**1 Implementation Task**

A persistent storage is given as a map of key value pairs. The map has the following interface:

* void put(String key, Object value): store a value under a given key
* Object get(String key): retrieve the value for a given key
* boolean contains(String key): returns true if the key is already used to store a value in the persistent storage.
* boolean remove(String key): returns true if the remove was successful

1. Implement the persistent storage – you may use standard language libraries (e.g. a map implementation in Java). The storage must be able to survive application restarts;
2. Implement a) without the standard language libraries (e.g. in Java do not use any of the standard provided map implementations);
3. Describe (e.g. using text and diagrams) how could a distributed (existing with the same state in multiple processes at the same time) version of the persistent storage be implemented;  No implementaion is required;

**2 Object-Oriented Design**

Design and implement an object model of the object in the following picture. Please describe the context, in which the object model should be applied.



**Answers**:

**1 Implementation Task:**

**a) and b) –** Implemented.

**c)**  **Describe (e.g. using text and diagrams) how could a distributed (existing with the same state in multiple processes at the same time) version of the persistent storage be implemented;  No implementaion is required;**

I suppose it would be like in RAID(**Redundant Array of Independent Disks).** RAID works as a form of storage virtualization that combines multiple physical disks into one logical volume. There are many RAID levels, but this one works best:  
RAID 5: This setup requires at least three drives, and uses block-level striping (as in RAID 0) and distributed parity. This means that the data is written in such a way so if one drive is damaged or fails, you can still recover all your data. But you don't get all the capacity of the drives, because of this protection. For example, if you have three 4 TB drives in a unit, you'll get 8 TB capacity with RAID 5. With five such drives, you'll get 16 TB. Most RAID units allow you to hot-swap a drive - change it without restarting the unit - and it automatically "rebuilds" when you do so, ensuring that the new drive takes its place in the array.

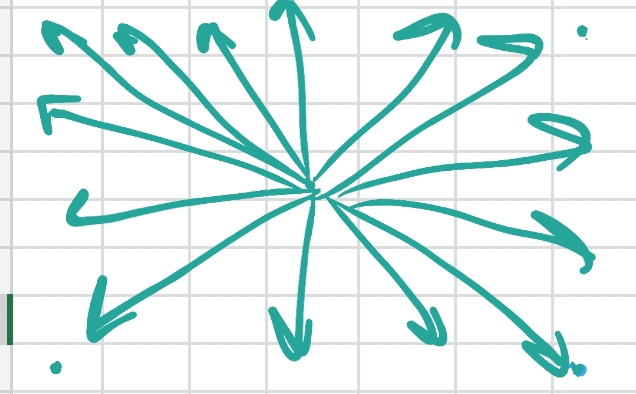
**2 Object-Oriented Design:**

**Design and implement an object model of the object in the following picture. Please describe the context, in which the object model should be applied.**

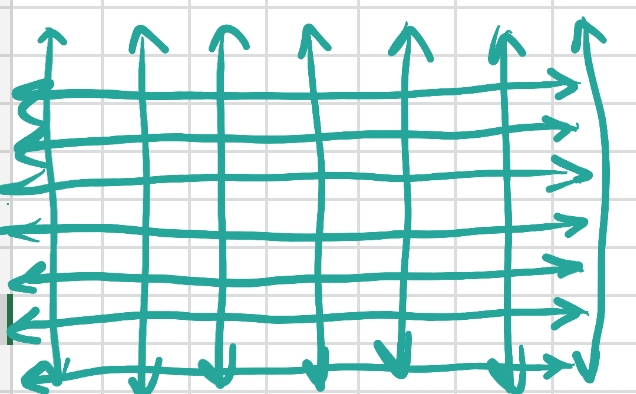
The object is a stapler. It has the properties e.g. fiels – COLOR, STAPLES\_COUNT, its actions e.g. methods would be STAP(), REFILL(). It may have way more attributes and appliances, but we will make an abstract object out of it, since OOP does not need to implement all features but only the essential ones for clients’ purposes.

**3 Open Question:**

**Please answer the question: “How many buses (public transport) are there in Sofia?” by approximating on the basis of logical conclusions. Feel free to provide also any assisting drawings or other representations supporting your conclusions.**

There are 56 districts in Sofia according to Wikipedia. If we represent them in an array, it would have dimension 7x8 squares like in the graphic below.We assume that the center of this array is the starting point – from there to each of the end districts a bus would be needed – in total 28. Each bus route is about 3.5 – 4 districts long.

We should connect the districts horizontally and vertically as well (like on the table below) which means additional 15 bus lines. In this case each bus route would be 7 or 8 districts long, which is too far away for a bus line (we assume that machines needs refuel and bus drivers would need break). So we should stick to the previous case bus route distance and cut the bus distance in half.That would double the number of needed bus lines for this graphic and make them 30.

With the two theories combined, we would need **43 bus lines** in total covering public transportation in Sofia.