Тесты: Сессия #2. Наследование

Часть 1. Выберете правильный ответ

1. Наследование это -

концепция ООП, позволяющая расширить функциональность одного класса за счет методов и свойств другого класса

Копирование свойств и методов

Создание новой функции на основе существующей

В языке JavaScript нет наследования

2. Класс, унаследованный от другого класса, называется:

предком

базовым классом

суперклассом

подклассом

Все вышеперечисленное

3. Какой метод позволяет обнаружить принадлежит ли свойство самому объекту или его наследнику?

haveOwnProperty

hasInProperty

hasOwnProperty

haveInProperty

4. Класс, от которого новый класс наследуется, называется:

предком

базовым классом

суперклассом

Все вышеперечисленное

5. Что делает Object.create()?

Он возвращает уже готовый пустой объект с прототипом, который задан в аргументе

Всегда создает объект с полями, которые были переданы как аргументы

Такой метод уже не используется

Метод для создания объекта в браузерах IE7 и IE8

6. Найдите ошибку в прототипном наследовании:

1. function Phone (number) {

2. this.number = number;

3. }

4.

5. Phone.prototype.calling = function() {

6. console.log(`Набор номера: ${this.number}`);

7. }

8.

9. function Samsung (number) {

10. this.number = number;

11. }

12.

13. Samsung.prototype = Phone.prototype;

14. Samsung.prototype.constructor = Samsung;

15.

16. Samsung.prototype.calling = function() {

17. console.log(`Вызов ${this.number}`);

18. }

19. let samsung = new Samsung(101);

Пропущено Object.create( ... ) в 13 строчке

Переместить 13 строчку перед 9 строчкой

Функция конструктор не нужна в данном случае

Samsung пишется с маленькой буквы

7. Если оставить предыдущий код без изменений что будет в таком случае :

1. function Phone (number) {

2. this.number = number;

3. }

4.

5. Phone.prototype.calling = function() {

6. console.log(`Набор номера: ${this.number}`);

7. }

8.

9. function Samsung (number) {

10. this.number = number;

11. }

12.

13. Samsung.prototype = Phone.prototype;

14. Samsung.prototype.constructor = Samsung;

15.

16. Samsung.prototype.calling = function() {

17. console.log(`Вызов ${this.number}`);

18. }

19. let samsung = new Samsung(101);

Методы просто копируются, наследование фактически не будет реализовано корректно

Произойдет ошибка

Такая реализация наследования вполне допустима

Это исключение из правил, мы получим 2 метода calling одновременно

8. Что произойдет если не выполнить строку Samsung.prototype.constructor = Samsung; в примере выше?

Свойство constructor дочернего объекта будет перезаписано родительским свойством constructor

Ничего не произойдет

Наследования не произойдет

Интерпретатор вернет ошибку

9. Каков будет результат выполнения этого кода?

function Car (model) {

this.model = model;

}

Car.prototype.getInfo = function(){

return {

state: "new",

fuel: "full"

}

}

function Truck (model) {

this.model = model;

}

Truck.prototype = Object.create(Car.prototype);

Truck.prototype.constructor = Truck;

Truck.prototype.getInfo = function() {

return {

state: "used",

fuel: "middle"

}

}

let truck = new Truck("Volvo");

truck.getInfo();

Объект {state: "new", fuel: "full"}

Объект {state: "used", fuel: "middle"}

10. Сработает ли этот код?

1. function Car (model) {

2. this.model = model;

3. }

4.

5. Car.prototype.getInfo = function(){

6. return {

7. state: "new",

8. fuel: "full"

9. }

10. }

11.

12. function Truck (model) {

13. Car(this, model);

14. }

15.

16. Truck.prototype = Object.create(Car.prototype);

17. //Truck.prototype.constructor = Truck;

18.

19. let truck = new Truck("Volvo");

20. truck.getInfo();

Да, все в порядке

Нет, так как закомментирована строка 17

Нет, ошибка в строке 13

11. Какую строку кода необходимо вставить в 17 строку, чтобы этот код работал корректно?

1. function Car (model) {

2. this.model = model;

3. }

4.

5. Car.prototype.getInfo = function(){

6. return this;

7. }

8.

9. function Truck (model) {

10. this.model = model;

11. }

12.

13. Truck.prototype = Object.create(Car.prototype);

14. Truck.prototype.constructor = Truck;

15.

16. Truck.prototype.setInfo = function(prop, value) {

17. let carInfo = ??????

18. carInfo[prop] = value;

19. }

20.

21. let truck = new Truck("Volvo");

22.

23. truck.setInfo("color", "red");

25. truck.getInfo();

Car.prototype.getInfo.call(this);

Car.prototype.getInfo();

Car.getInfo.call(this);

Car.getInfol();