

ADDS Prac 5 Design

Liam Stevens – a1742143

2 May 2019

UML Diagram

MapGeneric
- mapper: vector<int>
+ f(int) = 0: virtual int + map(vector<int>): vector<int>

MapTriple: public MapGeneric	MapSquare: public MapGeneric	MapAbsoluteValue: public MapGeneric
+ f(int): int	+ f(int): int	+ f(int): int

FilterGeneric
- lastFiltered: vector<int>
+ g(int) = 0: virtual bool + filter(vector<int>): vector<int>

FilterOdd: public FilterGeneric	FilterNonPositive: public FilterGeneric	FilterForTwoDigitPositive: public FilterGeneric
+ g(int): bool	+ g(int): bool	+ g(int): bool

ReduceGeneric
- store: vector<int>
+ binaryOperator(int, int) = 0: virtual int + reduce(vector<int>): vector<int>

ReduceMinimum: public ReduceGeneric	ReduceGCD: public ReduceGeneric
+ binaryOperator(int, int): int	+ binaryOperator(int, int): int

Description

MapGeneric

Calls f() function of respective child class when the child calls map, and performs recursive calculations based on the result

MapTriple

Returns given number after it is multiplied by three

MapSquare

Returns given number after it is squared

MapAbsoluteValue

Returns the absolute value of given number

FilterGeneric

Calls g() function of respective child class when the child calls filter, and recursively moves integers based on the result

FilterOdd

returns true if odd, otherwise returns false

FilterNonPositive

Returns true if number is less than 0, otherwise returns false

FilterForTwoDigitPositive

Returns true if number is between 9 and 100, otherwise returns false

ReduceGeneric

Calls binaryOperator function of respective child class when the child calls reduce, and performs recursive calculations based on the result

ReduceMinimum

Returns the lower of two given numbers

ReduceGCD

Finds and returns the GCD of two given numbers

Main

The main function will take a single string input and then converts it to an integer list stored in the vector. Using the MapTriple and MapAbsolute classes, it will transform the vector. It will then use the FilterOdd and FilterForTwoDigitPositive classes to transform the vector. Finally it will use the ReduceMinimum and ReduceGCD classes and store the results of their return in separate values then output them to the console.

Testing

Input: -101 -100 -99 -11 -10 -9 -1 0 1 9 10 11 99 100 101 33 34 35 -33 -34

Expected Output: 27 3

Input: 9 -10 2 100 345 -102 66 33 2 -6 82 91 -201 12 23 7 3 -5 99 1

Expected Output: 15 3