**Аналогия**

У страны может быть только один президент. Он должен действовать, когда того требуют обстоятельства и долг. В данном случае президент — одиночка.

**Вкратце**

Шаблон позволяет удостовериться, что создаваемый объект — единственный в своём классе.

**Википедия**

Шаблон «Одиночка» позволяет ограничивать создание класса единственным объектом. Это удобно, когда для координации действий в рамках системы требуется, чтобы объект был единственным в своём классе.

На самом деле шаблон «Одиночка» считается антипаттерном, не следует им слишком увлекаться. Он необязательно плох и иногда бывает полезен. Но применяйте его с осторожностью, потому что «Одиночка» вносит в приложение глобальное состояние, так что изменение в одном месте может повлиять на все остальные случаи использования, а отлаживать такое — не самое простое занятие. Другие недостатки шаблона: он делает ваш код сильно связанным (tightly coupled), а создание прототипа (mocking) «Одиночки» может быть затруднено.

**Паттерн Singleton важен в случаях, когда для некоторых классов нужно чтобы существовало определенное количество экземпляров, например один экземпляр.**

Использование глобальных переменных не обеспечивает ограничение на создание дополнительных экземпляров классов. Глобальная переменная дает доступ к объекту, но не запрещает создавать несколько экземпляров класса. Поэтому, целесообразно эти ограничения реализовать в самом классе. Именно такой подход предлагает паттерн Singleton.

В паттерне Singleton на класс накладываются следующие обязанности:

* контроль количества экземпляров согласно условию задачи;
* возвращение нужного количества экземпляров по требованию;
* ведение учета полученных экземпляров.

Случаи применения паттерна Singleton:

* когда нужно создать ровно один экземпляр класса;
* когда нужно создать не более заданного количества экземпляров класса;
* когда нужно расширить единственный экземпляр класса производными подклассами. В этом случае клиент имеет возможность использовать расширенные средства (операции, методы) подклассов без модификации своего кода.

<https://radioprog.ru/post/1470>

<https://www.bestprog.net/ru/2020/11/14/patterns-singleton-pattern-overview-features-of-use-implementation-in-c-ru/#contents>

<http://cpp-reference.ru/patterns/creational-patterns/singleton/>