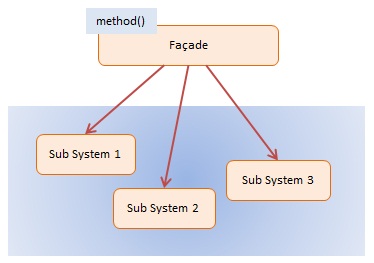
Facade Pattern

Facade Pattern – предоставляет единый интерфейс для нескольких методов.(паттерн структуры)

Цель данного паттерна – скрыть сложность кода за счет использования анонимных функций в качестве внешнего слоя-обертки

Внутренние методы будут доступны только через фасад, что увеличивает уровень безопасности кода



*Facade* – имеет доступ ко всем доступным методам, делегирует «запросы» нужному методу

*Sub Systems* – реализуют функционал, не знают о фасаде и не имеют никакой ссылки на него

***Преимущества:***

*-* увеличивает безопасность приложения

- хорошо работает в совокупности с другими паттернами

- легок в реализации

- позволяет «безболезненно изменять реализацию внутренних методов»

- предоставляет удобный публичный интерфейс

|  |
| --- |
| A simple facade that masks the various browser-specific methods |
| function addEvent( element, event, callback ) {  if( window.addEventListener ) {  element.addEventListener( event, callback, false );  } else if( document.attachEvent ) {  element.attachEvent( 'on' + event, callback );  } else {  element[ 'on' + event ] = callback;  }  } |

|  |
| --- |
| More advanced version of the facade pattern that adds security to internal methods. |
| var MyModule = ( function( window, undefined ) {  // revealing module pattern ftw  function MyModule() {  function someMethod() {  alert( 'some method' );  }    function someOtherMethod() {  alert( 'some other method' );  }    // expose publicly available methods  return {    // in our normal revealing module pattern, we'd do the following:  someMethod : someMethod,    // in the facade pattern, we mask the internals so no one has direct access by doing this:  someMethod : function() {  someMethod();  }  };  }  } )( window ); |

Links

<https://carldanley.com/js-facade-pattern/>

<http://www.dofactory.com/javascript/facade-design-pattern>

http://addyosmani.com/resources/essentialjsdesignpatterns/book/#facadepatternjavascript