

Timer & Error 设计报告

数字媒体 3180101044 沈吕可晟

一、模块概述

Timer 负责在系统运行选择等 sql 语句时进行计时操作, 并能打印经过的时间 (s), Error 类贯彻整个数据库系统, 为各种不符合 sql 语法语句的输入以及重复建表等错误抛出异常, 返回错误信息与结果。

二、设计思路

Timer 类采用定义一个 Timer 类对象的方法, 在需要计时的函数的开始与结束调用时调用相关函数, 记录时间戳, 最后通过 PrintTime 函数打印整体过程所经过的时间, 以 s 为单位。

Error 定义了一个通用的异常类 Error 类, 不同的错误类型报错可以通过在抛出其时后面跟上不同的描述字符串来进行区分, 使用简洁明了。

三、数据结构以及类定义

1, Timer 类数据结构定义

```
class Timer {  
    clock_t starttime;  
    clock_t endtime;  
};
```

2, Error 类数据结构定义

```
class Error
{
protected:
    std::string ErrorInfo;
    std::string ErrorPos;
};
```

四、函数接口

Timer 类函数接口:

```
Timer& GetGlobalTimer();
class Timer {
public:
    void Start() {
        starttime = clock();
    }
    void End() {
        endtime = clock();
    }
    void PrintTime() {
        cout << "select time:" << (double)(endtime - starttime) /
CLOCKS_PER_SEC << "s" << endl;
    }
};
```

1. 开始计时函数

```
void Start();
```

2. 结束计时函数

```
void End();
```

3. 打印该计时器经过的时间 (s)

```
void PrintTime();
```

4. 获取全局计时器

```
Timer& GetGlobalTimer();
```

Error 类函数接口:

```
extern std::fstream log_file;
class Error
{
public:
    void PrintError() const;
    Error(const std::string s);
};
// 错误处理函数
void DispatchError(const Error &error);
```

1、构造函数

```
Error::Error(const std::string s)
{
    //ErrorInfo = "Command format error!";
    ErrorInfo = s;
}
```

2、打印错误函数

```
void Error::PrintError() const
{
    // 输出异常
    std::cout << ErrorInfo;
}
```

3、错误处理函数

```
void DispatchError(const Error &error)
{
    error.PrintError();
}
```

五、关键函数和代码

Timer 和 Error 类较为简单，关键函数和代码已经在上一节中提及，此处不再赘述。