

WEBES ALKALMAZÁSFEJLESZTÉS 1.

Horváth Győző

Egyetemi adjunktus

1117 Budapest,

Pázmány Péter sétány 1/C, 2.420

Tel: (1) 372-2500/1816

Objektumok

2

- Szinte minden objektum
- Alapkonceptió – objektum-orientáltság
- Név-érték párok
 - ▣ Tulajdonság
 - ▣ Metódus

```
//objektum definiálása
var obj = {
  tulajdonsag: 1,
  'ez is tulajdonság': 2,
  metodus: function () {}
};
//hivatkozás
obj.tulajdonsag === 1
obj['ez is tulajdonság'] === 2
```

Két alappillér

3

- **Objektumok dinamikussága**
- Minden objektumhoz **prototípus-objektum** csatlakozik

4

Dinamikus objektumok

Dinamikus objektumok

5

- Tulajdonságaik tetszőlegesen módosíthatók

```
//Objektum megadása
```

```
var obj = {
```

```
  a: 1
```

```
};
```

```
//Objektum módosítása
```

```
obj.b = 2;
```

```
obj.b === 2
```

Tulajdonságleíró objektumok

6

- Objektumok tulajdonságainak finomhangolása
- ECMAScript 5 descriptor
 - ▣ **value**: a tulajdonság értéke (undefined)
 - ▣ **writable**: megváltoztatható-e értéke? (hamis)
 - ▣ **enumerable**: for..in / Object.keys() felsorolás (hamis)
 - ▣ **configurable**: törölhető-e, változtatható-e (hamis)
 - ▣ **set**: setter (undefined)
 - ▣ **get**: getter (undefined)

```
var descriptor = {  
  value: "érték",  
  writable: true,  
  enumerable: true,  
  configurable: true,  
  set: function(value) { valami = value },  
  get: function() { return test }  
};
```

Üres objektum létrehozása

7

```
//Üres objektumliterál
```

```
var obj = {};
```

```
Object.getPrototypeOf(obj) === Object.prototype
```

```
//Üres objektumliterállal kompatibilis objektum létrehozása
```

```
var obj = Object.create(Object.prototype);
```

```
Object.getPrototypeOf(obj) === Object.prototype
```

```
//Prototípus nélküli objektum létrehozása
```

```
var obj = Object.create(null);
```

```
Object.getPrototypeOf(obj) === null
```

Előre megadott tulajdonságok

8

```
//Minta
var obj = {
  alma: 'piros',
  'körte': 'sárga'
};

//Teszt
obj.alma === 'piros'
obj['körte'] === 'sarga'
```

```
//Objektumliterállal kompatibilis objektum
létrehozása előre megadott tulajdonságokkal
var obj = Object.create(Object.prototype, {
  alma: {
    value: 'piros',
    configurable: true,
    enumerable: true,
    writable: true
  },
  'körte': {
    value: 'sárga',
    configurable: true,
    enumerable: true,
    writable: true
  }
});

obj.alma === 'piros'
obj['körte'] === 'sárga'
```


Dinamikus tulajdonságok

9

- Műveletek
 - Hozzáadni
 - Lekérdezni
 - Módosítani
 - Törölni
- Eszközök
 - Hivatkozás (. operátor vagy [] elérés)
 - `Object.defineProperty(obj, név, leíró)`
 - `Object.defineProperties(obj, leírók)`
 - `Object.getOwnPropertyDescriptor(obj, név)`

Dinamikus tulajdonságok

10

```
//Üres objektum létrehozása
var obj = Object.create(Object.prototype);

//Tulajdonság létrehozása
Object.defineProperty(obj, 'a', {
  value: 1,
  writable: true,
  configurable: true,
  enumerable: true
});

//Leíróobjektum lekérdezése
var desc = Object.getOwnPropertyDescriptor(obj, 'a');

obj.a === 1
desc.value === 1
desc.writable === true
desc.configurable === true
desc.enumerable === true
```

Dinamikus tulajdonságok

11

```
//Tulajdonság módosítása és új létrehozása egyszerre
Object.defineProperty(obj, {
  a: {
    value: 42,
    enumerable: false
  },
  b: {
    value: 2
  }
});

var desc = Object.getOwnPropertyDescriptor(obj, 'a');

obj.b === 2
obj.a === 42
desc.writable === true
desc.configurable === true
desc.enumerable === false
```

Objektumszintű metódusok

12

- ❑ **Object.keys(obj):** az objektum felsorolható tulajdonságainak nevét tartalmazó tömb.
- ❑ **Object.getOwnPropertyNames(obj):** az objektum összes tulajdonságának nevét tartalmazó tömb.
- ❑ **Object.preventExtensions(obj):** megakadályozza új tulajdonság hozzáadását az objektumhoz.
- ❑ **Object.isExtensible(obj):** a fenti tulajdonság lekérdezése.
- ❑ **Object.seal(obj):** a tulajdonságok és leírók attribútumának változtatását tiltja le, kivéve az értékek megváltoztatását.
- ❑ **Object.isSealed(obj):** a fenti tulajdonság lekérdezése.
- ❑ **Object.freeze(obj):** az objektum befagyasztása, azaz semmiféle változtatás sem engedélyezett az objektumon.
- ❑ **Object.isFrozen(obj):** a fenti tulajdonság lekérdezése.

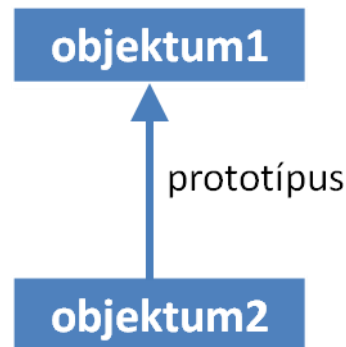
13

A prototípus-objektum

Prototípus-objektum

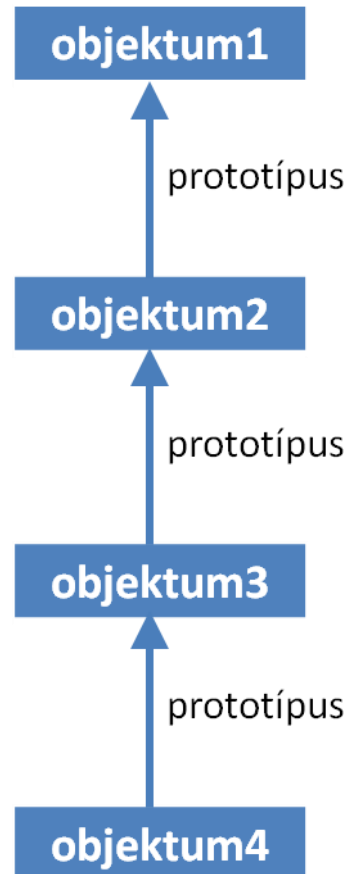
14

- Minden objektum tartalmaz egy rejtett hivatkozást egy másik objektumra
- → prototípus-objektum



A prototípuslánc

15



A prototípus beállítása, lekérdezése

16

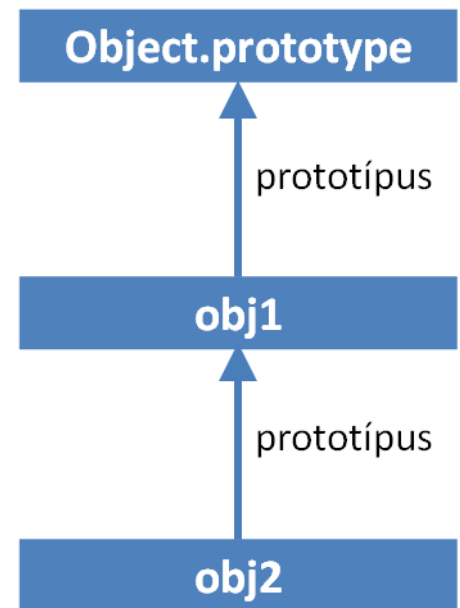
- Beállítás
 - **Object.create(protoObj)**
- Lekérdezés
 - **a.isPrototypeOf(b)**: visszaadja, hogy a szerepel-e b prototípusláncában.
 - **Object.getPrototypeOf(obj)**: megadja obj prototípus-objektumát.
 - **obj.__proto__**: a `__proto__` tulajdonság az obj prototípus-objektumára mutat (ES6).

A prototípus beállítása, lekérdezése

17

```
//A prototípuslánc kialakítása
var obj1 = Object.create(Object.prototype);
var obj2 = Object.create(obj1);

//Tesztek
obj2.__proto__ === obj1
obj1.__proto__ === Object.prototype
obj1.isPrototypeOf(obj2)
Object.prototype.isPrototypeOf(obj2)
Object.getPrototypeOf(obj2) === obj1
```



Object.create (ECMAScript 3)

18

```
if (!Object.create) {  
  Object.create = function (o) {  
    if (arguments.length > 1) {  
      throw new Error('Object.create implementation'  
        + ' only accepts the first parameter.');    }  
    function F() {}  
    F.prototype = o;  
    return new F();  
  };  
}
```

Tulajdonság lekérdezése (olvasás)

19

A prototípuslánc

obj2 látható interfésze

Object.prototype

toString()

↑
prototípus

obj1

a = 1

b = 2

↑
prototípus

obj2

b = 22

c = 3

obj2

a = 1

b = 22

c = 3

toString()

```
//A prototípuslánc létrehozása  
előre feltöltött objektumokkal
```

```
var o1 = {
```

```
  a: 1,
```

```
  b: 2
```

```
};
```

```
var o2 = Object.create(o1);
```

```
o2.b = 22;
```

```
o2.c = 3;
```

```
//Teszt
```

```
o2.c === 3
```

```
o2.a === 1
```

```
typeof o2.toString === 'function'
```

```
o2.b === 22
```

Tulajdonságok felsorolása

20

- Saját tulajdonságok
 - ▣ **Object.keys(obj)**
 - ▣ **Object.getOwnPropertyNames(obj)**
 - ▣ **obj.hasOwnProperty(név)**
- Tulajdonságok (prototípuslánc)
 - ▣ **for..in ciklus**

Tulajdonság beállítása (írás)

21

```
//A prototípuslánc létrehozása  
előre feltöltött objektumokkal
```

```
var o1 = {  
  a: 1,  
  b: 2  
};
```

```
var o2 = Object.create(o1);
```

```
//Teszt
```

```
o2.b === 2
```

```
o2.hasOwnProperty('b') === false
```

```
//Írás o2-be
```

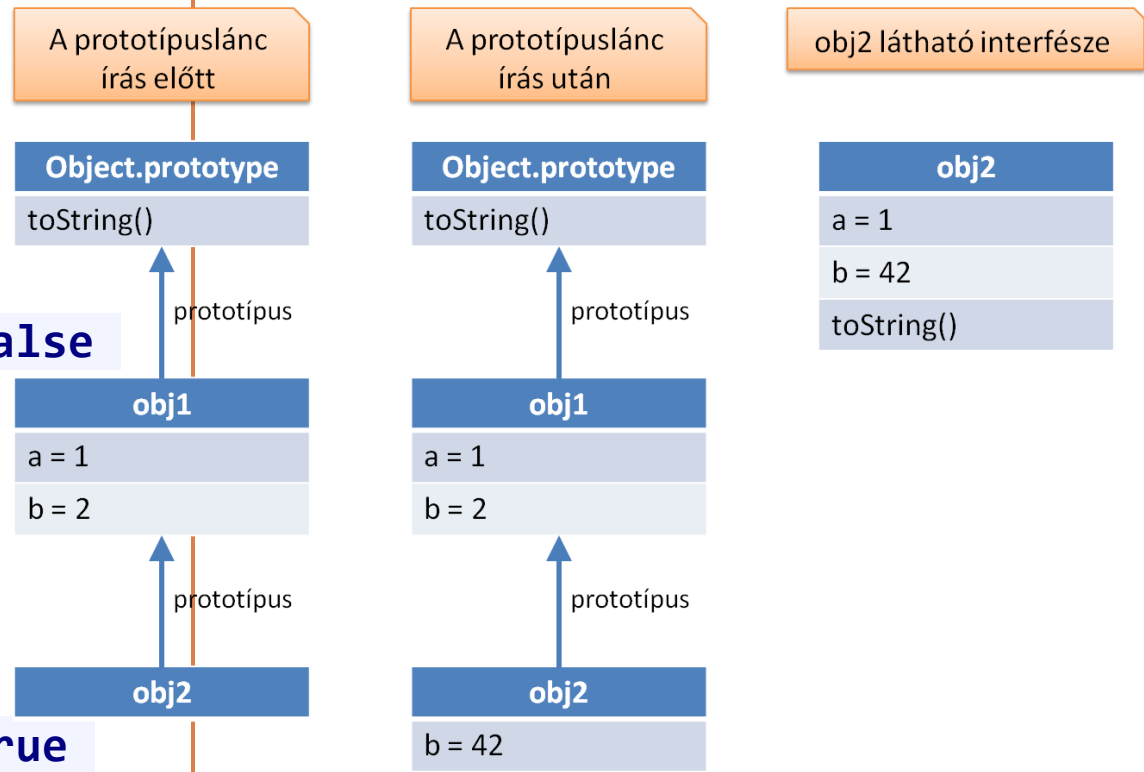
```
o2.b = 42;
```

```
//Teszt
```

```
o2.b === 42
```

```
o2.hasOwnProperty('b') === true
```

```
o1.b === 2
```



22

Kód-újrhasználtság dinamikus objektumokkal

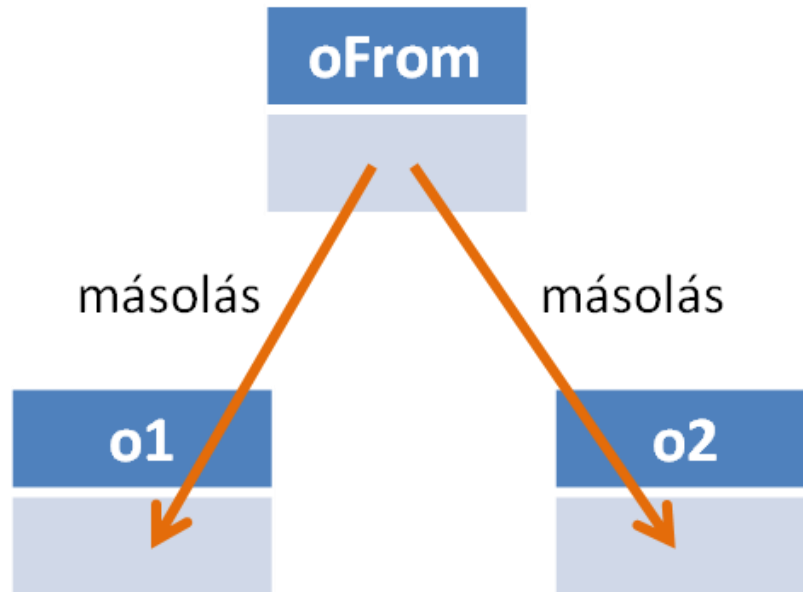
Prototípusosság

23

- Prototípus = modell objektum
- A JavaScript prototípusos nyelv
- Prototípusosság
 - ▣ dinamikus objektumokkal
 - ▣ prototípus-objektumokkal
- → kód-újrahasznosítás eszköze

Sematikus ábra

24



Tulajdonságok másolása – bővítés

25

- bővítés
- mixin

```
//Kiindulási objektumok
var o1 = {
  a: 1,
  b: 2
};
var o2 = {
  b: 42,
  c: 3
};

//Mixin
o2.a = o1.a;
o2.b = o1.b;

//Teszt
o2.a === 1
o2.b === 2
o2.c === 3
```

extendShallow()

26

```
var extendShallow = function extendShallow(objTo, objFrom) {
  if (arguments[2] && typeof arguments[2] === 'string') {
    for (var i = 2; i < arguments.length; i++) {
      var propName = arguments[i];
      objTo[propName] = objFrom[propName];
    }
  } else {
    [].slice.call(arguments, 1).forEach(function(source) {
      for (var prop in source) {
        if (source.hasOwnProperty(prop)) {
          objTo[prop] = source[prop];
        }
      }
    });
  }
  return objTo;
};
```

extendShallow()

27

```
//Kiindulási objektumok  
var o1 = {  
  a: 1,  
  b: 2  
};  
var o2 = {  
  b: 42,  
  c: 3  
};  
extendShallow(o2, o1);
```

```
var o3 = extendShallow({}, o2, o1);
```

```
extendShallow(o2, o1, 'a');
```

```
o2.a === 1  
o2.b === 42
```

Kiindulási állapot

obj1
a = 1
b = 2

extendShallow

Végállapot

obj2
a = 1
b = 2
c = 3

obj2
b = 42
c = 3

Referenciatípusok másolása

28

```
//A kiindulási objektumok
```

```
var oFrom = {  
  arr: [],  
  obj: {  
    a: 1  
  },  
  put: function (elem) {  
    this.arr.push(elem);  
  }  
};
```

```
//Célobjektumok
```

```
var o1 = extendShallow({}, oFrom);  
var o2 = extendShallow({}, oFrom);
```

```
//Manipulálás
```

```
o1.put(42);  
o1.obj.a = 100;
```

```
//Teszt
```

```
o1.put === o2.put  
o1.arr === o2.arr  
o1.obj === o2.obj  
o2.arr[0] === 42  
o2.obj.a === 100
```

```
//Teljes objektumcsere
```

```
o1.obj = {  
  b: 2  
};
```

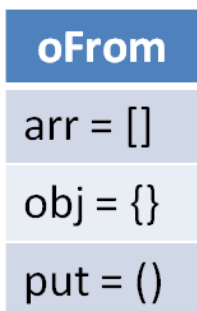
```
o2.obj.a === 100  
o1.obj.a === undefined
```

Referenciatípusok másolása

29

Kiindulási állapot

Végállapot

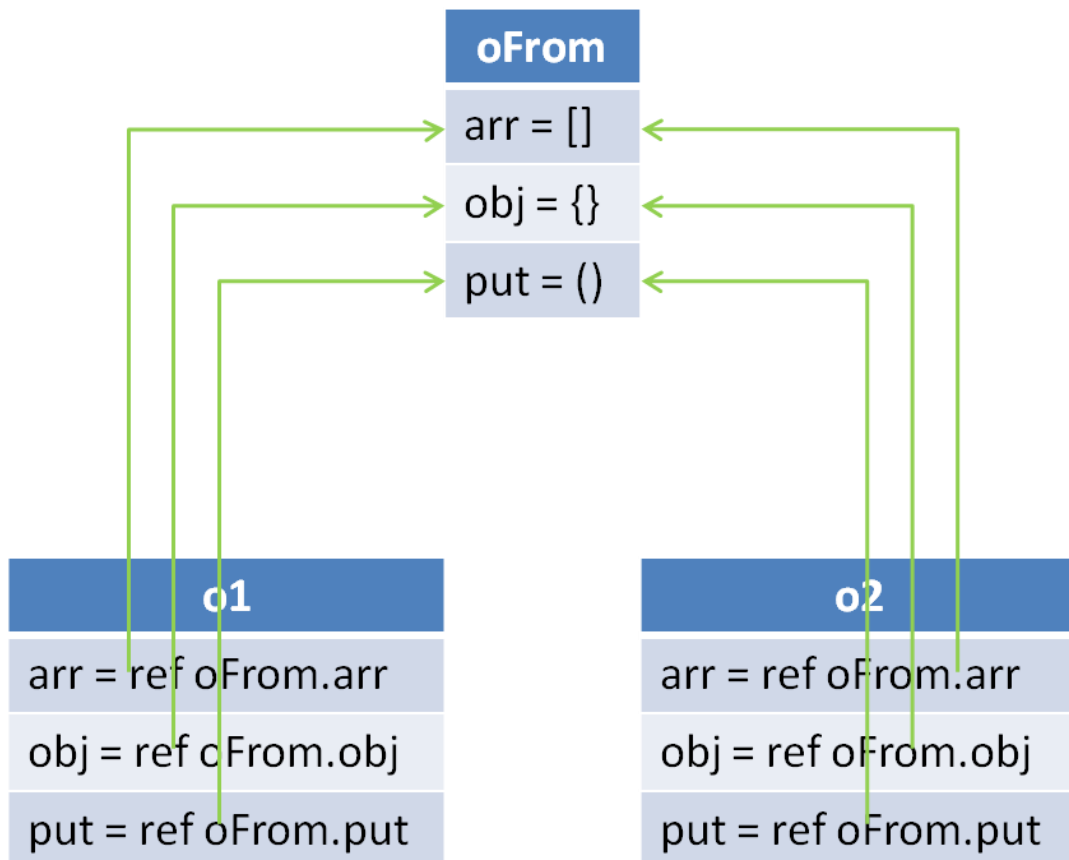


extendShallow

extendShallow

o1

o2



extendDeep()

```
var extendDeep = function extendDeep(objTo, objFrom) {
  var copy = function copy(objTo, objFrom, prop) {
    if (typeof objFrom[prop] === "object") {
      objTo[prop] =
        (Object.prototype.toString.call(objFrom[prop]) === '[object Array]') ? [] : {};
      extendDeep(objTo[prop], objFrom[prop]);
    } else {
      objTo[prop] = objFrom[prop];
    }
  };

  if (arguments[2] && typeof arguments[2] === 'string') {
    for (var i = 2; i < arguments.length; i++) {
      var prop = arguments[i];
      copy(objTo, objFrom, prop);
    }
  } else {
    [].slice.call(arguments, 1).forEach(function(source) {
      for (var prop in source) {
        if (source.hasOwnProperty(prop)) {
          copy(objTo, source, prop);
        }
      }
    });
  }

  return objTo;
};
```

extendDeep()

31

```
//Használata a legutóbbi oFrom, o1  
és o2 objektumokon
```

```
var o1 = extendDeep({}, oFrom);  
var o2 = extendDeep({}, oFrom);
```

```
//Manipulálás
```

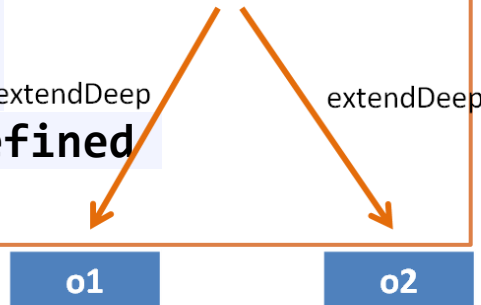
```
o1.put(42);  
o1.obj.a = 100;
```

```
//Teszt
```

```
o1.put === o2.put  
o1.arr !== o2.arr  
o1.obj !== o2.obj  
o2.arr[0] === undefined  
o2.obj.a === 1
```

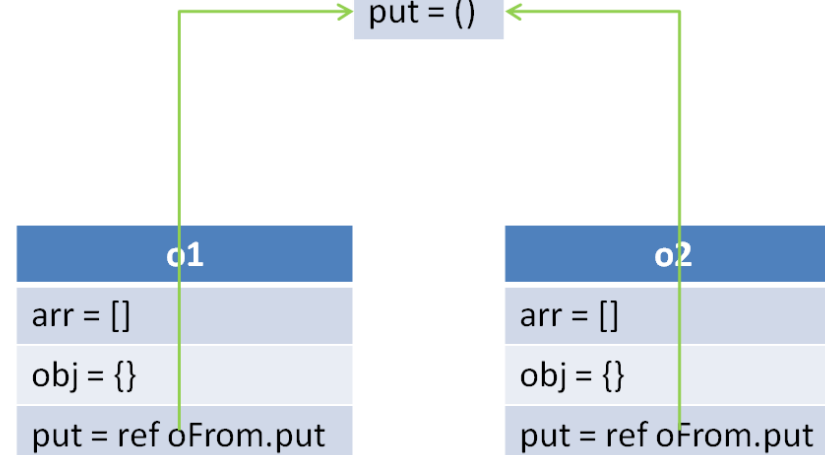
Kiindulási állapot

oFrom
arr = []
obj = {}
put = ()



Végállapot

oFrom
arr = []
obj = {}
put = ()



Funkcionális bővítés

32

```
var funcFrom = function funcFrom() {  
  this.arr = [];  
  this.obj = {  
    a: 1  
  };  
  this.put = function (elem) {  
    this.arr.push(elem);  
  };  
  return this;  
};  
var o1 = funcFrom.call({});  
var o2 = funcFrom.call({});  
  
//Teszt  
o1.put !== o2.put  
o1.arr !== o2.arr  
o1.obj !== o2.obj
```

Kiindulási állapot

funcFrom
arr = []
obj = {}
put = ()

extendFunc

o1

o2

Végállapot

funcFrom
arr = []
obj = {}
put = ()

o1

arr = []
obj = {}
put = ()

o2

arr = []
obj = {}
put = ()

extendFunc()

33

```
var extendFunc = function extendFunc(obj) {  
  [].slice.call(arguments, 1).forEach(function(source) {  
    if (typeof source === 'function') {  
      source.call(obj);  
    }  
  });  
  return obj;  
};
```

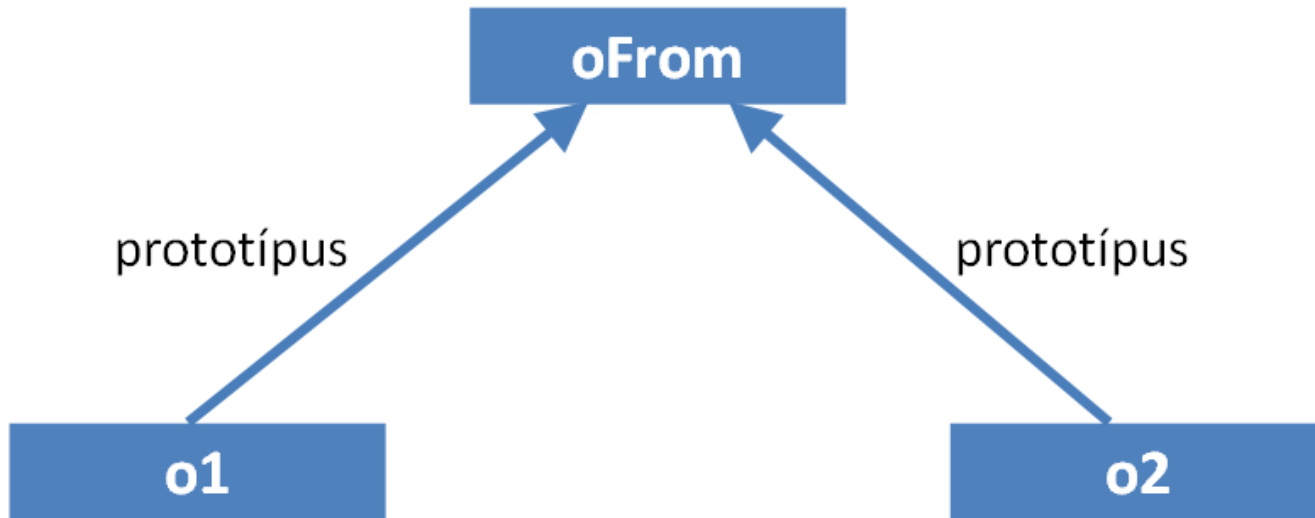
```
//Másolandó funkcionalitások  
var func1 = function func1() {  
  extendShallow(this, {  
    random: function (n) {  
      return Math.floor(Math.random() * n);  
    }  
  });  
};  
var func2 = function func2() {  
  extendShallow(this, {  
    add: function (a, b) {  
      return a + b;  
    }  
  });  
};  
//Célobjektumok létrehozása  
var o1 = extendFunc({}, func1, func2);  
var o2 = extendFunc({}, func1, func2);  
//Teszt  
typeof o1.random === 'function'  
typeof o2.add === 'function'  
o1.random !== o2.random
```

34

Kód-újrahasznosítás prototípus- objektummal

Sematikus ábra

35



Prototípusosság

```
//Prototípus
var oFrom = {
  a: 1,
  obj: {
    l: true
  },
  hello: function () {
    return 'hello';
  }
};

//Célobjektumok
var o1 = Object.create(oFrom);
var o2 = Object.create(oFrom);

o1.a === 1
o1.hello() === 'hello'
```

```
//Írás a célobjektumokba
o1.b = 2;
o2.a = 11;

o1.b === 2
o2.a === 11
o1.hello === o2.hello

//Összetett adatszerkezetek
o1.obj.l = false;

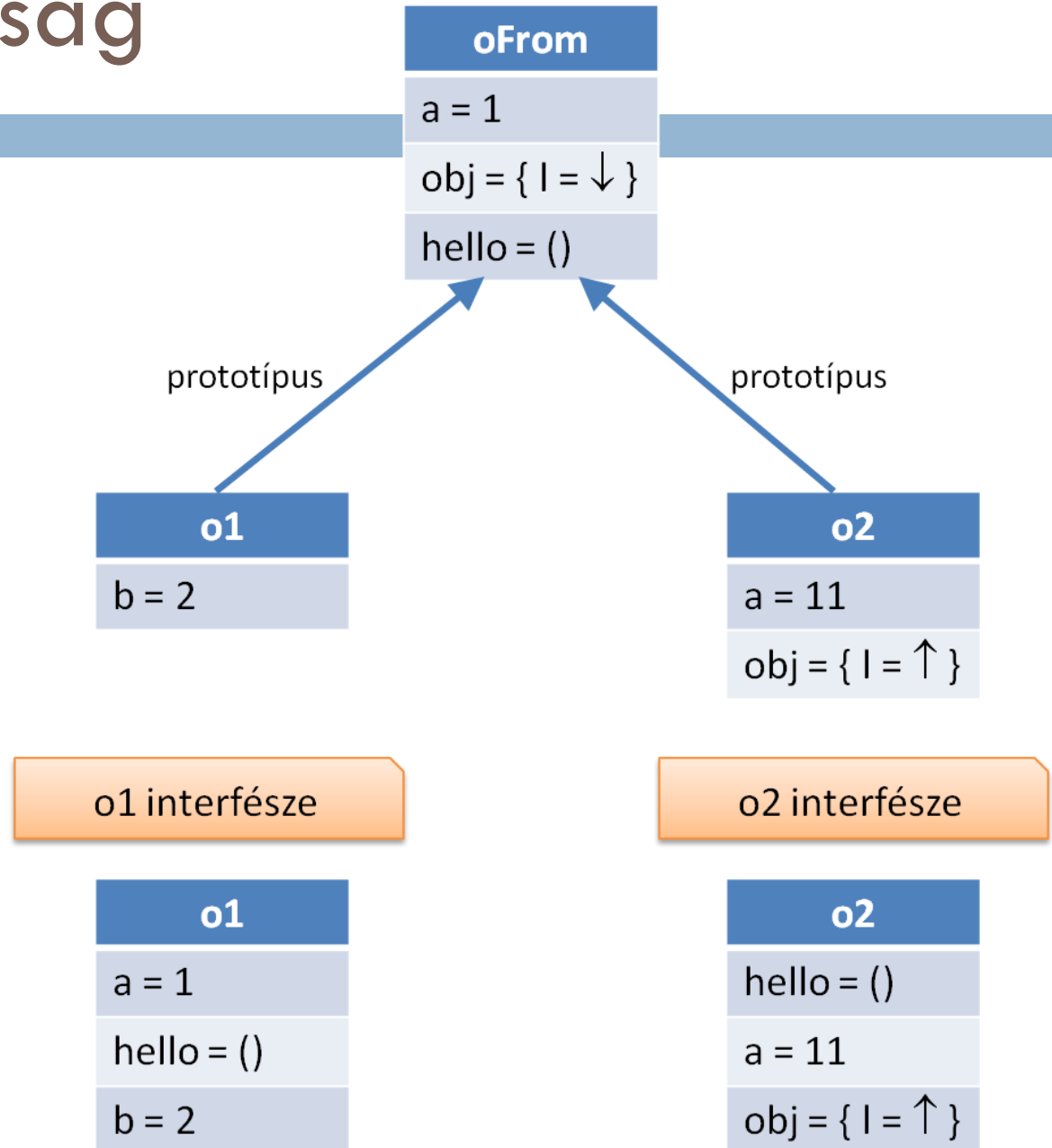
ok( o1.obj === o2.obj )
ok( o2.obj.l === false )

o2.obj = {
  l: true
};

o1.obj !== o2.obj
o2.obj.l !== o1.obj.l
```

Prototípusosság

37



Objektum-létrehozási minták

Egy objektum létrehozása

39

- Objektumliterál
- `Object.create()`;

```
var c = {  
  name: 'Sári',  
  dateOfBirth: {  
    year: 2004,  
    month: 11,  
    day: 14  
  },  
  getName: function getName() {  
    return this.name;  
  },  
  setName: function setName(name) {  
    this.name = name;  
  }  
};
```

Objektumgyár (Factory)

40

```
var child = function child() {  
  return {  
    name: 'Anonymous',  
    dateOfBirth: {  
      year: 1970,  
      month: 1,  
      day: 1  
    },  
    getName: function getName() {  
      return this.name;  
    },  
    setName: function setName(name) {  
      this.name = name;  
    }  
  };  
};
```

```
var c1 = child();  
var c2 = child();  
  
c1 !== c2  
c1.dateOfBirth !== c2.dateOfBirth  
c1.getName !== c2.getName
```


Objektumgyár

41

```
var child = {  
  create: function create() {  
    return {  
      name: /*...*/,  
      dateOfBirth: /*...*/,  
      getName: /*...*/,  
      setName: /*...*/  
    };  
  }  
};  
  
var c1 = child.create();  
var c2 = child.create();
```

Paraméteres gyárfüggvény

42

```
var child = {  
  create: function create(props) {  
    return extendDeep({  
      name: /*...*/,  
      dateOfBirth: /*...*/,  
      getName: /*...*/,  
      setName: /*...*/  
    }, props || {});  
  }  
};
```

```
var c1 = child.create();  
var c2 = child.create({  
  name: 'Zsófi',  
  dateOfBirth: {  
    year: 2006,  
    month: 9,  
    day: 1  
  }  
});  
  
c1.name === 'Anonymous'  
c2.name === 'Zsófi'
```

Privát és privilegizált adattagok és metódusok

43

```
var child = {
  create: function create(props) {
    //Privát adattag
    var secretNickName = '';

    return extendDeep({
      //Publikus adattagok és metódusok
      name: 'Anonymous',
      dateOfBirth: { /* ... */ },
      getName: /* ... */,
      setName: /* ... */,
      //Privilegizált metódusok
      setSecretNickName: function (name) {
        secretNickName = name;
      },
      getSecretNickName: function () {
        return secretNickName;
      }
    }, props || {});
  }
};
```

```
var c1 = child.create();
c1.setSecretNickName('Fairy');

c1.secretNickName === undefined
c1.getSecretNickName() === 'Fairy'
```

Metódusok hatékony tárolása

44

Elvárt interfész

c1
name = ""
dateOfBirth = {}
setName = ()
getName = ()

c2
name = ""
dateOfBirth = {}
setName = ()
getName = ()

A tárolás sematikus módja

metódusok
setName = ()
getName = ()

c1
name = ""
dateOfBirth = {}

c2
name = ""
dateOfBirth = {}

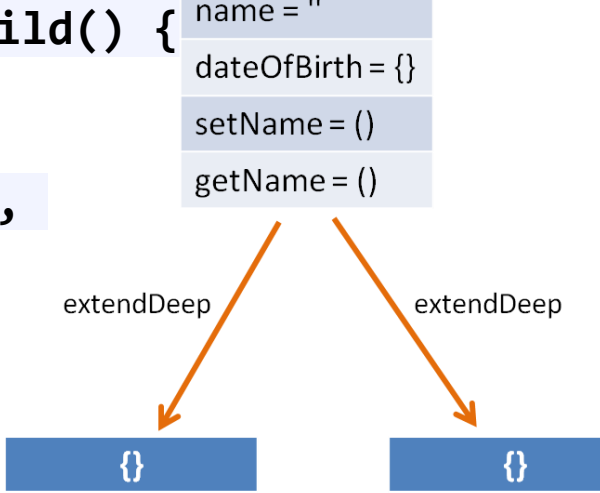
Hatékony tárolás bővítéssel

45

Üres objektum bővítése

Létrejött objektumok

```
var child = (function child() {  
  var childProto = {  
    name: /*...*/,  
    dateOfBirth: /*...*/,  
    getName: /*...*/,  
    setName: /*...*/  
  };  
  
  return {  
    create: function create(props) {  
      return extendDeep({}, childProto, props);  
    }  
  }  
})();
```



c1

```
name = ""  
dateOfBirth = {}  
setName = ref childProto.setName  
getName = ref childProto.getName
```

c2

```
name = ""  
dateOfBirth = {}  
setName = ref childProto.setName  
getName = ref childProto.getName
```

