

Классы. Объектно-ориентированное программирование

1. Инкапсуляция
2. Наследование
3. Полиморфизм
4. Класс: базовый синтаксис
5. Наследование классов
6. Статические свойства и методы
7. Приватные и защищённые методы и свойства
8. Проверка класса: "instanceof"
9. Примеси

Задачи

Сотрудник

базовый

Реализуйте класс `Employee` для ведения бухгалтерии по сотруднику компании. Необходимо хранить следующие данные:

- **fullName.** Имя и Фамилия. Два слова из латинских букв длиной от 3 до 7 символов.
- **rate.** Дневная ставка. Число от 0 до 100000.
- **days.** Количество отработанных дней. Число от 0 до 31.

Конструктор класса принимает один объект с этими данными. В момент создания объекта данные проверяются на валидность. В случае, если одно или несколько полей заполнены не валидно, создание отменяется с ошибкой.

Также, в классе есть вычисляемое свойство `salary` (только для чтения), который возвращает зарплату сотрудника. Зарплатой является произведение ставки `rate` на количество дней `days`.

Ссылки на экземпляры класса хранятся в статическом свойстве `employees`.

```
const jack = new Employee({ fullName: 'Jack Shepard',  
  rate: 500 });  
jack.days = 22;  
console.log(jack.salary); // 11000  
  
const tom = new Employee({ fullName: 'Tom Atkins' });  
tom.days = 25;  
console.log(tom.salary); // 2500  
  
console.log(Employee.getGeneralSum()); // 13500
```

Кроме того, у конструктора есть метод `getGeneralSum()`, который считает сумму зарплат всех сотрудников в текущем месяце.

Примечание 1. Для реализации валидации на стороне сервера можно использовать регулярные выражения и свойства-аксессоры.

Секретный сотрудник

базовый

Реализуйте класс `SecretOfficer`, для ведения бухгалтерии по секретному сотруднику. Он наследует от `Employee`. Свойство `rate` теперь становится недоступным вне класса. Ссылка на экземпляр класса нигде не сохраняется, а его зарплата не учитывается в значении `getGeneralSum()`.

```
const james = new SecretOfficer({ fullName: 'James B',  
  rate: '10000' });  
console.log(james.fullName) // James B  
console.log(james.rate) // Error  
  
james.days = 22;  
console.log(james.salary); // 220000
```