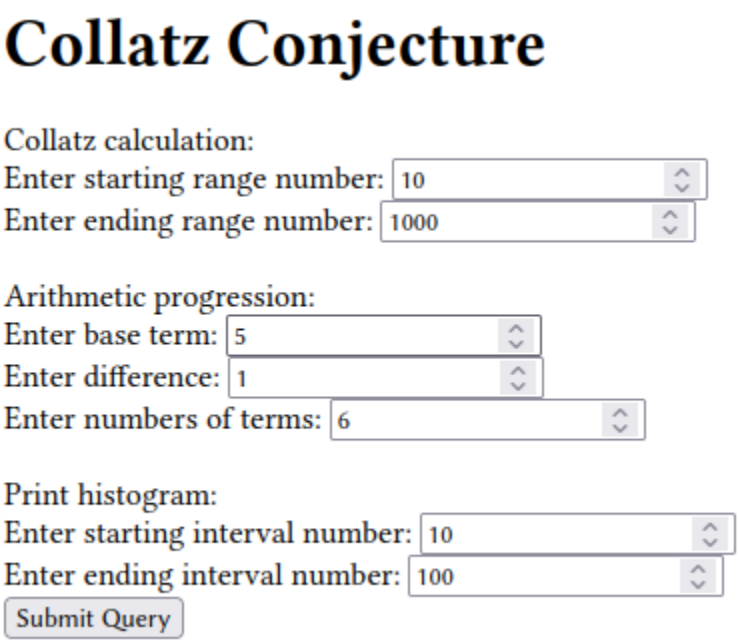
Task: Inheritance, UML

Vladyslav Vorobiov



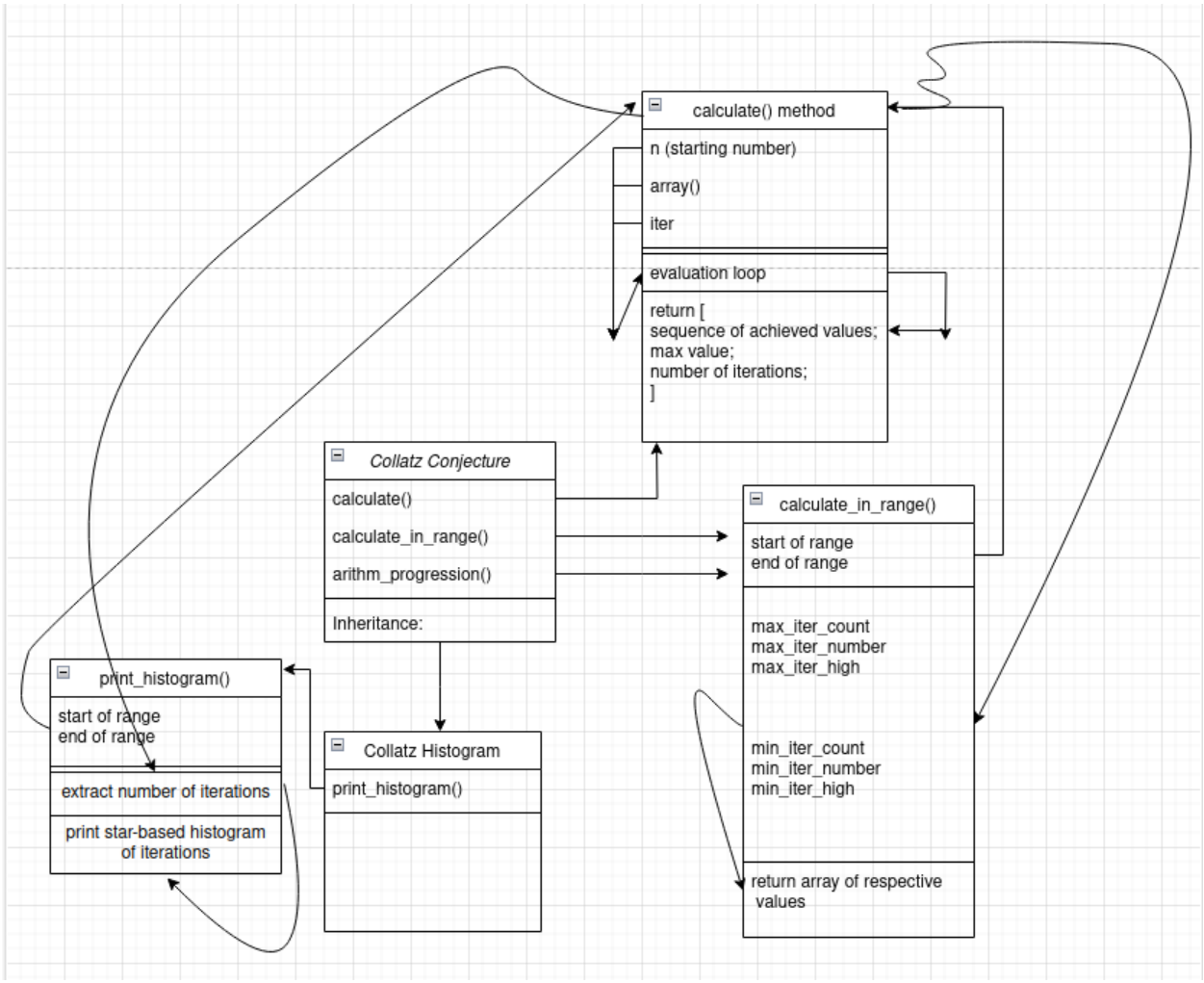
Three screenshots below:

Range: 10 to 1000
Max iterations: 178 (Number: 871, Max value: 190996)
Min iterations: 4 (Number: 16, Max value: 16)

Usage of arithmetic progression:
5 -> 6 -> 7 -> 8 -> 9 -> 10

Histogram of iterations in the range [10;100]:
Number: 10 [******] (6 iters)
Number: 11 [**************] (14 iters)
Number: 12 [*********] (9 iters)
Number: 13 [*********] (9 iters)
Number: 14 [*****************] (17 iters)
Number: 15 [*****************] (17 iters)
Number: 16 [****] (4 iters)
Number: 17 [************] (12 iters)
Number: 18 [********************] (20 iters)
Number: 19 [********************] (20 iters)
Number: 20 [*******] (7 iters)
Number: 21 [*******] (7 iters)
Number: 22 [***************] (15 iters)
Number: 23 [***************] (15 iters)
Number: 24 [**********] (10 iters)
Number: 25 [************************] (23 iters)
Number: 26 [**********] (10 iters)
Number: 27 [**] (111 iters)
Number: 28 [******************] (18 iters)
Number: 29 [******************] (18 iters)
Number: 30 [******************] (18 iters)
Number: 31 [***] (106 iters)
Number: 32 [*****] (5 iters)
Number: 33 [*************************] (26 iters)
Number: 34 [*************] (13 iters)
Number: 35 [*************] (13 iters)
Number: 36 [*********************] (21 iters)
Number: 37 [*********************] (21 iters)
Number: 38 [*********************] (21 iters)
Number: 39 [**********************************] (34 iters)
Number: 40 [********] (8 iters)
Number: 41 [**] (109 iters)
Number: 42 [********] (8 iters)
Number: 43 [*****************************] (29 iters)
Number: 44 [****************] (16 iters)
Number: 45 [****************] (16 iters)
Number: 46 [****************] (16 iters)
Number: 47 [***] (104 iters)
Number: 48 [***********] (11 iters)
Number: 49 [************************] (24 iters)
Number: 50 [************************] (24 iters)

Number: 52 [**********] (11 iters)
Number: 53 [**********] (11 iters)
Number: 54 [**********] (112 iters)
Number: 55 [**********] (112 iters)
Number: 56 [*******************] (19 iters)
Number: 57 [********************************] (32 iters)
Number: 58 [*******************] (19 iters)
Number: 59 [********************************] (32 iters)
Number: 60 [*******************] (19 iters)
Number: 61 [*******************] (19 iters)
Number: 62 [**********] (107 iters)
Number: 63 [**********] (107 iters)
Number: 64 [******] (6 iters)
Number: 65 [**************************] (27 iters)
Number: 66 [**************************] (27 iters)
Number: 67 [**************************] (27 iters)
Number: 68 [**************] (14 iters)
Number: 69 [**************] (14 iters)
Number: 70 [**************] (14 iters)
Number: 71 [**********] (102 iters)
Number: 72 [**********************] (22 iters)
Number: 73 [**********] (115 iters)
Number: 74 [**********************] (22 iters)
Number: 75 [**************] (14 iters)
Number: 76 [**********************] (22 iters)
Number: 77 [**********************] (22 iters)
Number: 78 [**********************************] (35 iters)
Number: 79 [**********************************] (35 iters)
Number: 80 [*********] (9 iters)
Number: 81 [**********************] (22 iters)
Number: 82 [**********] (110 iters)
Number: 83 [**********] (110 iters)
Number: 84 [*********] (9 iters)
Number: 85 [*********] (9 iters)
Number: 86 [*****************************] (30 iters)
Number: 87 [*****************************] (30 iters)
Number: 88 [*****************] (17 iters)
Number: 89 [*****************************] (30 iters)
Number: 90 [*****************] (17 iters)
Number: 91 [**********] (92 iters)
Number: 92 [*****************] (17 iters)
Number: 93 [*****************] (17 iters)
Number: 94 [**********] (105 iters)
Number: 95 [**********] (105 iters)
Number: 96 [************] (12 iters)
Number: 97 [**********] (118 iters)
Number: 98 [************************] (25 iters)
Number: 99 [************************] (25 iters)
Number: 100 [************************] (25 iters)


calculate() method
n (starting number)
array()
iter
evaluation loop
return [
sequence of achieved values;
max value;
number of iterations;
]
Collatz Conjecture
calculate()
calculate_in_range()
arithm_progression()
Inheritance:
calculate_in_range()
start of range
end of range
max_iter_count
max_iter_number
max_iter_high
min_iter_count
min_iter_number
min_iter_high
return array of respective values
print_histogram()
start of range
end of range
extract number of iterations
print star-based histogram of iterations
Collatz Histogram
print_histogram()