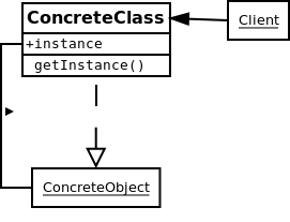
Паршаков Геннадий Пи-15-1

Singleton (Одиночка)

Одиночка — это порождающий паттерн проектирования, который гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа.

**UML Диаграмма:**



1. Пользователь обращается к классу, основанному на паттерне «Одиночка».
2. Класс создаёт новый объект (если он не был создан ранее) и производит определённое преобразование объекта.
3. После чего пользователь может получить значение «одиночки».

**Области применения:**

Все реализации одиночки сводятся к тому, чтобы скрыть конструктор по умолчанию и создать публичный статический метод, который и будет контролировать жизненный цикл объекта-одиночки.

Если у вас есть доступ к классу-одиночке, значит, будет доступ и к этому статическому методу. Из какой точки кода вы бы его не вызвали, он всегда будет отдавать один и тот же объект.

Правительство государства — хороший пример одиночки. В государстве может быть только одно официальное правительство. Вне зависимости от того, кто конкретно заседает в правительстве, оно имеет глобальную точку доступа «Правительство страны N».

**Особенности:**

* Гарантирует наличие единственного экземпляра класса.
* Предоставляет к нему глобальную точку доступа.
* Нарушает принцип единственной ответственности класса.
* Могут возникать проблемы в мультипоточных приложениях.

**Пример реализации:**

В классе объявляется статическая переменная – одиночный объект.

Есть метод, выполняющий преобразование.

И метод, который используется для получения одиночки.

class Singleton

{

    private static Singleton instance;

    private Singleton()

    {}

    public static Singleton getInstance()

    {

        if (instance == null)

            instance = new Singleton();

        return instance;

    }

}