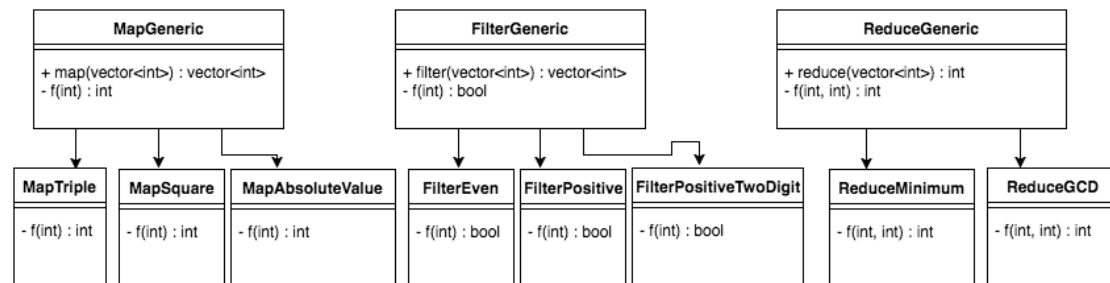


Map, Filter and Reduce

UML Diagram



Description

MapGeneric:

MapGeneric is the parent class for specific map classes. It has two functions:

`map()` – this function takes a vector of integers and recursively applies a function (the function that is applied is `f()`.) to each value in the vector and returns the resulting vector.

`f()` – this function is private. It is used by `map()` to take an integer and do something to it and hand back the resulting integer. This function is purely virtual and depends on the child class's implementation.

MapTriple, MapSquare and MapAbsoluteValue:

These classes are different maps, each contains its own implementation of the `f()` function:

MapTriple's `f()` triples the given integer,

MapSquare's `f()` returns the given integer squared,

MapAbsoluteValue's `f()` returns the absolute value of the given integer.

FilterGeneric:

FilterGeneric is the parent class for specific filter classes. It has two functions:

`filter()` – this function takes a vector of integers and recursively tests each one using the function `f()`. The resulting vector contains only the values that are true for the filter.

f() – this function is private. It is used by filter() to take an integer and check it and return true or false. This function is purely virtual and depends on the child class's implementation.

FilterEven, FilterPositive and FilterPositiveTwoDigits:

These classes are different filters, each contains its own implementation of the f() function:

FilterEven's f() returns true if the given integer is even,
FilterPositive's f() returns true if the given integer is positive,
FilterPositiveTwoDigit's f() returns true if the given integer is positive and it has only two digits

ReduceGeneric:

ReduceGeneric is the parent class for specific reduce classes. It has two functions:

reduce() – this function takes a vector of integers and recursively compares or combines each one using the function binary_operator(). The result is only one integer that is the outcome of all the comparisons or the combination.

binary_operator() – this function is private. It is used by reduce() to take two integers and compare or combine them in a certain way and return one resulting integer. This function is purely virtual and depends on the child class's implementation.

ReduceMinimum and ReduceGCD:

These classes are different reduces, each contains its own implementation of the binary_operator() function:

ReduceMinimum's binary_operator() returns the smallest of two integers,
ReduceGCD's binary_operator() returns the greatest common denominator of two given integers

Main:

The main function creates a vector of 20 values that are input from the user using a loop. The vector then has each value tripled and converted to its absolute value using maps, then the positive, two digit, even numbers are filtered out using filters, two integers are then created to store the minimum value in the vector and the greatest common denominator, these are found using reduces. Finally the minimum and GCD are output with a space inbetween.

Testing

Test cases:

Input:

Expected output: