

Tema. Objektai

1 užduotis

JavaScript kalboje objektai yra savybių ir joms priskirtų reikšmių ar rinkinių. objektus galime kurti naudodami JASON (JavaScript object notation) notaciją. Pavyzdžiui, duotas programos kodas:

```
var objektas1 = {  
  "savybė1": 1,  
  "savybė2": 2,  
  "metodas1": function ()  
  {  
    console.log("labas");  
  }  
}  
  
console.log(objektas1["savybė1"]);  
console.log(objektas1.savybė2);  
objektas1.metodas1();
```

čia, sukuriamas kintamasis vardu **objektas1** ir jam priskiriamas JASON notacija nusakytas objektas, kuris:

- turi savybę **savybė1** kuriam priskirta **number** duomenų tipo reikšmė **1**;
- turi savybę **savybė2** kuriam priskirta **number** duomenų tipo reikšmė **2**;
- turi savybę **metodas1** kuriai priskirta funkcija, kuri išveda į konsolę simbolių eilutę **labas**.

Objekto savybės kurioms priskirtos funkcijos yra priimta vadinti objekto **metodais**. Objekto savybėms priskirtas reikšmės pasiekiamas nurodant objekto vardą ir laužtiniuose skliaustuose kabutėmis apskliaustą savybės vardą. Objekto savybės galime pasiekti ir po objekto vardo dėdami tašką ir rašydami savybės vardą, tačiau taip pasiekiamos tik tos savybės kurių vardai atitinka kintamųjų vardams taikomus reikalavimus (prasideda raide, neturi tarpo ir kitų draudžiamų simbolių ir t.t.).

Papildykite programos kodą:

- sukurkite kintamąjį **korta**;
- priskirkite jam JASON notacija nusakytą objektą, kuris:
 - turi savybę **rūšis**, jai priskirta simbolių eilutė (duomenų tipas **string**) **širdys**;
 - turi savybę **simbolis**, jai priskirta simbolių eilutė (duomenų tipas **string**) **Q**;

- turi **metoda** (savybę kuriai priskirta funkcija) vardu **pasiveikinti** kuris į konsolę išveda simbolių eilutę **labas, aš esu korta**.
- išveskite į konsolę objekto **korta** savybei **rūšis** priskirtą reikšmę;
- išveskite į konsolę objekto **korta** savybei **simbolis** priskirtą reikšmę;
- iškvieskite objekto **korta** metodą **pasiveikinti**.

2 užduotis

Skirtingai nei funkcijos, objektai neturi kintamųjų erdvės, todėl norint objekto metode pasiekti iš to paties objekto savybės, reikia nurodyti to objekto vardą arba vietoje jo naudoti bazinį žodį **this**. Pavyzdžiui, duotas programos kodas:

```
var objektas1 = {
  "savybė1": 1,
  "savybė2": 2,
  "metodas1": function ()
  {
    console.log(objektas1["savybė1"]);
    console.log(this["savybė2"])
  }
}
```

objektas1["metodas1"]();

Papildykite programos kodą:

- sukurkit kintamąjį **korta**;
- priskirkite jam JSON notacija nusakytą objektą, kuris:
 - turi savybę **rūšis**, jai priskirta simbolių eilutė (duomenų tipas **string**) **vynai**;
 - turi savybę **simbolis**, jai priskirta simbolių eilutė (duomenų tipas **string**) **A**;
 - turi metodą **išvesti_į_konsolę**, kuris į konsolę išveda objekto savybėms **rūšis** ir **simbolis** priskirtas reikšmes.
- iškvieskite objekto **korta** metodą **pasiveikinti**.

3 užduotis

Jau sukurtus objektus galime papildyti naujomis savybėmis. Kad papildyti objektą norimo vardo savybę, pakanka priskirti to vardo savybei reikšmę. Pavyzdžiui, duotas programos kodas:

```
var korta = {  
  rūšis: "bugnai"  
}  
  
korta["simbolis"] = "J";
```

Papildykite programos kodą:

- papildykite objektą `korta` metodu `išvesti_į_konsolę`, kuris į konsolę išveda objekto savybėms `rūšis` ir `simbolis` priskirtas reikšmės;
- iškvieskite objekto `korta` metodą `išvesti_į_konsolę`.

4 užduotis

JavaScript kalboje objektas gali iš kito objekto paveldėti visas jo savybes ir metodus.

Pavyzdžiui, duotas programos kodas:

```
var objektas1 = {  
  "savybė1": 1,  
  "savybė2": 2,  
  "metodas1": function ()  
  {  
    console.log(objektas1["savybė1"]);  
    console.log(this["savybė2"]);  
  }  
}  
var objektas2 = {  
  "savybė3": 3  
}  
  
Object.setPrototypeOf(objektas2, objektas1);  
  
objektas2.metodas1();
```

pavyzdyje, metodo `Object.setPrototypeOf` pagalba objektas `objektas2` paveldi visas `objektas1` savybes.

Papildykite programos kodą:

- sukurkit kintamąjį `objektas3`;
- priskirkite jam JASON notacija nusakytą objektą, kuris turi savybę `savybė4`, kuriai priskirta reikšmė 4 (duomenų tipas `number`);
- `objektas3` turi iš `objektas1` paveldėti visas jo savybes ir metodus.
- iškvieskit `objektas3` metodą `metodas1`.

