

Padrão Singleton

Costruindo uma classe de Logs

Wagner Reck¹

¹Ciência da Computação - UNIPAMPA

GEDESPA

Sumário

- 1 Sobre esse Talk
- 2 Singleton
 - Padrões
 - Singleton
- 3 Classe de Log
 - Por que Usar Log?
 - Mensagem de Log
 - Armazenando os Logs
 - Usando a classe Log

será visto:

- O que são padrões de projeto
- Introdução ao padrão singleton
- Por que utilizar Log em aplicações
- Como usar singleton para criar um log
- Nível do talk: básico/intermediário
- Disponível em www.github.com/wiglot/GEDESPA

Outline

- 1 Sobre esse Talk
- 2 Singleton
 - Padrões
 - Singleton
- 3 Classe de Log
 - Por que Usar Log?
 - Mensagem de Log
 - Armazenando os Logs
 - Usando a classe Log

O que é um padrão de projeto.

Design pattern

Definitions

Padrões de projeto visam facilitar a reutilização de soluções de desenho, não considerando o código.

- Usados em projetos OO;
- É difícil tratar um um problema;
 - Por que tratar várias vezes?
- Encontrando padrões aumenta reuso;

O que é um padrão de projeto.

Design pattern

Definitions

Padrões de projeto visam facilitar a reutilização de soluções de desenho, não considerando o código.

- Usados em projetos OO;
- É difícil tratar um um problema;
 - Por que tratar várias vezes?
- Encontrando padrões aumenta reuso;

O que é um padrão de projeto.

Design pattern

Definitions

Padrões de projeto visam facilitar a reutilização de soluções de desenho, não considerando o código.

- Usados em projetos OO;
- É difícil tratar um um problema;
 - Por que tratar várias vezes?
- Encontrando padrões aumenta reuso;

Outline

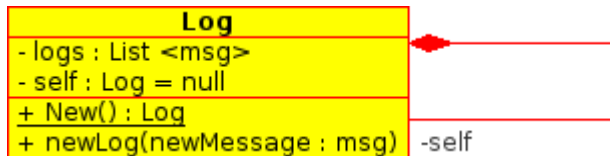
- 1 Sobre esse Talk
- 2 **Singleton**
 - Padrões
 - **Singleton**
- 3 Classe de Log
 - Por que Usar Log?
 - Mensagem de Log
 - Armazenando os Logs
 - Usando a classe Log

O que é o padrão Singleton.

Singleton Pattern

Garante que exista apenas uma instância de uma classe.

- Acesso à objetos de qualquer parte do programa;
- Muito util para classes como MainWindow, Application, Log;
- Fim de variáveis globais (ugh!!);
- Apenas um objeto por classe;



Exemplo básico de classe

```
SingleObject * SingleObject::self = 0;
class SingleObject{
private:
    SingleObject(){ }
    ~SingleObject(){ }
    Static SingleObject * self;
public:
    static SingleObject * New(){
        if (SingleObject::self == 0)
            SingleObject::self = new SingleObject;
        return SingleObject::self;
    };
};
```

Explicando...

```
SingleObject * SingleObject::self = 0;
class SingleObject{
private:
    SingleObject(){ }
    ~SingleObject(){ }
    Static SingleObject * self;
public:
    static SingleObject * New(){
        if (SingleObject::self == 0)
            SingleObject::self = new SingleObject;
        return SingleObject::self;
    };
};
```

Onde é instanciado o Objeto

```
SingleObject * SingleObject::self = 0;
class SingleObject{
private:
    SingleObject(){ }
    ~SingleObject(){ }
    Static SingleObject * self;
public:
    static SingleObject * New(){
        if (SingleObject::self == 0)
            SingleObject::self = new SingleObject;
        return SingleObject::self;
    };
};
```

Usando a classe

```
SingleObject * var = SingleObject::New();  
var->someAction();
```

Outline

- 1 Sobre esse Talk
- 2 Singleton
 - Padrões
 - Singleton
- 3 **Classe de Log**
 - **Por que Usar Log?**
 - Mensagem de Log
 - Armazenando os Logs
 - Usando a classe Log

Posso usar o printf!

