Code Test

Instructions:

Test time: ~ 4 hours

Utilities: Your favorite IDE

The test: Consists of five tasks (1+1+3)

The result: Submit runnable code (Java or C#) including unit tests

Good luck!

Interview questions

1. Numbers

Write a function (*getValue(n)*) that returns the n:th value in the following series:

```
1,1,2,3,5,8,13,21,34
```

Thus the n:th value is the sum of the two values before. Examples: getValue(6) -> 8 and getValue(8) -> 21

2. Flatten

Assume the following javascript function exists:

```
function() {
       var movieLists = [
           {
                    name: "New Releases",
                    videos: [
                           {
                              "id": 70111470,
                              "title": "Die Hard",
                              "boxart": "http://cdn-
0.nflximg.com/images/2891/DieHard.jpg",
                              "uri":
"http://api.netflix.com/catalog/titles/movies/70111470",
                              "rating": 4.0,
                              "bookmark": []
                           },
                           {
                              "id": 654356453,
                              "title": "Bad Boys",
                              "boxart": "http://cdn-
0.nflximg.com/images/2891/BadBoys.jpg",
                              "uri":
"http://api.netflix.com/catalog/titles/movies/70111470",
                              "rating": 5.0,
                              "bookmark": [{ id:432534, time:65876586
}]
                            }
                           ]
          },
          {
                     name: "Dramas",
                     videos: [
                            {
                              "id": 65432445,
                              "title": "The Chamber",
                              "boxart": "http://cdn-
0.nflximg.com/images/2891/TheChamber.jpg",
                              "uri":
"http://api.netflix.com/catalog/titles/movies/70111470",
```

```
                                "rating": 4.0,
                                "bookmark": []
                            },
                            {
          "id": 675465,
          "title": "Fracture",
          "boxart": "http://cdn-
0.nflximg.com/images/2891/Fracture.jpg",
          "uri":
"http://api.netflix.com/catalog/titles/movies/70111470",
          "rating": 5.0,
          "bookmark": [{ id:432534, time:65876586 }]
          }
      ]
    }
  ],
  allVideoIdsInMovieLists = [];

  // ------------   INSERT CODE HERE!  ----------------------------
  // Use two nested forEach loops to flatten the movieLists into a
list of video ids.
  // ------------   INSERT CODE HERE!  ----------------------------
  return allVideoIdsInMovieLists;
}
```

Insert the missing code in the function above to flatten the list into a list of ids.

3. The Web Shop

3.1 Data structures

You are building a web shop. In this web shop *customers* can add *products* to a *shopping cart*. A customer can add many products to the shopping cart. A product can have many different types of *attributes*, and they can vary between products, e.g. a Bike can have the following attributes: {price, color, size} and the product Table: {price, width, depth, height}. *Product Attributes* are organized in a hierarchical group structure, where a *Product Attribute Group* can contain one or several Product Attributes and/or Product Attribute Groups:

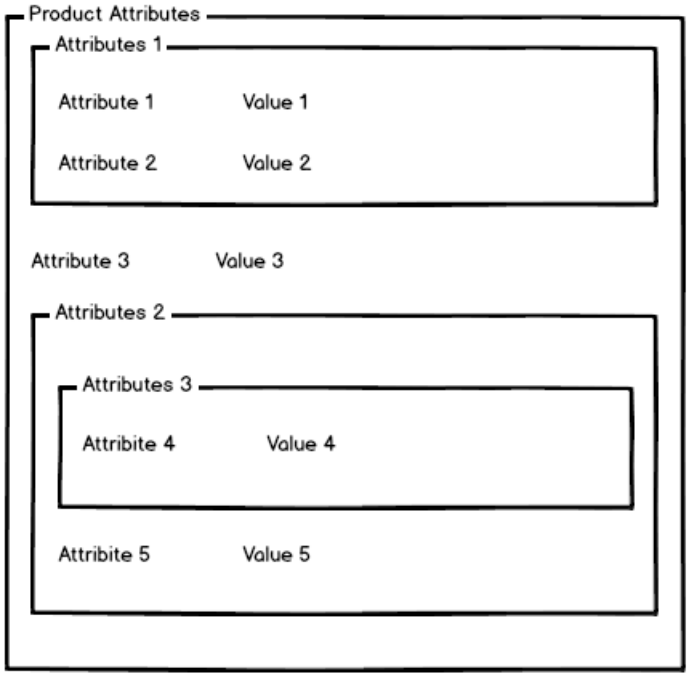
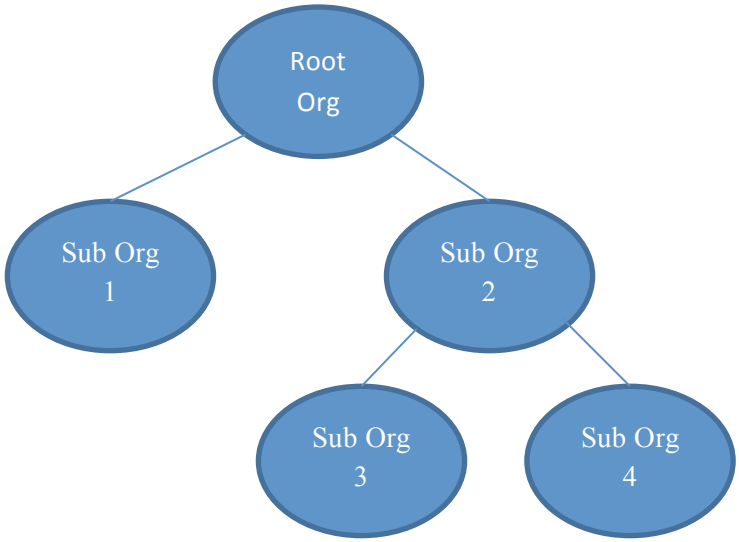


Figure 1: Note that the order is important, in the figure above Attributes 1, Attribute 3 and Attributes 2 all belong to the Product Attributes group.

The *customer* shall have a *name* and an *address* and be connected to an *organization*. Organizations are structured as a tree. It shall also be possible for the customer to create different shopping carts.

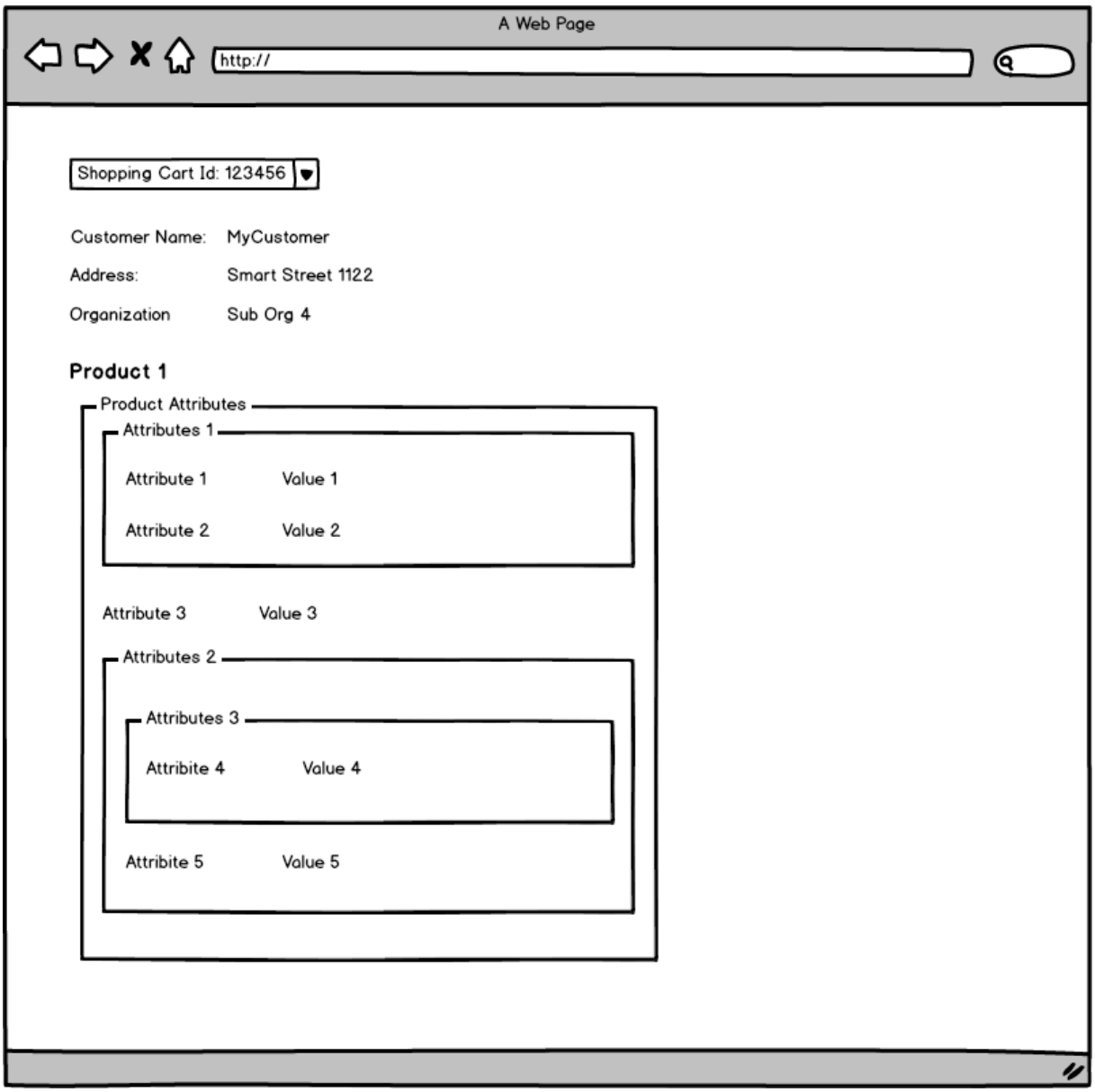
Example of organization tree:



Write the code that implements the Web Shop described above (Objects, Object Attributes, methods). The solution shall be as generic as possible, thus support any type of product and attributes (no hard coded attributes).

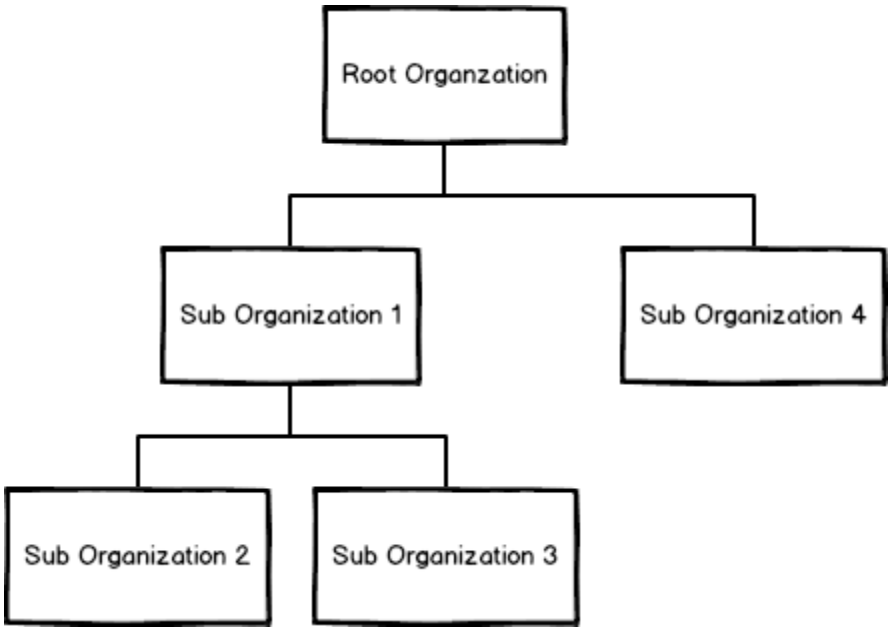
2.1.1. Print Shopping Cart

Write a function – print () – that displays the complete shopping cart in a browser (HTML), including the product attributes. First shall the customer data be printed for the selected Shopping Cart, then a table of the products and their attributes. See picture below.



2.1.1. Print Organizations

Write a function hat prints the organization tree to the browser (HTML), example:



Or alternatively:

Level 1	Level 2	Level 3
Root Organization		
	Sub Organization 1	
		Sub Organization 2
		Sub Organization 3
	Sub Organization 4	