# 03. Clothing magazine



*You are an appraiser and you have to audit clothing magazines. Let's get started!*

**Preparation**

Download the skeleton provided in Judge. **Do not** change the **packages**!

**Pay attention to name the package magazine, all the classes, their fields and methods the same way they are presented in the following document. It is also important to keep the project structure as described.**

**Problem Description**

Your task is to create a repository that stores clothes by creating the classes described below.

### Cloth

First, write a class **Cloth** with the following properties:

* **color: String**
* **size: int**
* **type: String**

The class **constructor** should receive **color, size and type**. You need to create the appropriate **getters and setters**. Override the **toString()** method in the following format:  
**"Product: {type} with size {size}, color {color}"**

### Magazine

**Next**, write a class **Magazine** that has **data** (a List that stores the entity **Cloth**). All entities inside the repository have the **same properties**. Also, the **Magazine** class should have those **properties**:

* **type: String**
* **capacity: int**

The class **constructor** should receive **type** and **capacity**, also it should initialize the **data** with a new instance of the collection.Implement the following features:

* **List<Cloth> data** – **collection** that holds added clothes
* **Method addCloth(Cloth cloth)** – **adds** an **entity** to the data **if** **there** **is** **room** for it
* **Method removeCloth(String color)** – removes a cloth by **given color,** if such **exists**, and **returns boolean** (**true** if it is removed, otherwise – **false**)
* **Method getSmallestCloth()** – **returns the smallest cloth**
* **Method getCloth(String color)** – **returns** the **cloth** with the **given color**
* **Method getCount** – **returns** the **number** of **clothes**
* **Method report()** – **returns** a **string** in the following **format** (print the clothes in **order of appearance**):
  + **"{type} magazine contains:  
    {Cloth1}  
    {Cloth2}  
    (…)"**

**Constraints**

* The **color** and **size** of the clothes will be **always unique**.
* You will always have clothes added before receiving methods manipulating the Magazines’ clothes.

**Examples**

This is an example of how the **Magazine** class is **intended to be used**.

|  |
| --- |
| **Sample code usage** |
| //Initialize the repository (Magazine)Magazine magazine = **new** Magazine(**"**Zara**"**, 20);  //Initialize entity (Cloth)Cloth cloth1 = **new** Cloth(**"**red**"**, 36, **"**dress**"**);  //Print ClothSystem.***out***.println(cloth1); //Product: dress with size 36, color red//Add Clothmagazine.addCloth(cloth1);  //Remove ClothSystem.***out***.println(magazine.removeCloth(**"**black**"**)); //falseCloth cloth2 = **new** Cloth(**"**brown**"**, 34, **"**t-shirt**"**);  Cloth cloth3 = **new** Cloth("blue", 32, "jeans");  //Add Clothmagazine.addCloth(cloth2);  magazine.addCloth(cloth3);  //Get smallest clothCloth smallestCloth = magazine.getSmallestCloth(); //Product: jeans with size 32, color blue  System.***out***.println(smallestCloth);  //Get clothCloth getCloth = magazine.getCloth(**"**brown**"**); //Product: t-shirt with size 34, color brownSystem.***out***.println(getCloth);   System.***out***.println(magazine.report()); //Zara magazine contains: //Product: dress with size 36, color red //Product: t-shirt with size 34, color brown  //Product: jeans with size 32, color blue |

**Submission**

Submit **single .zip file**, containing **magazine** package, **with the classes inside** (**Magazine** and **Cloth** and the **Main** **class)**, there is no specific content required inside the **Main** class e. g. you can do any kind of local testing of you program there. However, there should be **main(String[] args)** method inside.