

fév 21, 10 16:34

## Gadget.h

Page 1/1

```

#ifndef GADGET_H
#define GADGET_H

#include <string>
5 using namespace std;

class Gadget {

private:
10 static const string defaultColor;
static int lastSerialNumber;
static int numberOfGadgets;

static int makeNewSerialNumber();

15 int my_serialNumber;
string my_color;

20 public:
// constructeurs
Gadget ();
Gadget (const string & color);
Gadget (const Gadget & otherGadget);

25 // méthodes de classe
static int getNumberOfGadgets();

// méthodes d'instance
30 Gadget & operator=(const Gadget & otherGadget);
int getSerialNumber() const;
string getColor() const;
};
#endif

```

fév 21, 10 16:38

## Gadget.cc

Page 1/1

```

#include "Gadget.h"

#include <string>
using namespace std;
5 const string Gadget::defaultColor = "black";

int Gadget::lastSerialNumber = 2010000;
int Gadget::numberOfGadgets = 0;

10 // constructeurs

Gadget::Gadget () {
my_serialNumber = makeNewSerialNumber();
15 my_color = defaultColor;
}

Gadget::Gadget (const string & color) {
20 my_serialNumber = makeNewSerialNumber();
my_color = color;
}

Gadget::Gadget (const Gadget & otherGadget) {
25 my_serialNumber = makeNewSerialNumber();
my_color = otherGadget.my_color;
}

// affectation

30 Gadget & Gadget::operator=(const Gadget & otherGadget) {
my_color = otherGadget.my_color;
return *this;
}

35 // méthodes de classe

int Gadget::makeNewSerialNumber() {
numberOfGadgets ++;
return lastSerialNumber++;
40 }

int Gadget::getNumberOfGadgets() {
return numberOfGadgets;
}

45 // méthodes d'instances

int Gadget::getSerialNumber() const {
50 return my_serialNumber;
}

string Gadget::getColor() const {
return my_color;
}

```

fév 21, 10 16:37

testGadget.cc

Page 1/1

```

#include "Gadget.h"
#include <cassert>
#include <iostream>

5 using namespace std;

int main()
{
    cout << "Tests for Gadget" << endl;
10    cout << "- default constructor" << endl;

    assert (Gadget::getNumberOfGadgets() == 0);
    Gadget g;
15    assert (Gadget::getNumberOfGadgets() == 1);
    assert (g.getColor() == "black");

    cout << "- constructor" << endl;

20    Gadget john("red"), paul("green");
    assert (Gadget::getNumberOfGadgets() == 3);

    assert( john.getColor() == "red");
    assert( paul.getColor() == "green");
25    assert( john.getSerialNumber()
           != paul.getSerialNumber());

    cout << "- copy constructor" << endl;
30    Gadget julian = john;

    assert (julian.getColor() == john.getColor());
    assert (julian.getSerialNumber() != john.getSerialNumber());

35    Gadget jared("whatever");

    cout << "- assignment" << endl;
    jared = paul;

40    assert (jared.getColor() == paul.getColor());
    assert (jared.getSerialNumber() != paul.getSerialNumber());

    cout << "Ok" << endl;
}

```

fév 21, 10 16:41

Makefile

Page 1/1

```

CXXFLAGS=-Wall -pedantic
SOURCES=Gadget.h Gadget.cc testGadget.cc Makefile
A2PSOPT=-q -A fill -C --toc

5 all: testGadget Gadget.pdf

testGadget: testGadget.o Gadget.o
$(LINK.cc) -o $@ $^

10 testGadget.log: testGadget
./testGadget > $@ 2>&1

Gadget.pdf: $(SOURCES) testGadget.log
a2ps $(A2PSOPT) $(SOURCES) testGadget.log -o - | ps2pdf - $@

15 depend:
makedepend testGadget.cc Gadget.cc

clean :
20 -rm *~ *.o *.bak

# DO NOT DELETE

testGadget.o: Gadget.h
25 Gadget.o: Gadget.h

```

fév 21, 10 16:41	testGadget.log	Page 1/1
<pre> Tests for Gadget - default constructor - constructor - copy constructor 5 - assignment Ok                     </pre>		

fév 21, 10 16:41	Table of Content		Page 1/1
	<b>Table of Contents</b>		
	1 <i>Gadget.h</i> ..... sheets	1 to 1 ( 1) pages	1- 1 35 lines
	2 <i>Gadget.cc</i> ..... sheets	1 to 1 ( 1) pages	2- 2 55 lines
	3 <i>testGadget.cc</i> ..... sheets	2 to 2 ( 1) pages	3- 3 45 lines
5	4 <i>Makefile</i> ..... sheets	2 to 2 ( 1) pages	4- 4 26 lines
	5 <i>testGadget.log</i> ..... sheets	3 to 3 ( 1) pages	5- 5 7 lines