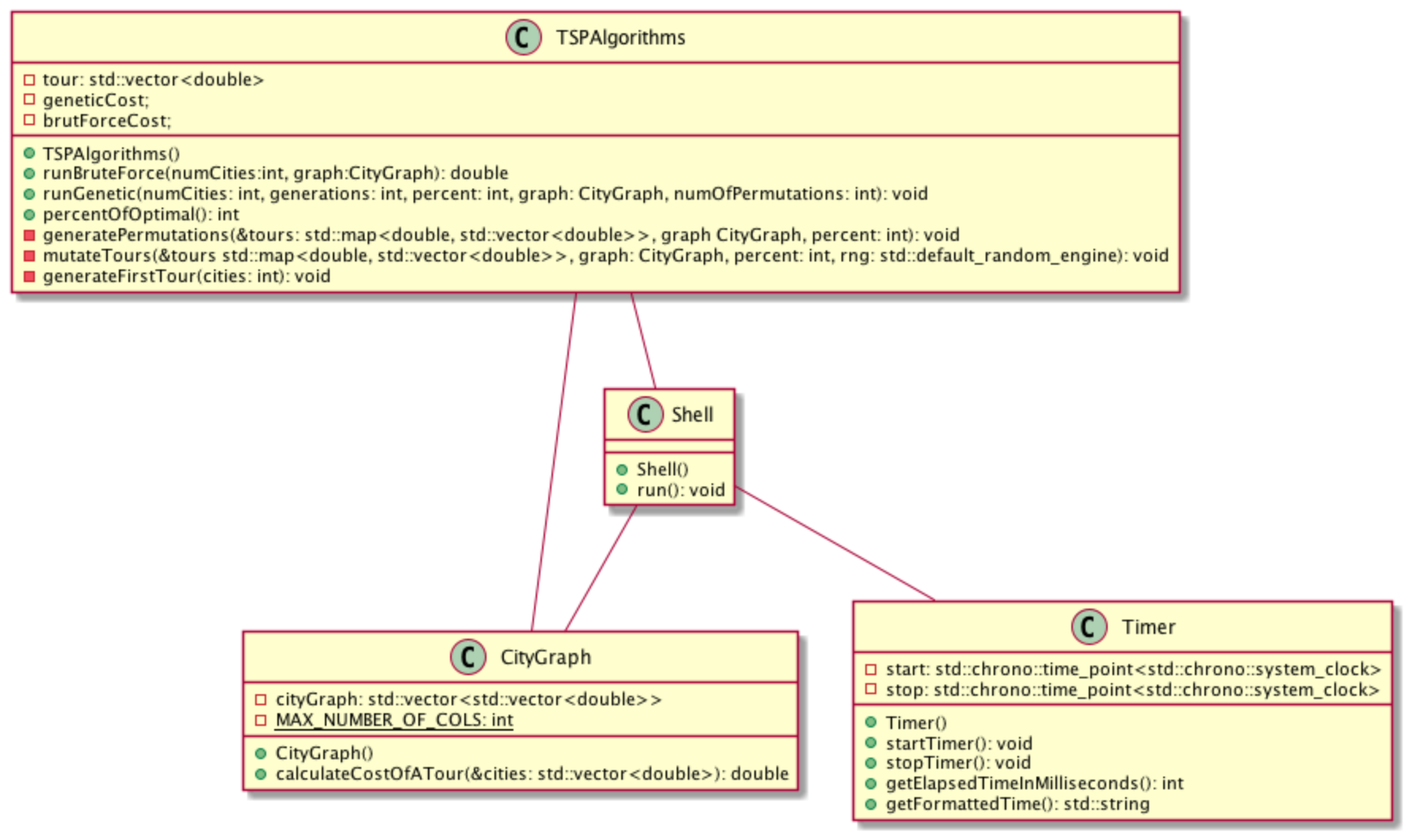
TSPAlgorithms
tour: std::vector<double>
geneticCost;
brutForceCost;
TSPAlgorithms()
runBruteForce(numCities:int, graph:CityGraph): double
runGenetic(numCities: int, generations: int, percent: int, graph: CityGraph, numOfPermutations: int): void
percentOfOptimal(): int
generatePermutations(&tours: std::map<double, std::vector<double>>, graph CityGraph, percent: int): void
mutateTours(&tours std::map<double, std::vector<double>>, graph: CityGraph, percent: int, rng: std::default_random_engine): void
generateFirstTour(cities: int): void
Shell
Shell()
run(): void
CityGraph
cityGraph: std::vector<std::vector<double>>
MAX_NUMBER_OF_COLS: int
CityGraph()
calculateCostOfATour(&cities: std::vector<double>): double
Timer
start: std::chrono::time_point<std::chrono::system_clock>
stop: std::chrono::time_point<std::chrono::system_clock>
Timer()
startTimer(): void
stopTimer(): void
getElapsedTimeInMilliseconds(): int
getFormattedTime(): std::string