

Varianta B – JSON Objects

1 Problema

Se consideră o intrare în format JSON¹. Se cere să se afișeze informații despre valorile de tip obiect (*object*).

Pentru fiecare valoare de tip obiect se vor afișa:

- *calea valorii* – cheile (pentru obiecte) sau indecșii (pentru liste) valorilor în care este conținut obiectul.
 - calea va fi afișată folosind săgeată (->) pentru cheile strămoș și indexul (e.g. [2]) pentru strămoșii care sunt elemente de listă (0-based).
 - dacă rădăcina fișierului este un obiect, aceasta va fi denumită „rădăcină”.
- tipurile de valori pe care le conține.
 - valorile pot fi: număr, șir de caractere (pe scurt, șir), boolean, listă, obiect, null.
 - în lista de tipuri conținute, fiecare tip va apărea o singură dată.
 - se vor afișa numai valorile care corespund unor chei ale obiectului, nu valori imbricate în alte liste / obiecte din interior.
 - ordinea în care sunt specificate tipurile valorilor conținute nu este importantă.
 - afișarea pentru fiecare obiect se va realiza de la începutul liniei.
- afișarea informațiilor pentru fiecare obiect se va face *după* ce explorarea obiectului s-a terminat (după ce s-au afișat informațiile despre eventuale obiecte pe care acesta le conține).

Note:

- recunoașterea tipului unei valori se va face numai pe baza primului caracter din valoare.
- valorile de tip număr nu vor conține exponent, dar pot fi numere fracționare sau negative.
- valorile de tip șir de caractere nu vor conține escape characters (nu vor conține caracterul \).
- orice obiect va fi imbricat în (va avea o adâncime de) maxim 10 obiecte / liste.
- cheile vor avea o lungime de maxim 12 caractere.
- formatul pentru ieșire prezentat în exemple nu trebuie respectat întocmai, dar este bine să aveți un format similar, care să respecte regulile prezentate mai sus.

¹JSON <https://www.json.org>

2 Exemple

2.1 Exemplul 1

Intrare

```
{
  "colors": [
    {
      "color": "black",
      "category": "hue",
      "type": "primary",
      "code": {
        "rgba": [255,255,255,1],
        "hex": "#000"
      }
    },
    {
      "color": "white",
      "category": "value",
      "code": {
        "rgba": [0,0,0,1],
        "hex": "#FFF"
      }
    },
    {
      "color": "blue",
      "category": "hue",
      "type": "primary",
      "code": {
        "rgba": [0,0,255,1],
        "hex": "#00F"
      }
    },
    {
      "color": "green",
      "category": "hue",
      "type": "secondary",
      "code": {
        "rgba": [0,255,0,1],
        "hex": "#0F0"
      }
    }
  ]
}
```

Ieșire

```
colors [0] -> code conține liste, șiruri
colors [0] conține șiruri, obiecte
colors [1] -> code conține liste, șiruri
colors [1] conține șiruri, obiecte
colors [2] -> code conține liste, șiruri
colors [2] conține șiruri, obiecte
colors [3] -> code conține liste, șiruri
colors [3] conține șiruri, obiecte
rădăcina conține liste
```

2.2 Exemplul 2

Intrare

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    { "type": "home", "number": "212 555-1234" },
    { "type": "office", "number": "646 555-4567",
      "sections": [1, 2, 54]
    },
    { "type": "mobile", "number": "123 456-7890"
  }
],
  "children": [],
  "spouse": null
}
```

Ieșire

```
address conține șiruri
phoneNumbers [0] conține șiruri
phoneNumbers [1] conține șiruri, liste
phoneNumbers [2] conține șiruri
rădăcina conține șiruri, booleeni, numere, obiecte, liste, null
```

2.3 Exemplul 3

Intrare

```
[
  {
    "_id": "5973782bdb9a930533b05cb2",
    "isActive": true,
    "balance": 1446.35,
    "age": 32,
    "name": "Logan Keller",
    "friends": [
      {
        "id": 0,
        "name": "Colon Salazar"
      },
      {
        "id": 1,
        "name": "French McNeil"
      },
      {
        "id": 2,
        "name": "Carol Martin",
        "relatives": [ "Jane Martin", "Josh Martin-Stocks" ]
      }
    ],
    "favoriteFruit": "banana"
  },
  {
    "_id": "42",
    "isActive": [true, false],
    "name": "John Answer",
    "friends": ["Falsity", "Tautology"]
  }
]
```

Ieșire

```
[0] -> friends [0] conține numere, șiruri
[0] -> friends [1] conține numere, șiruri
[0] -> friends [2] conține numere, șiruri, liste
[0] conține șiruri, booleeni, numere, liste
[1] conține șiruri, liste
```

2.4 Exemplul 4

Intrare

```
{
  "$schema": "http://json-schema.org/draft-06/schema#",
  "title": "Product",
  "description": "A product from Acme's catalog",
  "type": "object",
  "properties": {
    "id": {
      "description": "The unique identifier for a product",
      "type": ["integer", "string", "boolean"]
    },
    "name": {
      "description": "Name of the product",
      "type": "string",
      "accepted_values": [[1,2],[4,5]]
    },
    "price": {
      "type": "number",
      "exclusiveMinimum": 0,
      "required": [true,false],
      "value": {
        "price": 100,
        "make": ["bmw", "audi"],
        "city": ["munich", "ingolstadt"],
        "owners": [
          { "name": ["Kaiser","Muller"] },
          { "name": ["Thomas","Schwein"] },
          { "name": ["Jennet"] },
          { "name": [null, null, null] },
          { "name": [null] }
        ]
      }
    },
    "tags": {
      "type": "array",
      "items": {
        "type": "string",
        "object": {
          "nothing": true
        }
      },
      "minItems": 1,
      "uniqueItems": true
    }
  },
  "required": ["id", "name", "price"]
}
```

Ieșire

```
properties -> id conține șiruri, liste
properties -> name conține șiruri, liste
properties -> price -> value -> owners [0] conține liste
properties -> price -> value -> owners [1] conține liste
properties -> price -> value -> owners [2] conține liste
properties -> price -> value -> owners [3] conține liste
properties -> price -> value -> owners [4] conține liste
properties -> price -> value conține numere, liste
properties -> price conține șiruri, numere, liste, obiecte
properties -> tags -> items -> object conține booleani
properties -> tags -> items conține șiruri, obiecte
properties -> tags conține șiruri, obiecte, numere, booleani
properties conține obiecte
rădăcina conține șiruri, obiecte, liste
```