Derek Florimonte
909422688
dflomo@gmail.com

README

This program is designed to test the implementation of virtual function, and fluid use of derived class and subclass functions. It will create multiple objects of MapItem class, utilizing different subclasses (residential, factory, etc...), and test outputs, ensure proper implementation of key functions.

Sample Output:

```
➜ 6 git:(master) ✗ make clean; make rm -f *.o driver
g++ -std=c++11 -c -g -Wall main.cpp
g++ -std=c++11 -c -g -Wall inheritance_test.cpp
g++ -std=c++11 -c -g -Wall map_item.cpp
g++ -std=c++11 -c -g -Wall residential.cpp
g++ -std=c++11 -c -g -Wall factory.cpp
g++ -std=c++11 -c -g -Wall no_op_item.cpp
g++ -std=c++11 -c -g -Wall queue.cpp
g++ -std=c++11 -c -g -Wall node.cpp
g++ -std=c++11 main.o inheritance_test.o map_item.o residential.o factory.o no_op_item.o queue.o
node.o -g -o driver

➜ 6 git:(master) ✗ ./driver
--- Testing Checking if NoOpItem is instantiable --- PASS
--- Testing Checking if Factory is instantiable
--- PASS --- Testing Checking Factory::produceItem() and Factory::productionCount() --- PASS
--- Testing Checking Factory::tick(), Factory::producedCount(), and Factory::getProducedItem() ---
PASS
--- Testing Checking Factory::tick() for empty production queue --- PASS
--- Testing Checking Factory::getProducedItem() returns -1 when empty --- PASS
--- Testing Checking if Residential is instantiable --- PASS
--- Testing Checking Residential::collectTaxes() at time 0 --- PASS
--- Testing Checking Residential::tick() --- PASS
➜ 6 git:(master) ✗
```