Hatékony tárolás prototípusláncsal

46

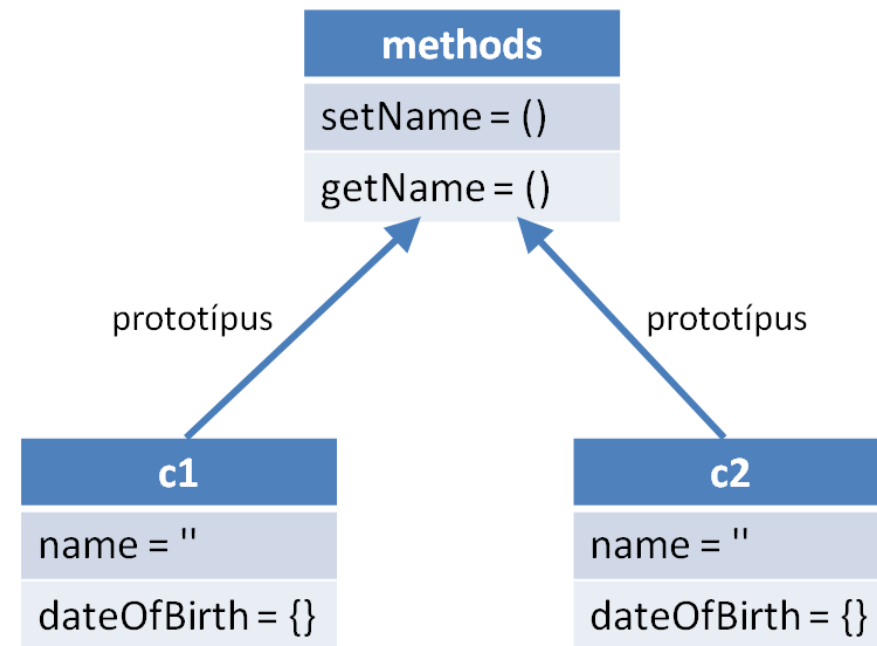
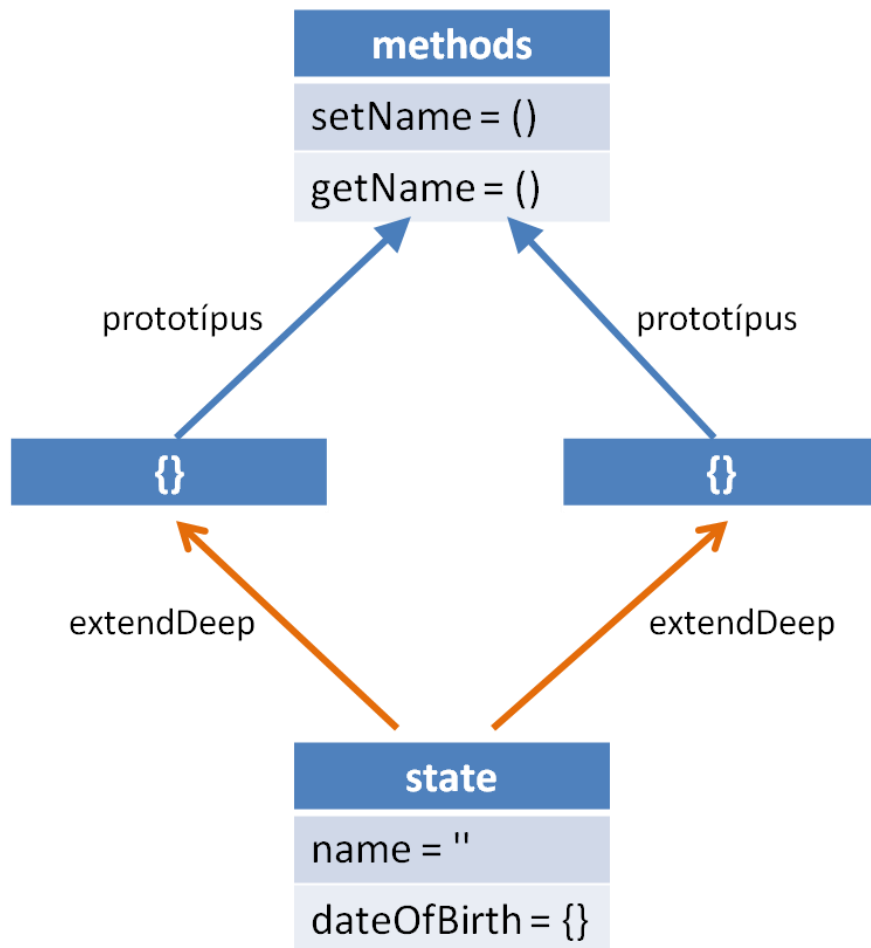
```
var child = (function() {  
  var methods = {  
    getName: /*...*/,  
    setName: /*...*/  
  };  
  var data = {  
    name: /*...*/,  
    dateOfBirth: /*...*/  
  };  
  return {  
    create: function(props) {  
      return extendDeep(  
        Object.create(methods),  
        data,  
        props || {}  
      );  
    },  
    methods: methods  
  };  
})();
```

Hatékony tárolás prototípusláncsal

47

Prototípuslánc és bővítés

Létrejött objektumok



Hatékony tárolás és privát adattagok

48

- Privilegizált metódusokat nem lehet hatékonyan tárolni
- Privát adattag = closure = függvény
- → privilegizált metódusnak is a closure-t biztosító függvényben van a helye
- → annyi példány, ahány objektum

49

Öröklési minták

Sematikus ábra

50

Elvárt interfész

p1	p2
name = ""	name = ""
dateOfBirth = {}	dateOfBirth = {}
setName = ()	setName = ()
getName = ()	getName = ()
sign = ""	sign = ""
setSign = ()	setSign = ()
getSign = ()	getSign = ()

A tárolás sematikus módja

szülő metódusok

setName = ()
getName = ()

gyerek metódusok

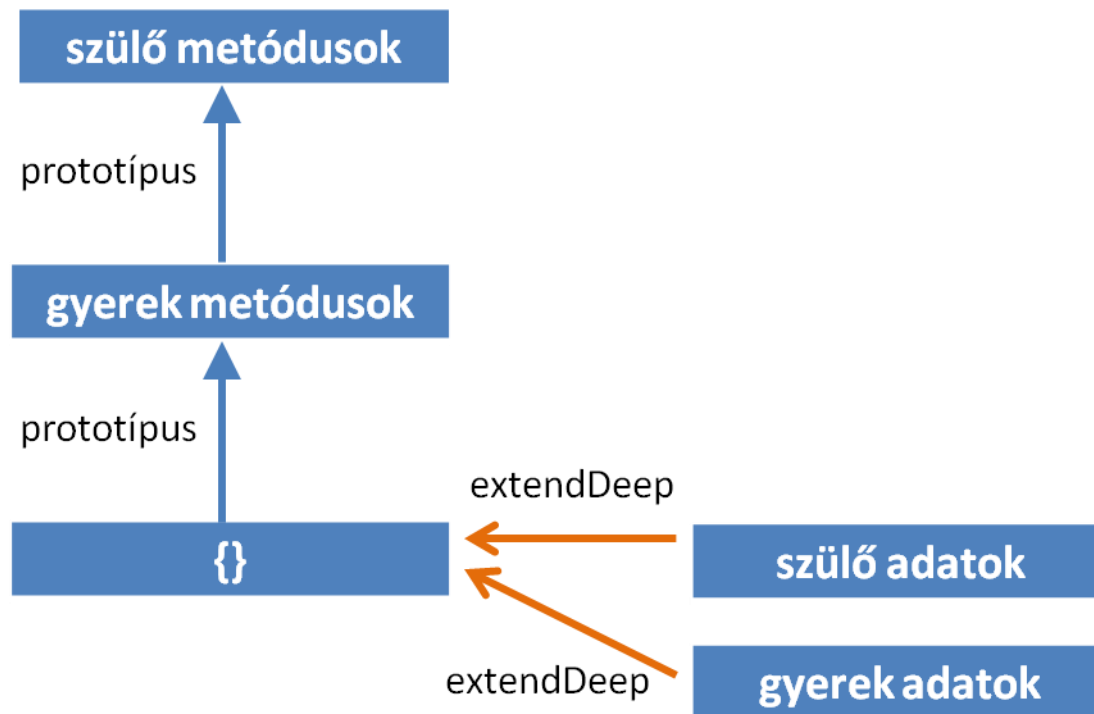
setSign = ()
getSign = ()

p1	p2
name = ""	name = ""
dateOfBirth = {}	dateOfBirth = {}
sign = ""	sign = ""

