# Применение шаблона проектирования Singleton для класса Controller

### Назначение

Гарантирует существование только одного объекта данного класса и обеспечивает глобальный доступ к этому объекту.

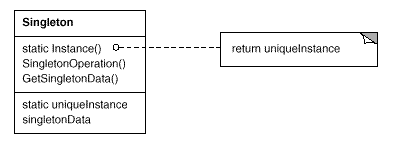
Мотивация

Иногда важно, чтобы существовал ровно один экземпляр какого-либо класса. Система может иметь много принтеров, но должен существовать только один спулер принтера. Должна существовать только одна файловая система и один администратор оконного режима. Цифровой фильтр всегда имеет один A/D преобразователь.

Как можно гарантировать, что класс имеет только один объект и что этот образец является легко доступным? Глобальная переменная делает объект доступным, но не предохраняет Вас от инстанцирования множества объектов.

Лучшее решение состоит в том, чтобы определить класс, непосредственно ответственный за слежение за собственным объектом. Класс может гарантировать, что никакой другой экземпляр не может быть создан (прерывая запросы на создание новых объектов). Этот шаблон - Singleton.

Структура



Составные части

* **Singleton**
  + Определяет операцию Instance, которая позволяет клиентуре обращаться к ее уникальному экземпляру. Instance - операция класса (то есть метод класса в Smalltalk и статический метод в C++).
  + Может быть ответствен за создание его собственного уникального экземпляра.

### Реализация

Класс Controller, координирует работу игры и должен существовать в единственном экземпляре. Для реализации такого класса одиночки идеально подходит паттерн Одиночка (Singleton).

В классе Controller добавляем статическую переменную, содержащую ссылку на единственный экземпляр класса:

static Controller\* \_instance; //!< Единственный объект в программе

Так же добавляем статическую функцию для получения указателя на объект:

//! Если объект уже создан, возвращает ссылку, иначе создает объект

static Controller\* Instance(CMainDlg\* dlg = 0);

Добавляем реализацию и начальную инициализацию переменной:

Controller\* Controller::\_instance = 0;

Controller\* Controller::Instance(CMainDlg\* dlg)

{

if (\_instance == 0)

\_instance = new Controller(dlg);

return \_instance;

}

Далее приведена строчка, где в функции WinMain первый раз запрашивается объект, что приводит к его созданию

Game::Controller\* g\_control = Game::Controller::Instance(dlgMain);