ООП в функциональном стиле

Занятие 13

GO IT

goit.com.ua

ООП в функциональном стиле

GO IT

goit.com.ua

ООП

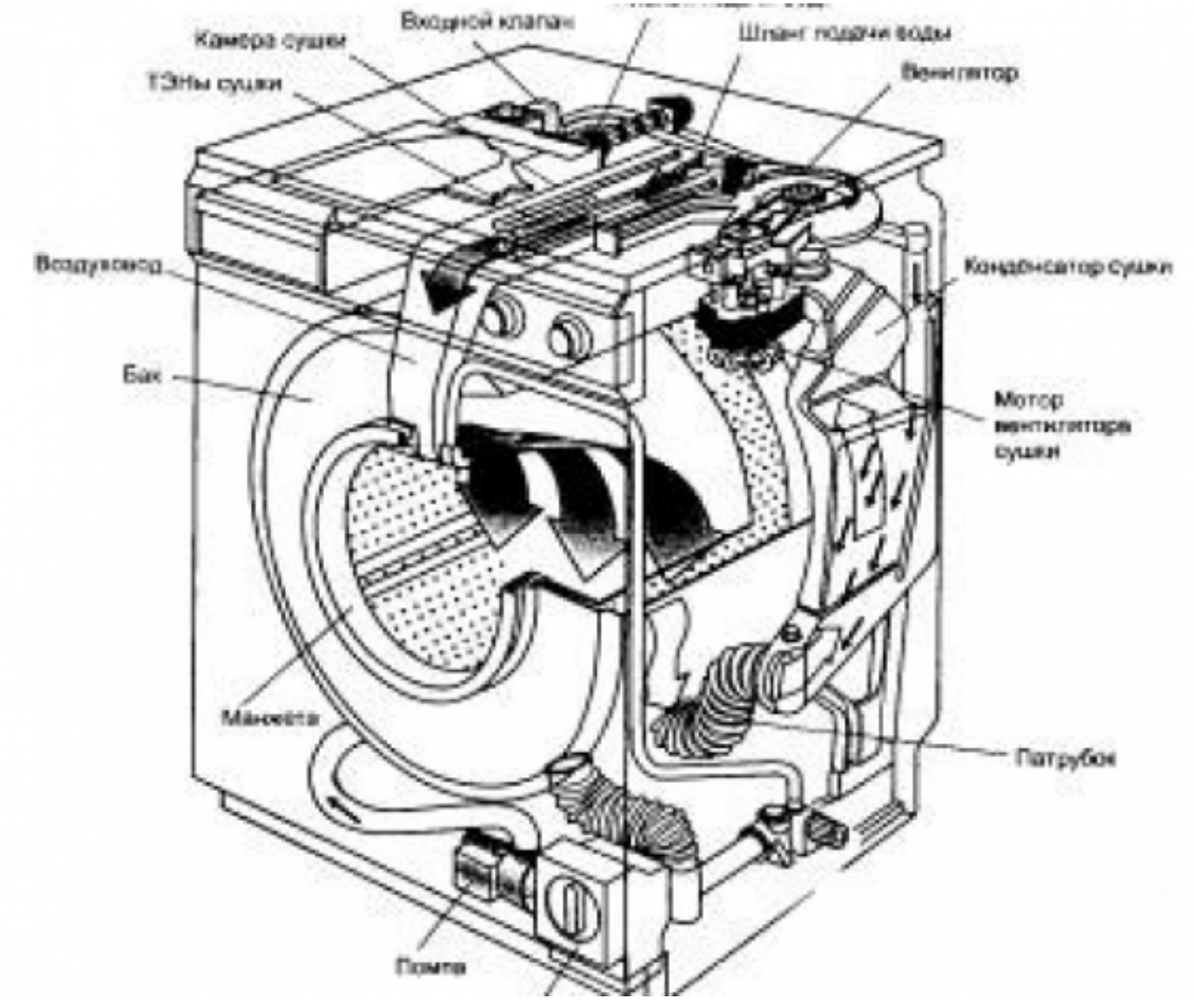
```
function User(name) {

    this.sayHi = function() {
        alert( "Привет, я " + name );
    };

}

var vasya = new User("Вася");
vasya.sayHi();
```

Интерфейсы

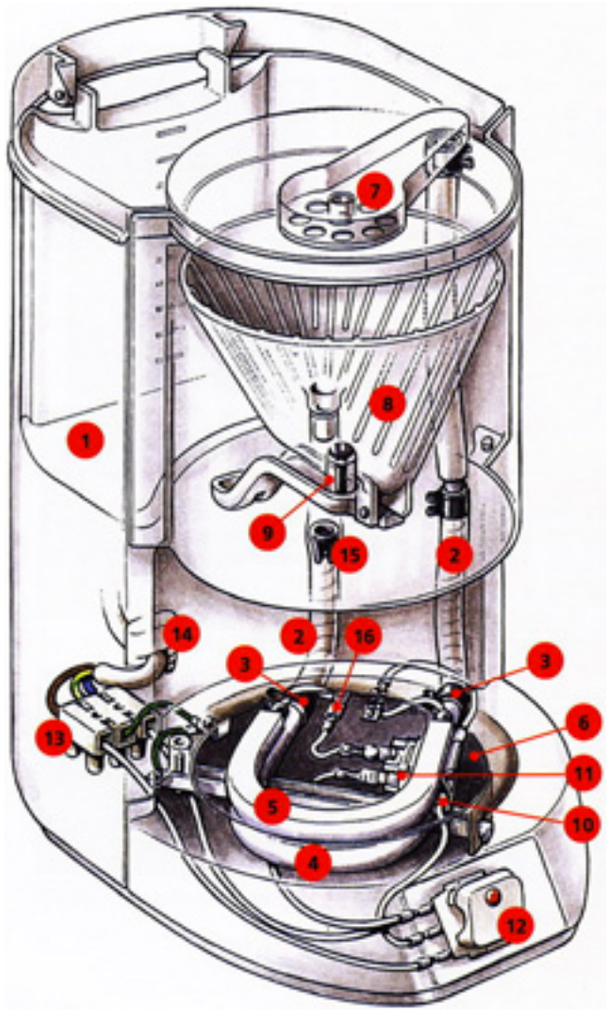
GO IT

goit.com.ua

Интерфейсы


Камера сушки
Входной клапан
Шланг подачи воды
ТЭНы сушки
Вентилятор
Воздуховод
Конденсатор сушки
Бак
Мотор вентилятора сушки
Манжета
Патрубок
Помпа

Интерфейсы



Интерфейсы

- Внутренний интерфейс
- Внешний интерфейс

Публичное и приватное свойство

```
function CoffeeMachine(power) {
    this.waterAmount = 0; // количество воды в кофеварке

    alert( 'Создана кофеварка мощностью: ' + power + ' ватт' );
}

// создать кофеварку
var coffeeMachine = new CoffeeMachine(100);

// залить воды
coffeeMachine.waterAmount = 200;
```

Публичный и приватный методы

```
function CoffeeMachine(power) {
// …
    // расчёт времени для кипячения
     function getBoilTime() {
         return 1000; // точная формула расчета будет позже
     }

     // что делать по окончании процесса
     function onReady() {
         alert( 'Кофе готово!' );
     }

     this.run = function() {
         // setTimeout – встроенная функция,
         // она запустит onReady через getBoilTime() миллисекунд
         setTimeout(onReady, getBoilTime());
     };
}

var coffeeMachine = new CoffeeMachine(100);
coffeeMachine.waterAmount = 200;
coffeeMachine.run();
```

Константа

```
function CoffeeMachine(power) {

    this.waterAmount = 0;

    // физическая константа – удельная теплоёмкость воды для getBoilTime
    var WATER_HEAT_CAPACITY = 4200;

    // расчёт времени для кипячения
    function getBoilTime() {
        return this.waterAmount * WATER_HEAT_CAPACITY * 80 / power; //ошибка!
    }

    // что делать по окончании процесса
    function onReady() {
        alert( 'Кофе готово!' );
    }

    this.run = function() {
        setTimeout(onReady, getBoilTime());
    };

}
```

Доступ к объекту из внутреннего метода

```
this.run = function() {
    setTimeout(onReady, getBoilTime.call(this));
};
```

Привязка через bind

```
function CoffeeMachine(power) {
    this.waterAmount = 0;

    var WATER_HEAT_CAPACITY = 4200;

    var getBoilTime = function() {
        return this.waterAmount * WATER_HEAT_CAPACITY * 80 / power;
    }.bind(this);

    function onReady() {
        alert( 'Кофе готово!' );
    }

    this.run = function() {
        setTimeout(onReady, getBoilTime());
    };
}
```

Сохранение this в замыкании

```
function CoffeeMachine(power) {
    this.waterAmount = 0;

    var WATER_HEAT_CAPACITY = 4200;

    var self = this;

    function getBoilTime() {
        return self.waterAmount * WATER_HEAT_CAPACITY * 80 / power;
    }

    function onReady() {
        alert( 'Кофе готово!' );
    }

    this.run = function() {
        setTimeout(onReady, getBoilTime());
    };
}
```

Геттеры и сеттеры

GO IT

goit.com.ua

Геттер и сеттер для воды

```
function CoffeeMachine(power) {
    // количество воды в кофеварке
    this.waterAmount = 0;

    //    ...
}

coffeeMachine.waterAmount = 1000000;

coffeeMachine.waterAmount -= 1000000;
```

```
function CoffeeMachine(power, capacity) { // capacity – ёмкость кофеварки
    var waterAmount = 0;
    var WATER_HEAT_CAPACITY = 4200;

    function getTimeToBoil() {
        return waterAmount * WATER_HEAT_CAPACITY * 80 / power;
    }

    // "умная" установка свойства
    this.setWaterAmount = function(amount) {
        if (amount < 0) {
            throw new Error("Значение должно быть положительным");
        }
        if (amount > capacity) {
            throw new Error("Нельзя залить воды больше, чем " + capacity);
        }
        waterAmount = amount;
    };

    function onReady() { alert('Кофе готов!' ); }

    this.run = function() {
        setTimeout(onReady, getTimeToBoil());
    };
}

var coffeeMachine = new CoffeeMachine(1000, 500);
coffeeMachine.setWaterAmount(600); // упс, ошибка!
```

getter

```
function CoffeeMachine(power, capacity) {
    //...
    this.setWaterAmount = function(amount) {
        if (amount < 0) {
            throw new Error("Значение должно быть положительным");
        }
        if (amount > capacity) {
            throw new Error("Нельзя залить воды больше, чем " + capacity);
        }

        waterAmount = amount;
    };

    this.getWaterAmount = function() {
        return waterAmount;
    };
}

var coffeeMachine = new CoffeeMachine(1000, 500);
coffeeMachine.setWaterAmount(450);
alert( coffeeMachine.getWaterAmount() ); // 450
```

Единый геттер-сеттер

```
function CoffeeMachine(power, capacity) {
    var waterAmount = 0;
    this.waterAmount = function(amount) {
        // вызов без параметра, значит режим геттера, возвращаем свойство
        if (!arguments.length) return waterAmount;

        // иначе режим сеттера
        if (amount < 0) {
            throw new Error("Значение должно быть положительным");
        }
        if (amount > capacity) {
            throw new Error("Нельзя залить воды больше, чем " + capacity);
        }
        waterAmount = amount;
    };
}

var coffeeMachine = new CoffeeMachine(1000, 500);
// пример использования
coffeeMachine.waterAmount(450);
alert( coffeeMachine.waterAmount() ); // 450
```

Задачка

Напишите конструктор User для создания объектов:

- С приватными свойствами имя firstName и фамилия surname.
- С сеттерами для этих свойств.
- С геттером getFullName(), который возвращает полное имя.

```
function User() {
    /* ваш код */
}

var user = new User();
user.setFirstName("Петя");
user.setSurname("Иванов");

alert( user.getFullName() ); // Петя Иванов
```

Функциональное наследование

GO IT

goit.com.ua

Наследование от Machine

```
function Machine() {
    var enabled = false;

    this.enable = function() {
        enabled = true;
    };

    this.disable = function() {
        enabled = false;
    };
}
```

Наследование от Machine

```
function CoffeeMachine(power) {
    Machine.call(this); // отнаследовать

    var waterAmount = 0;

    this.setWaterAmount = function(amount) {
        waterAmount = amount;
    };

}

var coffeeMachine = new CoffeeMachine(10000);

coffeeMachine.enable();
coffeeMachine.setWaterAmount(100);
coffeeMachine.disable();
```

Защищённые свойства

```
function Machine() {
    var enabled = false;

    this.enable = function() {
        enabled = true;
    };

    this.disable = function() {
        enabled = false;
    };
}

function CoffeeMachine(power) {
    Machine.call(this);

    this.enable();

    // ошибка, переменная не определена!
    alert( enabled );
}
```

Защищённые свойства

```
function Machine() {
    this._enabled = false; // вместо var enabled

    this.enable = function() {
        this._enabled = true;
    };

    this.disable = function() {
        this._enabled = false;
    };
}

function CoffeeMachine(power) {
    Machine.call(this);

    this.enable();

    alert( this._enabled ); // true
}

var coffeeMachine = new CoffeeMachine(10000);
```

Перенос свойства в защищённые

```
function Machine(power) {
    this._power = power; // (1)

    this._enabled = false;

    this.enable = function() {
        this._enabled = true;
    };

    this.disable = function() {
        this._enabled = false;
    };
}

function CoffeeMachine(power) {
    Machine.apply(this, arguments); // (2)

    alert( this._enabled ); // false
    alert( this._power ); // 10000
}

var coffeeMachine = new CoffeeMachine(10000);
```

2015