- Muito usado no desenvolvimento.
- “Resolve” o problema.
- Não preciso perder tempo escrevendo novas classes.

Posso usar o printf!

- Muito usado no desenvolvimento.
- “Resolve” o problema.
- Não preciso perder tempo escrevendo novas classes.

Posso usar o printf!

- Muito usado no desenvolvimento.
- “Resolve” o problema.
- Não preciso perder tempo escrevendo novas classes.

printf não deve ser usado!

...a não ser que tenha que comunicar algo...

- Como gerenciar diferentes classes de mensagens?
- Como garantir que mensagens não cheguem ao cliente?
- Se acontecer um erro no cliente, consigo quais dados?

printf não deve ser usado!

...a não ser que tenha que comunicar algo...

- Como gerenciar diferentes classes de mensagens?
- Como garantir que mensagens não cheguem ao cliente?
- Se acontecer um erro no cliente, consigo quais dados?

printf não deve ser usado!

...a não ser que tenha que comunicar algo...

- Como gerenciar diferentes classes de mensagens?
- Como garantir que mensagens não cheguem ao cliente?
- Se acontecer um erro no cliente, consigo quais dados?

Outline

- 1 Sobre esse Talk
- 2 Singleton
 - Padrões
 - Singleton
- 3 Classe de Log
 - Por que Usar Log?
 - Mensagem de Log
 - Armazenando os Logs
 - Usando a classe Log

Armazenando mensagens

- Armazenar Mensagem e nível;
 - Mensagem é um texto
 - Nível pertence a um conjunto (log, warning, error, fatal...)

classe Message

```
class Mwssage{  
    enum Nivel{log, warning, error};  
private:  
    string msg;  
    Nivel nivel;  
};
```

Outline

- 1 Sobre esse Talk
- 2 Singleton
 - Padrões
 - Singleton
- 3 **Classe de Log**
 - Por que Usar Log?
 - Mensagem de Log
 - **Armazenando os Logs**
 - Usando a classe Log

Classe Log

- Usa uma classe singleton...
- e uma lista de mensagens (classe message)

Classe Log (cont.)

classe Log

```
class Log {  
private:  
    static Log * self;  
    List <Message> list;  
public:  
    static Log * New(){  
        if (Log::self == null){ Log::self = new Log(); }  
        return Log::self;  
    }  
    static void newLog(Message m){  
        Log::New()->list.append(m);  
    }  
};
```

Reparem na função...

classe Log

```
class Log {  
private:  
    static Log * self;  
    List <Message> list;  
public:  
    static Log * New(){  
        if (Log::self == null){ Log::self = new Log(); }  
        return Log::self;  
    }  
    static void newLog(Message m){  
        Log::New()->list.append(m);  
    }  
};
```

Outline

- 1 Sobre esse Talk
- 2 Singleton
 - Padrões
 - Singleton
- 3 **Classe de Log**
 - Por que Usar Log?
 - Mensagem de Log
 - Armazenando os Logs
 - **Usando a classe Log**

Exemplo simples

```
funcBla(){  
    //.. do something...  
    Log::newLog(msg("Passou aqui..", Log::log));  
    //...continue...  
}
```

Para finalizar

- Breve introdução ao padrão singleton.
- Exemplos de uso em C++
- Melhorias do que foi apresentado.
 - Salvar em arquivo.
 - interface gráfica.
 - www.github.com/wiglot/LogView



Desing Patterns: Elements of Reusable Object-Oriented Software

Eric Gamma

Addison-Wesley Professional.



Utilizando UML e Padrões de Projeto

Craig Larman

Bookman



Wikipedia - Padrões de projeto de software

http://pt.wikipedia.org/wiki/Padrões_de_projeto_de_software



Gambi Design Patterns

http://desciclo.pedia.ws/wiki/Gambi_Design_Patterns <não ler!>

Perguntas

Perguntas??
Wagner Reck
wagner.reck@gmail.com