5 užduotis

JavaScript kalboje paveldėjimas vyksta per prototipus. Visi objektai turi savybę `__proto__` per kuria paveldi kito objekto savybes ir metodus. Per savybę `__proto__` paveldimos visos savybės ir metodai kuriuos turi jai priskirtas objektas. Objektai turi tik po vieną prototipą (savybę `__proto__`), todėl ir tiesiogiai paveldėti gali tik iš vieno objekto. Paveldėjimas per savybę `__proto__` yra grandininis, todėl paveldima ne tik savybei `__proto__` priskirto objekto savybės, bet ir savybės kurias tas objektas paveldi iš savo savybei `__proto__` priskirto objekto ir t.t. Objekto prototipui (savybę `__proto__`) priskirti objektą galime metodu `Object.setPrototypeOf` arba priskyrimo operatoriumi `=`. Pavyzdžiui, duotas programos kodas:

```
var objektas1 = {
  "savybė1": 1,
  "savybė2": 2,
  "metodas1": function ()
  {
    console.log(objektas1["savybė1"]);
    console.log(this["savybė2"]);
  }
}
var objektas2 = {
  "savybė3": 3
}

var objektas3 = {
  "savybė4": 4
}

objektas2.__proto__ = objektas1;
console.log(objektas2.__proto__)

objektas3.__proto__ = objektas2;
console.log(objektas3.__proto__)

objektas3.metas1();
```

Papildykite programos kodą:

- sukurkit kintamąjį `objektas4`;
- priskirkite jam JSON notacija nusakytą objektą, kuris turi savybę `savybė5`, kuriai priskirta reikšmė 5 (duomenų tipas `number`);
- `objektas4` turi iš `objektas3` paveldėti visas jo savybes ir metodus.

- Išveskit į konsolę `objektas4` savybėms `savybė1` `savybė2` `savybė3` `savybė4` `savybė5` priskirtas reikšmes.

6 užduotis

Kuriant objektus turinčius identiškus savybių vardus, derėtų susikurti tokį objektą konstruojančią funkciją. Pavyzdžiui, duotas programos kodas:

```
var Korta = function (in_rūšis, in_simbolis)
{
    this["rūšis"] = in_rūšis;
    this["simbolis"] = in_simbolis;
}

var korta1 = new Korta("širdys", "Q");
var korta2 = new Korta("širdys", "K");
```

čia, kintamajam `Korta` (objektą konstruojančias funkcijas priimta vadinti iš didžiosios raidės) priskirta nauja funkcija, kuri turi argumentus `in_rūšis` ir `in_simbolis`. Išsaukta naujo objekto viduje ji:

- sukurtu savybę `rūšis` ir jai priskirtu jos argumentui `in_rūšis` priskirtą reikšmę;
- sukurtu savybę `simbolis` ir jai priskirtu jos argumentui `in_simbolis` priskirtą reikšmę.

Objektą konstruojančios funkcijos naudoja bazinį žodį `this`, todėl jas kviesti derėtų tik objekto viduje. Naujo objekto kūrimui su objektą konstruojančia funkcija naudojamas operatorius `new`. Operatorius `new` sukuria naują objektą ir jo viduje iššaukia operande nurodytą funkciją, bei gražina nuorodą į sukurtą objektą, kad sukurtą objektą galėtume priskirti kintamajam. Kadangi, kintamiesiems `korta1` ir `korta2` priskirti objektai buvo sukurti pasitelktus objektą konstruojančią funkciją `Korta`, šie objektai yra `Korta` tipo.

Papildykite programos kodą:

- sukurkite kintamąjį vardą `Automobilis`;
- priskirkite kintamajam `Automobilis` naują funkciją, kuri:

- turi argumentus `arg_gamintojas` ir `arg_modelis` ir `arg_pagaminimo_metai`;
- išsaukta objekto viduje, sukuria savybę `gamintojas` ir priskiria jos argumentui `arg_gamintojas` priskirtą reikšmę;
- išsaukta objekto viduje, sukuria savybę `gamintojas` ir priskiria jos argumentui `arg_gamintojas` priskirtą reikšmę;
- išsaukta objekto viduje, sukuria savybę `modelis` ir priskiria jos argumentui `arg_modelis` priskirtą reikšmę;
- išsaukta objekto viduje, sukuria savybę `pagaminimo_metai` ir priskiria jos argumentui `arg_pagaminimo_metai` priskirtą reikšmę;
- sukurkite kintamąjį vardu `automobiliai`;
- priskirkite kintamajam `automobiliai` naują tuščią masyvą;
- naudodami operatorių `new` ir objektą konstruojančią funkciją `Automobilis` sukurkite ir į masyvą `automobiliai` patalpinkite tris `Automobilis` tipo objektus.

7 užduotis

Su operatoriumi `new` sukurti objektai, automatiškai paveldi konstruojančios funkcijos savybei `prototype` priskirto objekto savybes ir metodus. Pavyzdžiui, duotas programos kodas:

```
var Korta = function (in_rūšis, in_simbolis)
{
    this["rūšis"] = in_rūšis;
    this["simbolis"] = in_simbolis;
}

Korta.prototype.išvesti_i_konsolę = function ()
{
    console.log("Korta:", this.rūšis, this.simbolis);
}

var korta1 = new Korta("širdys", "Q");
var korta2 = new Korta("širdys", "K");

korta1.išvesti_i_konsolę();
```

Papildykite programos kodą: