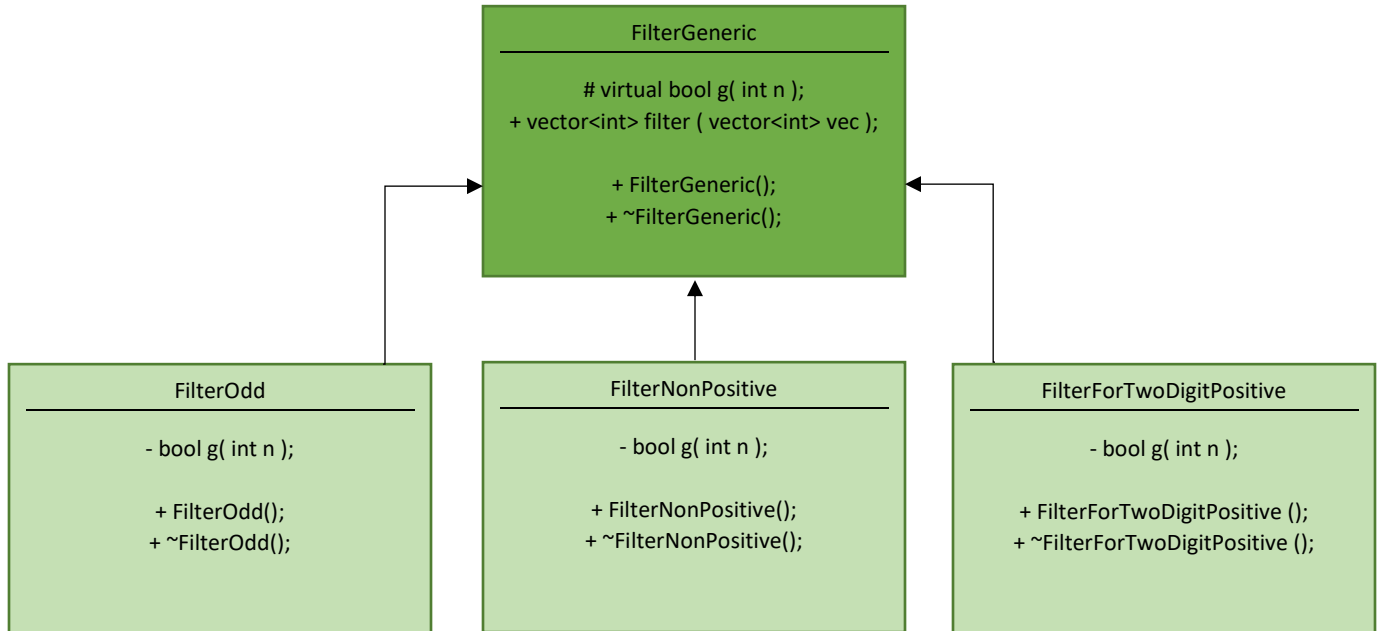
MapSquare

int f(int n): Takes in an integer n and squares it.

MapAbsoluteValue

int f(int n): Takes in an integer n and returns it's absolute value.

Filter



FilterGeneric

virtual bool g(int n) :

Specifies the operation we want to map onto a list. Method is overridden later in the derived classes to deliver specific filter operations. It is declared as a pure virtual function.

**vector<int> filter
(vector<int> vec) :**

Takes a vector as input and returns the resulting vector after filtering.

FilterOdd

bool g(int n) :

Takes in an integer n and returns True if said integer is odd.

FilterNonPositive

bool g(int n) :

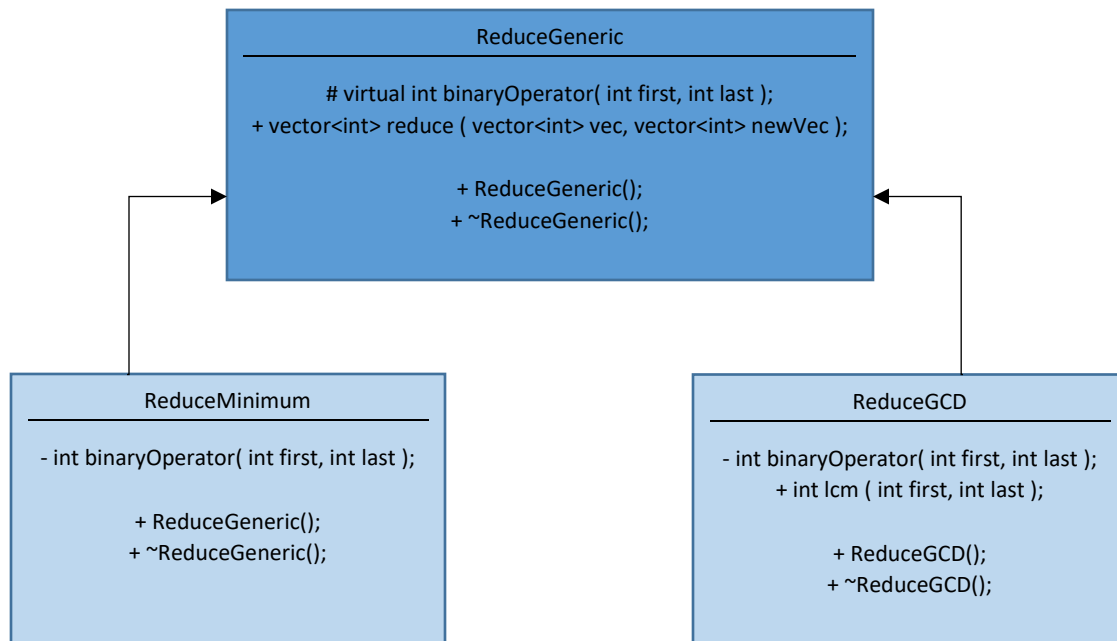
Takes in an integer n and returns True if integer is negative.

FilterForTwoDigitPositive

bool g(int n) :

Takes in an integer n and returns True if the integer has two digits.

Reduce



ReduceGeneric

virtual int binaryOperator

(int first, int last) :

Specifies the operator. Method is overridden in later derived classes to deliver specific map reduce operations. It is declared as a pure virtual function.

vector<int> reduce

**(vector<int> vec,
vector<int> newVec) :**

Takes two vectors, one with the input from the user (vec), and the other an empty vector with the size of 1. And returns the results of reduce.

ReduceMinimum

int binaryOperator

(int first, int last) :

Takes in two integers and compares the two of them, then returns the smaller value.

vector<int> reduce

**(vector<int> vec,
vector<int> newVec) :**

Takes in the first and last element of the entered vector and returns the smaller value of the two and adds it to the vector 'newVec', then deletes it from the initial vector 'vec'. It then recursively continues this process until vec's size equals 1. Then newVec is finally returned.

ReduceGCD

int binaryOperator

(int first, int last) :

Takes two integers named 'first' and 'last'. If last equals 0, then first is returned. Otherwise it returns recursively back through the function 'last' and 'first % last'.

vector<int> reduce

(vector<int> vec,

vector<int> newVec) :

Takes two integers from the original vector inputted and determines the overall greatest common denominator by recursively finding the lowest common denominator using the 'lcm' function between said two integers. Said lcm is added to newVec and the first integer in vec is deleted once checked.

The algorithm for finding the GCD looks like this:

$$GCD(a, b) = \frac{|a \cdot b|}{LCM(a, b)}$$

Test Cases

| Test Case ID | Input | Expected Output | Actual Output | Pass or Fail | Comments |
|--------------|---|-----------------|---------------|--------------|---|
| 01 | " 6, -15, 53, -16, 73, 128, 105, 104, -71, -179, 102, 12, 25, -145, -99, 199, -156, -186, 43, -189" | "45 15" | "45 15" | Pass | To test whether it takes an input of values correctly (including commas and spaces) and returns the expected output. |
| 02 | " 157, -24, -123, -81, 200, 157, 84, 67, -83, -60, -72, 192, -25, -20, -50, -181, -70, -23, -108, -123" | "69 3" | "69 3" | Pass | To test whether it takes this particular set of numbers, including negatives and multiple digits, and successfully returns expected values. |
| 03 | " -21, 91, 46, 74, -44, 149, -192, 41, -9, -32, -133, 137, 178, -4, 119, -9, -111, -144, -184, -33" | "27 9" | "27 9" | Pass | To test whether it takes an input of values correctly (including commas and spaces) and returns the expected output. |