Öröklés prototípuslánccal

51

Prototípuslánc és bővítés



Öröklés prototípusláncsal

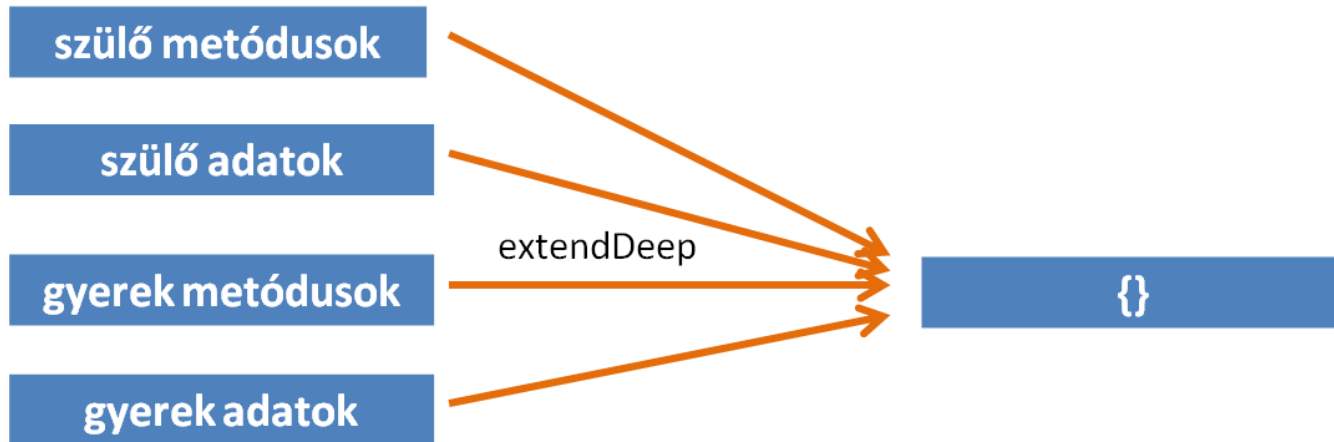
```
var preschool = (function(_super) {  
  var methods = {  
    getSign: function getSign() {  
      return this.sign;  
    },  
    setSign: function setSign(sign) {  
      this.sign = sign;  
    },  
    getName: function getName() {  
      var name = this._super.getName.call(this);  
      return name + ' (preschool)';  
    }  
  };  
};  
var publicMethods = extendShallow(  
  Object.create(_super.methods),  
  methods,  
  {  
    _super: _super.methods  
  }  
);  
var publicData = {  
  sign: 'default sign'  
};
```

```
return {  
  create: function(props) {  
    return extendDeep(  
      Object.create(publicMethods),  
      _super.create(),  
      publicData,  
      props || {}  
    );  
  },  
  methods: publicMethods  
};  
})(child);
```

Öröklés bővítéssel

53

Kompozíció bővítéssel



Eredmény

obj
ref szülő metódusok
szülő adatok
ref gyerek metódusok
gyerek adatok

Öröklés bővítéssel

54

```
var childProto = {
  name: /*...*/,
  dateOfBirth: /*...*/,
  getName: /*...*/,
  setName: /*...*/
};
var preschoolProto = {
  sign: /*...*/,
  getSign: /*...*/,
  setSign: /*...*/
};
var preschool = {
  create: function(props) {
    return extendDeep({}, childProto, preschoolProto, props);
  }
};
```

55

Gyárfüggvény-generátorok

Gyárfüggvény

56

- publikus adatok (alapértelmezett értékek)
- publikus metódusok
- privát adatok és privilegizált metódusok
- esetleges szülő-objektumot.

Modellobjektumok

```
var childProto = {
  methods: {
    getName: /*...*/,
    setName: /*...*/
  },
  data: {
    name: /*...*/,
    dateOfBirth: /*...*/
  },
  init: function () {
    //Privát adattag
    var secretNickName = /*...*/;
    //Privilegizált metódusok
    var privilegedMethods = {
      setSecretNickName: /*...*/,
      getSecretNickName: /*...*/
    };

    return extendShallow(this, privilegedMethods);
  }
};
```

```
var preschoolProto = {
  methods: {
    getSign: /*...*/,
    setSign: /*...*/,
    getName: /*...*/
  },
  data: {
    sign: 'car'
  },
  super: child
};
```

```
var createFactoryWithPrototype = function createFactoryWithPrototype(opts) {  
  //Paraméterek kiolvasása  
  var methods  = opts.methods || {};  
  var publicData = opts.data  || {};  
  var _super   = opts.super;  
  var init     = opts.init    || function () {};  
  
  var publicMethods = extendShallow(  
    Object.create(_super ? _super.methods : Object.prototype),  
    methods,  
    _super ? {  
      _super: _super.methods  
    } : {}  
  );  
  
  return {  
    create: function(props) {  
      var obj = extendDeep(  
        Object.create(publicMethods),  
        _super ? _super.create() : {},  
        publicData,  
        props || {}  
      );  
      return extendFunc(obj, init);  
    },  
    methods: publicMethods  
  };  
};
```

Bővítés

59

```
var createFactoryWithComposition = function createFactoryWithComposition() {
  var args = arguments;
  var methods = {};
  for (var i = 0; i < arguments.length; i++) {
    extendDeep(methods, args[i].methods || {});
  };
  return {
    create: function(props) {
      var obj = Object.create(methods);
      for (var i = 0; i < args.length; i++) {
        extendDeep(obj, args[i].data || {});
        extendFunc(obj, args[i].init || function () {});
      };
      return extendDeep(obj, props);
    }
  };
};
```