

Week 10

Singleton

https://youtu.be/oKxLg_fhoo8

2

Content

In this lab, we will review the following topics:

- How to implement Singleton

3

Assignments

H: YY: 02 (2/3 kinds of loggers)

3.1 Assignment Logger

A logger is a class that allows a developer to write messages. These messages help develop find bugs in a realworld systems.

For example, you build a e-commerce website. It runs very well in the daytime, from 8:00 AM to 22:00 PM. But, your website is often crashed at midnight. If you use a logger to write down messages when implementing the website, these messages will help you solve the problems.

Your task in this assignment is implement a simple logger in C++. Actually, you should support 2/3 types of loggers: console logger (print out to console), text file logger (write down to a text file) and csv file logger (write down to a text file in csv format).

Here are several use cases.

In the first project, I use your console logger and its output.

```
class A{
private:
    MySimpleLogger* logger;
public:
    A(){
        logger = MySimpleConsoleLogger::createInstance();
        logger->write("Begin of A::A()");
        logger->write("Middle of A::A()");
        logger->write("End of A::A()");
    }
    void fooA(){
        logger->write("Begin of A::fooA()");
        logger->write("Middle of A::fooA()");
        logger->write("Another logged message in A::fooA()");
        logger->write("End of A::fooA()");
    }
};

class B{
private:
    MySimpleLogger* logger;
public:
    B(){
        logger = MySimpleConsoleLogger::createInstance();
    }
    void fooB(){
        logger->write("A simple logged message in B::fooB()");
    }
};

MySimpleLogger* MySimpleLogger::logger = NULL;
int main(int argc, const char * argv[]) {
    A a1;
    B b1;
    b1.fooB();
    a1.fooA();
    return 0;
}
```

```
2020 8 10 21 13 52 Begin of A::A()
2020 8 10 21 13 52 Middle of A::A()
2020 8 10 21 13 52 End of A::A()
2020 8 10 21 13 52 A simple logged message in B::fooB()
2020 8 10 21 13 52 Begin of A::fooA()
2020 8 10 21 13 52 Middle of A::fooA()
2020 8 10 21 13 52 Another logged message in A::fooA()
2020 8 10 21 13 52 End of A::fooA()
```

In the second project, I use your csv file logger and its output.

```
class A{
private:
    MySimpleLogger* logger;
public:
    A(){
        logger = MySimpleCSVFileLogger::createInstance();
        logger->write("Begin of A::A()");
        logger->write("Middle of A::A()");
        logger->write("End of A::A()");
    }
    void fooA(){
        logger->write("Begin of A::fooA()");
        logger->write("Middle of A::fooA()");
        logger->write("Another logged message in A::fooA()");
        logger->write("End of A::fooA()");
    }
};

class B{
private:
    MySimpleLogger* logger;
public:
    B(){
        logger = MySimpleCSVFileLogger::createInstance();
    }
    void fooB(){
        logger->write("A simple logged message in B::fooB()");
    }
};

MySimpleLogger* MySimpleLogger::logger = NULL;
ofstream MySimpleCSVFileLogger::fout;
int main(int argc, const char * argv[]) {
    A a1;
    B b1;
    b1.fooB();
    a1.fooA();
    return 0;
}
```

2020	8	10	21	16	19	Begin of A::A()
2020	8	10	21	16	19	Middle of A::A()
2020	8	10	21	16	19	End of A::A()
2020	8	10	21	16	19	A simple logged message in B::fooB()
2020	8	10	21	16	19	Begin of A::fooA()
2020	8	10	21	16	19	Middle of A::fooA()
2020	8	10	21	16	19	Another logged message in A::fooA()
2020	8	10	21	16	19	End of A::fooA()

In the last project, I use your text file logger and its output.

```
class A{
private:
    MySimpleLogger* logger;
public:
    A(){
        logger = MySimpleTextFileLogger::createInstance();
        logger->write("Begin of A::A()");
        logger->write("Middle of A::A()");
        logger->write("End of A::A()");
    }
    void fooA(){
        logger->write("Begin of A::fooA()");
        logger->write("Middle of A::fooA()");
        logger->write("Another logged message in A::fooA()");
        logger->write("End of A::fooA()");
    }
};

class B{
private:
    MySimpleLogger* logger;
public:
    B(){
        logger = MySimpleTextFileLogger::createInstance();
    }
    void fooB(){
        logger->write("A simple logged message in B::fooB()");
    }
};

MySimpleLogger* MySimpleLogger::logger = NULL;
ofstream MySimpleTextFileLogger::fout;
int main(int argc, const char * argv[]) {
    A a1;
    B b1;
    b1.fooB();
    a1.fooA();
    return 0;
}
```

```
2020 8 10 21 17 45 Begin of A::A()
2020 8 10 21 17 45 Middle of A::A()
2020 8 10 21 17 45 End of A::A()
2020 8 10 21 17 45 A simple logged message in B::fooB()
2020 8 10 21 17 45 Begin of A::fooA()
2020 8 10 21 17 45 Middle of A::fooA()
2020 8 10 21 17 45 Another logged message in A::fooA()
2020 8 10 21 17 45 End of A::fooA()
```