Частное учреждение образования

«Колледж бизнеса и права»

|  |
| --- |
| УТВЕРЖДАЮ  Заведующий  методическим кабинетом  \_\_\_\_\_\_\_\_\_\_Е.В.Паскал  «\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ |

|  |  |
| --- | --- |
| Специальность:  2-40 01 01 «Программное обеспечение информационных технологий» | Дисциплина: «Скриптовые языки программирования» |

**Лабораторная работа № 28**

**Инструкционно-технологическая карта**

Тема: Наследование классов. Особенности наследования в JavaScript.

Цель: Получить умения по использованию наследования классов.

Время выполнения: 2 часа

# Порядок выполнения работы

1. Изучить теоретические сведения к лабораторной работе.
2. Реализовать алгоритм решения задачи.
3. Отлаженную, работающую программу сдать преподавателю. Работу программы показать с помощью самостоятельно разработанных тестов.
4. Ответить на контрольные вопросы.

# Теоретические сведения

## Наследование классов

Например, есть два класса.

Animal:

class Animal {

    constructor(name) {

        this.speed = 0;

        this.name = name;

    }

    run(speed) {

        this.speed = speed;

        alert(`${this.name} бежит со скоростью ${this.speed}.`);

    }

    stop() {

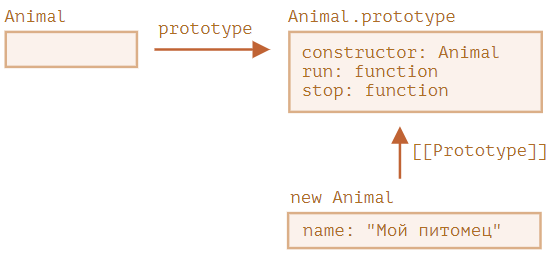
        this.speed = 0;

        alert(`${this.name} стоит.`);

    }

}

let animal = new Animal("Мой питомец");



И класс Rabbit:

class Rabbit {

    constructor(name) {

        this.name = name;

    }

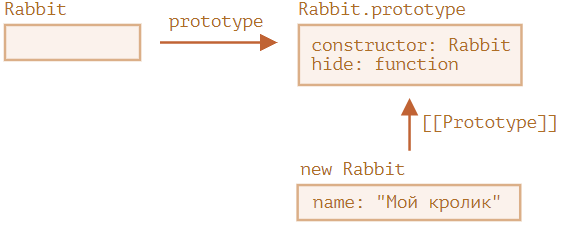
    hide() {

        alert(`${this.name} прячется!`);

    }

}

let rabbit = new Rabbit("Мой кролик");



Сейчас они полностью независимы.

Но необходимо, чтобы Rabbit расширял Animal. Другими словами, кролики должны происходить от животных, т.е. иметь доступ к методам Animal и расширять функциональность Animal своими методами.

Для того, чтобы наследовать класс от другого, нужно использовать ключевое слово «extends» и указать название родительского класса перед {..}.

Ниже Rabbit наследует от Animal:

class Animal {

    constructor(name) {

        this.speed = 0;

        this.name = name;

    }

    run(speed) {

        this.speed = speed;

        alert(`${this.name} бежит со скоростью ${this.speed}.`);

    }

    stop() {

        this.speed = 0;

        alert(`${this.name} стоит.`);

    }

}

// Наследуем от Animal указывая "extends Animal"

class Rabbit extends Animal {

    hide() {

        alert(`${this.name} прячется!`);

    }

}

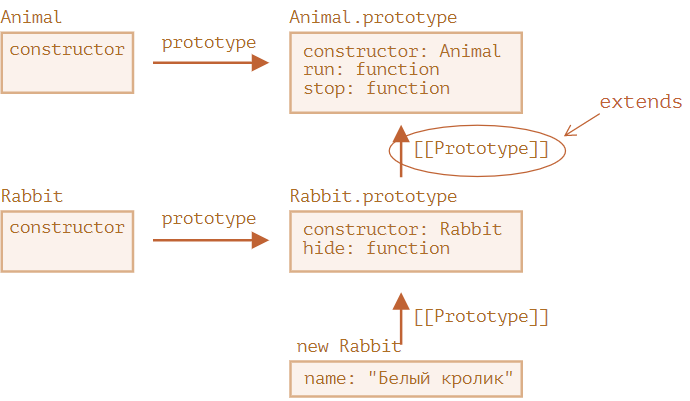
let rabbit = new Rabbit("Белый кролик");

rabbit.run(5); // Белый кролик бежит со скоростью 5.

rabbit.hide(); // Белый кролик прячется!

Теперь код **Rabbit** стал короче, так как используется конструктор класса **Animal** по умолчанию, и кролик может использовать метод **run** как и все животные.

Ключевое слово **extends** работает, используя прототипы. Оно устанавливает **Rabbit.prototype.[[Prototype]]** в **Animal.prototype**. Так что, если метод не найден в **Rabbit.prototype**, JavaScript берёт его из **Animal.prototype**.



## Переопределение методов

Сейчас **Rabbit** наследует от **Animal** метод **stop**, который устанавливает **this.speed = 0**.

Если переопределить метод **stop** в классе **Rabbit**, то он будет использоваться взамен родительского:

class Rabbit extends Animal {

    stop() {

        // ...будет использован для rabbit.stop()

    }

}

Впрочем, обычно нет необходимости полностью заменить родительский метод, а скорее необходимо сделать новый на его основе, изменяя или расширяя его функциональность. Для этого нужно написать что-то в дочернем методе и вызывать родительский метод до/после или в процессе.

У классов есть ключевое слово «**super**» для таких случаев.

* **super.method(...)** вызывает родительский метод.
* **super(...)** вызывает родительский конструктор (работает только внутри нашего конструктора).

Пусть кролик автоматически прячется при остановке:

class Animal {

    constructor(name) {

        this.speed = 0;

        this.name = name;

    }

    run(speed) {

        this.speed = speed;

        alert(`${this.name} бежит со скоростью ${this.speed}.`);

    }

    stop() {

        this.speed = 0;

        alert(`${this.name} стоит.`);

    }

}

class Rabbit extends Animal {

    hide() {

        alert(`${this.name} прячется!`);

    }

    stop() {

        super.stop(); // вызываем родительский метод stop

        this.hide(); // и затем hide

    }

}

let rabbit = new Rabbit("Белый кролик");

rabbit.run(5); // Белый кролик бежит со скоростью 5.

rabbit.stop(); // Белый кролик стоит. Белый кролик прячется!

Теперь у класса **Rabbit** есть метод **stop**, который вызывает родительский **super.stop()** в процессе выполнения.

# Контрольные вопросы

1. Какое ключевое слово используется для наследования?
2. Приведите пример наследования класса.
3. Как не переопределять метод класса, а дополнять?
4. Как вызвать родительский метод и родительский конструктор?

# Задания

Задание 1. Для класса, созданного в предыдущей лабораторной работе, создать класс-потомок с полями, указанными в индивидуальном задании

Реализовать в классе-потомке методы:

* конструктор;
* функцию обработки данных, указанную в индивидуальном задании;
* функцию формирования строки информации об объекте.

Создать проект для демонстрации работы: ввод и вывод информации об объектах: классе-родителе и классе-потомке.

| **№ вар.** | **Класс-родитель и его поля** | **Класс-потомок и его поля (поля класса-родителя выделены курсивом)** | **Функция-метод обработки данных объекта класса-потомка** |
| --- | --- | --- | --- |
| 1 | Дата (три числа): день, месяц, год | Список друзей: ФИО, телефон, *дата* рождения, | Количество дней до дня очередного рождения |
| 2 | Дата (три числа): день, месяц, год | Работник: ФИО, *дата* поступления на предприятие | Количество лет работы на предприятии |
| 3 | Дата (три числа): день, месяц, год | Лекарство: наименование, *дата* выпуска, фирма | Сколько прошло дней от изготовления лекарства |
| 4 | Время (три числа): часы, минуты, секунды | Расписание движения поездов: номер поезда, направление, *время* отправления | Количество минут до отправления поезда с указанным номером и введенное время |
| 5 | Время (три числа): часы, минуты, секунды | Абонент мобильной связи: фамилия, оператор, текущее *время* | Определить, является ли  время льготным для абонента (время от 0 до 8  часов) |
| 6 | Координаты изображения прямоугольника: *x*1, *y*1, *x*2, *y*2 | Изображение конверта (прямоугольник с линиями диагоналей):  *координаты прямо- угольника*, цвет линий | Площадь верхнего (над- диагонального) треугольника в пикселях |
| 7 | Время (три числа): часы, минуты, секунды | Расписание занятий: дисциплина, *время* начала, аудитория | Какая дисциплина по расписанию начинается в указанное время |
| 8 | Координаты изображения прямоугольника: *x*1, *y*1, *x*2, *y*2 | Изображение прямоугольника с вписанным в его центр кругом: *координаты прямоугольника*, радиус круга *R* (*R*< *x*2 – *x*1, *R*< *y*2 – *y*1) | Площадь фигуры между прямоугольником и кругом |
| 9 | Координаты изображения прямоугольника: *x*1, *y*1, *x*2, *y*2 | Изображение закрашенного прямоугольника с текстом: *координаты прямоугольника*, заданный текст, цвет закрашивания | Произведение периметра и длины диагонали прямоугольника в пиксе лях |
| 10 | Правильная дробь: числитель, знаменатель | Смешанная дробь: целая часть, ч*ислитель и знаменатель* | Представить смешанную дробь в виде десятичного вещественного числа. |
| 11 | Комплексное число: действительная (*a*1) и мнимая (*b*1) части числа | Два комплексных числа: *действительная (a1) и мнимая (b1) части первого числа;* действительная (*a*2) и мнимая (*b*2) части второго числа | Вычислить произведение двух комплексных чисел. |
| 12 | Комплексное число: действительная и мнимая часть числа | *Комплексное* сопротивление: сопротивление резистора (*действительная часть*), значение индуктивности (мнимая часть), угловая частота | Вычислить модуль и аргумент комплексной проводимости участка цепи «резистор - индуктивность». |
| 13 | Книга: название, количество страниц, цена | Библиотека: *название, количество страниц, цена*, скидка в процентах | Стоимость книги с учетом скидки. |
| 14 | Книга: название, автор, год издания | Книжный магазин: *название, автор, год издания*, цена | Уменьшить стоимость книги на 20%, если книге больше 5 лет. |
| 15 | Работник: фамилия, оклад, год поступления на работу | Работники предприятия: *фамилия, оклад, год поступления на работу,* год рождения | Определить, сколько лет нужно работать работнику до 60 лет, а если ему больше 60, то сколько лет он работает после 60 лет. |

Задание 2. Построить класс-родитель с указанными в индивидуальном задании полями и методами:

* конструктор;
* функция, которая определяет «качество» объекта – Q по заданной формуле;
* вывод информации об объекте.

Построить класс-потомок, который содержит:

* дополнительное поле P;
* функция, которая определяет «качество» объекта класса-потомка – Qp, которая перекрывает функцию качества класса-родителя (Q), выполняя вычисление по новой формуле.

Создать проект для демонстрации работы: ввод и вывод информации об объектах классов 1-го и 2-го уровней.

| **№**  **вар** | **Поля и функция «качества» (*Q* ) класса 1-го уровня** | **Поле и функция «качества» *Qp* класса 2-го уровня** |
| --- | --- | --- |
| 1 | Компьютер:   * наименование процессора; * тактовая частота процессора (МГц); * объем оперативной памяти (Мб).   *Q* = (0,1·частота) + память | *P*: объем винчестера (Гб) *Qp*=*Q*+0,5·*Р* |
| 2 | Оператор мобильной связи:   * название оператора; * стоимость 1 минуты разговора; * площадь покрытия.   *Q* = 100 · площадь покрытия / стоимость 1 минуты разговора | *P*: наличие платы за каждое соединение *Qp* = 0,7·Q, если *Р –* истина, иначе *Qp* = 1,5·*Q* |
| 3 | Товар на складе:   * наименование; * цена; * количество.   *Q* = цена / количество | *P*: год выпуска товара  *Qp* = *Q* + 0,5·(*Т–Р*),  где *Т* - текущий год |
| 4 | Кабель:   * тип; * количество жил кабеля; * диаметр.   *Q* = диаметр / количество жил | *P*: наличие оплетки  *Qp*: если *Р –* истина, то *Qp* = 2·*Q;* иначе *Qp* = 0,7·*Q* |
| 5 | Учебник по программированию:   * название; * количество страниц; * цена.   *Q* = цена / количество страниц | *P:* год издания  *Qp* = *Q* – 0,2· (*Т–Р*),  где *Т –* текущий год |
| 6 | Мобильный телефон:   * марка; * цена; * объем памяти.   *Q* = объем памяти / цена | *P*: количество SIM карт  *Qp* = *Q·Р* |
| 7 | Спутниковая антенна (тарелка):   * диаметр; * материал; * цена.   *Q* = диаметр / цена | *P*: тип подвески (азиму- тальная, полярная, торои- дальная)  *Qp*= *Q*, если тип подвески азимутальный,  *Qp* = 2·*Q*, если тип подвес- ки полярный,  *Qp* = 2,5·*Q*, если тип под- вески тороидальный |
| 8 | Экзамен:   * дисциплина; * число студентов на экзамене; * продолжительность экзамена (ч).   *Q* = число студентов / продолжительность | *P*: процент двоек  *Qp* = *Q*· ( 100-*Р*)/100 |
| 9 | Спортсмен:   * фамилия; * число соревнований; * сумма мест, занятых спортсменом в соревнованиях.   *Q* = (число соревнований)/ (сумма мест) | P: занимал ли хотя бы раз первое место  *Qp* = 1,5·*Q*, если *Р* - исти- на, иначе – *Qp*= *Q*, |
| 10 | Программист:   * фамилия; * число программ, написанных программистом; * число языков программирования, которыми он пишет программы.   Q = (число программ)\*(число языков) | *P*: число программ, кото- рые работают правильно  *Qp* = *Q* · *Р* / (число всех программ) |
| 11 | Спектакль:   * название; * *n*1 – число зрителей в начале; * *n*2 – число зрителей в конце.   *Q* = (*n*2 – *n*1)/ *n*1 | *P*: год написания пьесы  *Qp* = *Q* · (*Т-Р*+1),  где Т - текущий год |
| 12 | Алмаз:   * название; * вес (в каратах); * качество огранки в баллах (число).   *Q* = 0,4·вес + 0,6·качество огранки | *P*: цвет (белый, голубой, желтый, и тп)  *Qp*: если цвет голубой, то *Qp*=*Q+*1;  а если желтый,  то *Q*p = *Q* **–** 0,5 иначе *Qp*=*Q* |
| 13 | Компьютерная сеть:   * название организации; * число рабочих станций; * среднее расстояние между станциями (м).   *Q* = число станций · среднее расстояние | *P*: средняя скорость пере- дачи данных в сети (Мб/с)  *Qp* = *Q·Р* |
| 14 | Армия:   * вид войск; * численность (тыс человек): * вооруженность (баллы - число).   *Q*=0,3·численность+0,7·вооруженность | *P*: опыт (число месяцев, на протяжении которых армия вела боевые дей- ствия)  *Qp* = *Q* · (*Р*+1) |
| 15 | Автомобиль:   * марка автомобиля; * мощность двигателя (кВт); * число мест.   *Q* = 0,1·мощность·число мест | *P*: год изготовления  *Qp*=*Q*-1,5· (*Т*-*Р*) ,  где *Т* - текущий год |

# Литература

1. **Диков, А.В.** Клиентские технологии веб программирования: JavaScript и DOM: учебное пособие / А.В. Диков. – СПб: Лань, 2020 – 124 с.
2. **Читанамбри, Кирупа.** Изучаем React / Кирупа Читанамбри – 2-е изд. – М.: Эксмо, 2019. 368 с.
3. **Хавербеке, Марейн.** Выразительный JavaScript. Современное веб-программирование. / Марейн Хавербеке – 3-е изд. – СПб.: Питер, 2019. – 480 с.:

Преподаватель Рогалевич А.В.

Рассмотрено на заседании цикловой

комиссии программного обеспечения

информационных технологий №10

Протокол № \_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Председатель ЦК\_\_\_\_\_\_\_\_ В.Ю.Михалевич