# Christmas

A picture containing LEGO, vector graphics

Description automatically generated

*Can you imagine Santa without his special bag that carries presents for so many children? Neither can we. So let's have a quick sneak peek inside it and help Santa rearrange it.*

## Preparation

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

**Pay attention to the name of the package (christmas), 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 above.**

## Problem description

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

### Present

First, write a Java class Present with the following fields:

* **name: String**
* **weight: double**
* **gender: String**

The class **constructor** should receive (**name, weight, and gender**).

The class also should have the methods:

* getName()
* getWeight()
* getGender()
* Override the **toString()** method in the following format:

**"Present {name} ({weight}) for a {gender}"**

**Note: Format the weight to the second digit after the decimal point!**

### Bag

**Next**, write a **Java** class Bag that has **data** (a collection that stores the entity **Present**). All entities inside the repository have the **same fields**. Also, the Bag class should have those fields:

* **color:** String
* **capacity:** int
* **data:** List<Present> **-** holds all added presents in the bag

The class **constructor** should receive (**color**, **capacity**), also it should initialize the **data** with a new instance of the collection**.**

Implement the following features:

* getColor()
* getCapacity()
* count() method– **returns** the **number** of presents
* add(Present present) method – **adds** an **entity** to the data **if** **there** **is** **room** for it
* remove(String name) method – removes a present by **given name,** if such **exists**,   
  and **returns boolean**
* heaviestPresent() method – returns the **heaviest** present
* getPresent(String name) method – returns the present with the **given name**
* report() method – **returns** a **string** in the following **format** (print the presents in order of appearance):
  + **"{color of Bag} bag contains:  
    {Present1}  
    {Present2}  
    (…)**"

## Constraints

* The **names** of the presents will be **always unique**.
* The **weights** of the presents will always be with **positive values**.
* You will always have a present added before receiving methods.

## Examples

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

|  |
| --- |
| Sample code usage |
| *// Initialize the repository* Bag bag = **new** Bag(**"black"**, 10);  *// Initialize entity* Present present = **new** Present(**"Doll"**, 0.4, **"girl"**);  *// Print Present* System.***out***.println(present); *// Present Doll (0.40) for a girl*  *// Add Present* bag.add(present);  *// Remove Present* bag.remove(**"Toy"**); *// false*  Present secondPresent = **new** Present(**"Train"**, 2, **"boy"**); *// Add Present* bag.add(secondPresent);  Present heaviestPresent = bag.heaviestPresent();  System.***out***.println(heaviestPresent);  *// Present Train (2.00) for a boy*  Present p = bag.getPresent(**"Doll"**);  System.***out***.println(p);  *// Present Doll (0.40) for a girl*  System.***out***.println(bag.count()); *// 2* System.***out***.println(bag.report());  *// Black bag contains: // Present Doll (0.40) for a girl // Present Train (2.00) for a boy* |

## Submission

Submit a **single .zip file**, containing a **christmas package, with the classes inside (Present, Bag, 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 your program there. However, there should be a **main(String[] args)** method inside.