ДЕПАРТАМЕНТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ТОМСКОЙ ОБЛАСТИ

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

Специальность 09.02.07 «Информационные системы и программирование»

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7

|  |  |  |
| --- | --- | --- |
| Студент  «\_\_» \_\_\_\_\_\_\_\_ 2020 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | З.Д. Горелова |
| Преподаватель  «\_\_» \_\_\_\_\_\_\_\_ 2020 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | М.Н. Фунтиков |

Томск 2019

**Задание**

1. Создайте объект calculator (калькулятор) с тремя методами:
   * read() (читать) запрашивает два значения и сохраняет их как свойства объекта.
   * sum() (суммировать) возвращает сумму сохранённых значений.
   * mul() (умножить) перемножает сохраненные значения и возвращает результат.

**let *calculator*** = {

*// ... ваш код ...*

};

***calculator***.read();

*alert*( ***calculator***.sum() );

*alert*( ***calculator***.mul() );

1. Это ladder (лестница) – объект, который позволяет подниматься вверх и спускаться:

**let *ladder*** = {

**step**: 0,

up() {

**this**.**step**++;

},

down() {

**this**.**step**--;

},

showStep: **function**() { *// показывает текущую ступеньку*

*alert*( **this**.**step** );

}

};

Теперь, если нам нужно сделать несколько последовательных вызовов, мы можем выполнить это так:

ladder.up();

ladder.up();

ladder.down();

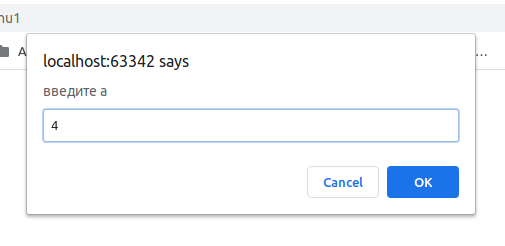
ladder.showStep(); *// 1*

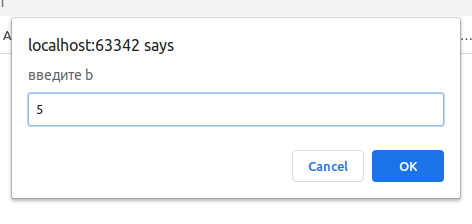
Измените код методов up, down и showStep таким образом, чтобы их вызов можно было сделать по цепочке, например так:

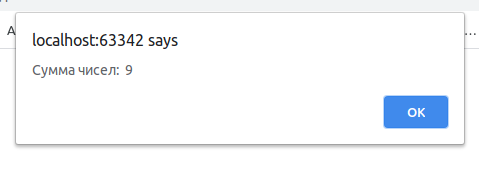
ladder.up().up().down().showStep(); *// 1*

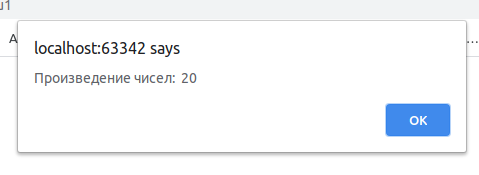
*Такой подход широко используется в библиотеках JavaScript.*

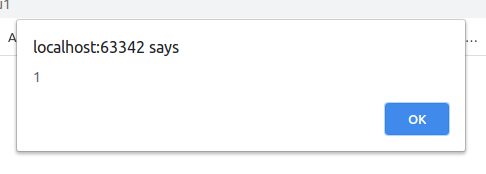
**Результат работы**

1







2

**Код программы**

//создаем объект с тремя методами  
let ***calculator*** = {  
 read() {  
 this.a = +prompt('введите a', 0)  
 this.b = +prompt('введите b', 0)  
 },  
 sum() {  
 return this.a + this.b;  
 },  
 mul(){  
 return this.a \* this.b;  
 }  
};  
  
***calculator***.read();  
alert( `Cумма чисел: ${***calculator***.sum()}`);  
alert( `Произведение чисел: ${***calculator***.mul()}` );

let ***ladder*** = {  
 step: 0,  
 up() {  
 this.step++;  
 return this;  
 },  
 down() {  
 this.step--;  
 return this;  
 },  
 showStep: function() { // показывает текущую ступеньку  
 alert( this.step );  
 return this;  
 }  
};  
  
***ladder***  
.up()  
 .up()  
 .down()  
 .showStep()  
 .up()  
 .down();//1  
// 1