

У нас есть два хомяка: шустрый (`speedy`) и ленивый (`lazy`); оба наследуют от общего объекта `hamster`.

Когда мы кормим одного хомяка, второй тоже наедается. Почему? Как это исправить?

```
let hamster = {
  stomach: []

  eat food() {
    this.stomach.push(food);
  }
}

let speedy = {
  __proto__: hamster
}

let lazy = {
  __proto__: hamster
}

// Этот хомяк нашёл еду
speedy.eat("apple");
alert(speedy.stomach); // apple

// У этого хомяка тоже есть еда. Почему? Исправьте
alert(lazy.stomach); // apple

// -----
```

В коде ниже класс `Rabbit` наследует `Animal`.

К сожалению, объект класса `Rabbit` не создаётся. Что не так? Исправьте ошибку.

```
class Animal {
  constructor(name) {
    this.name = name;
  }
}

class Rabbit extends Animal {
  constructor(name) {
    this.name = name;
    this.created = Date.now();
  }
}

let rabbit = new Rabbit("Белый кролик"); // Error: this is not defined
alert(rabbit.name);

// -----
```

Дополнительно.

Реализовать класс для управления модальными окнами.

1. Внутренности модального окна должны быть динамическими.
2. Возможность отображать “вложенность” модалок. На самом деле отражается только одна модалка в DOM.
3. Класс хранит весь список модальных окон который нужно отобразить.