Wstępny szkic podstawowych class

Typ zmiennych jest informacyjny nie sugerować się nim

Class Network

\_list\_of\_elements\_in\_network(Object\*\*)

Add\_element(Object\*)

Remove\_element(Object\*)

Class Object – do poliformizmu class

Id(int)

get\_id()

Class Warehouse

\_list\_of\_products(Product\*\*)

\_list\_of\_delivers(Worker\*\*)

Add\_Product(Product\*)

Add\_Deliver(Worker\*)

Class Ramp

time\_of\_delivery(const int)

full(bool) //myślę że nie będzie używana ale na razie zostawię

probability(float\*)

\_list\_of\_receivers(Receiver\*\*)

\_list\_of\_products(Product\*) //tylko 1 ale musimy przechowywać informacje o nim

Add\_Receiver(Receiver\*,float)

Add\_Product(Product\*)

Class Product

Id(int)

time\_of\_geting\_to\_warehouse(int)

Add\_time();

Class Worker

time\_of\_ production(const int)

\_list\_of\_products(Product\*\*)

\_list\_of\_receivers(Receiver\*\*)

Probability(float\*)

Add\_Receiver(Receiver\*,float)

Add\_Product(Product\*)

Tworzenie raportu element po elemencie(wykorzystanie jako suma stringów bez zbędnych warunków)

Class Report

Virtual Add\_Report(Object\*)=0

Class Report\_Worker

Add\_Report(Worker\*)

Class Report\_Ramp

Add\_Report(Ramp\*)

Class Report\_Warehouse

Add\_Report(Warehouse\*)

Class Simulation //myślę że w ten sposób będzie bardziej przejrzyście

\_network(Network\*)

start()

stop()

Funkcje zewnętrzne:

Create\_Raport\_State(const Network&)

Create\_Raport\_Framework(const Network&)

Create\_Raport() – forma interfejsu do wyboru sposobu przedstawienia raportu

//lub zastąpić 3-4 class.

Load\_elements\_from\_file(string ,Network&)

Save\_elements\_to\_file(string,const Network&)

//lub stworzyć klasę do obsługi wyjścia/wejścia