Паттерн Singleton ревью.

Данный паттерн должен обеспечивать создание только одного экземпляра класса.

Структура:



В классе Singlеton определяется операция Instance, которая позволяет клиентам получать доступ к единственному экземпляру класса.

Реализация класса в программе:

+class Logger

+{

+ private:

+ static Logger \* instance;

+//запрещаем все конструкторы, для невозможности создания объекта вне метода getInstance()

+ Logger() {} //конструктор

+ Logger( const Logger& ); //конструктор копирования

+ Logger& operator=( Logger& );//оператор присваивания

+ public:

+ //метод, возвращающий указатель на единственный объект типа Logger

+ static Logger \* getInstance() {

+ if(!instance)

+ instance = new Logger();

+ return instance;

+ }

+ bool logError(string mes);

+};

Конструктор класса остается пустым, что позволяет создавать новый объект класса только в функции getInstance(), как и необходимо для реализации паттерна.

Реализация функции logError:

+bool Logger::logError(string mes)

+{

+ ofstream out;

+ out.open("log.dat", ios::app);

+ if (out.is\_open())

+ {

+ time\_t t;

+ time(&t);

+ out << "Time: " << ctime(&t) << "Error: " << mes << endl;

+ return true;

+ }

+ else

+ {

+ return false;

+ }

+}

Функция необходима для логирования ошибки. Если объект уже создан, она вернет false, и новый объект уже не будет создан.

Вывод: программа реализует паттерн в соответствии со всеми условиями описанными в книге «Паттерны проектирования банды четырех».