Паттерн Одиночка

Паттерн Одиночка гарантирует, что класс имеет только один экземпляр, и предоставляет глобальную точку доступа к этому экземпляру.

Назначение

Гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа.

Применимость

Используйте паттерн одиночка, когда:

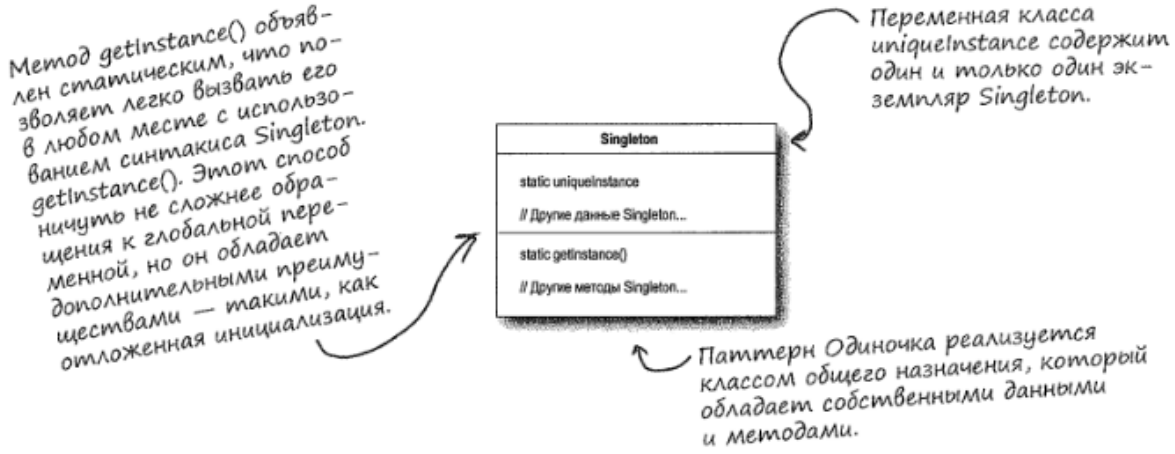
- Должен быть ровно один экземпляр некоторого класса, легко доступный всем клиентам;
- Единственный экземпляр должен расширяться путем порождения подклассов, и клиентам нужно иметь возможность работать с расширенным экземпляром без модификации своего кода.

Результаты

У паттерна одиночка есть определенные достоинства:

- Контролируемый доступ к единственному экземпляру.
- Уменьшение числа имен.
- Допускает уточнение операций и представления.
- Допускает переменное число экземпляров.
- Большая гибкость, чем у операций класса.

Обратимся к диаграмме классов:




```
template<typename T>class singleton
{
public:
    static T& instance()
```

```
    {
        if(!myInstance)
            myInstance = new T;
        return *myInstance;
    }
private:
    static T* myInstance;
    singleton (const singleton&){}
};
template<typename T>
T* singleton::myInstance = NULL;
```