

# Sac de boules

Prénom \_\_\_\_\_  
NOM \_\_\_\_\_

20

## Interrogation de Java

(Mercredi 15 mars 2006)

### Remarques :

- ↪ vous travaillerez dans un répertoire `evaluations/i3`
- ↪ vous pouvez utiliser **toutes les notes** que vous voulez ainsi que les programmes qui se trouvent dans votre répertoire,
- ↪ l'interro se fait sur linux1,
- ↪ vous disposez de 2 heures,
- ↪ vous travaillez sur machine et remettrez votre programme au terme du temps imparti.
- ↪ soyez attentif au **style**, à la **clarté** et à l' **indentation** du code

**E**NONCÉ. Les mathématiciens aiment faire des statistiques. Pour ce faire, ils aiment manipuler un ensemble de **boules** dans un sac .. généralement des boules rouges et des boules blanches.

## 1 Classe Boule

(6 pts)

Vous écrirez une classe `Boule` représentant une boule.

- ↪ une boule possède les **attributs**
  - ˘ un `String` représentant sa couleur et
  - ˘ un `int` représentant sa valeur
- ↪ ... et les méthodes
  - ˘ un constructeur (avec deux paramètres),
  - ˘ des *getters* (mais pas de *setters*),
  - ˘ des méthodes `equals` et `toString` habituelles,
  - ˘ une méthode **isochrome**<sup>1</sup> permettant de déterminer si deux boules ont la même couleur

## Javadoc

(2 pts)

Pour cette classe, vous écrirez la **javadoc**.

<sup>1</sup>Je sais, c'est sans doute un néologisme.

## 2 Classe Sac

(9 pts)

Vous écrirez une classe `Sac` représentant un sac de boules. Un sac contient un ensemble de boules rouge ou blanche, numérotée à partir de 0.

Un sac de boules possède un attribut de type *Collection* (à vous de choisir celle que vous préférez ... mais pas un tableau),

... et les méthodes

↪ deux constructeurs,

˘ un constructeur construisant un sac vide,

˘ un constructeur construisant un sac de  $n$  boules rouges numérotées de 0 à  $n - 1$  et de  $m$  boules blanches numérotées de 0 à  $m - 1$

↪ deux méthodes permettant de tirer des boules dans le sac,

˘ une méthode `tirerSansRemise` qui retourne une boule du sac et la supprime du sac et

˘ une méthode `tirerAvecRemise` qui retourne une boule du sac sans la supprimer

Pour écrire ces deux dernière méthodes, vous aurez besoin de mélanger votre sac. Si ce sac est une `List`, il vous suffit d'écrire

```
Collections.shuffle(sac) ;
```

↪ une méthode `main` permettant de visualiser le tirage de 5 boules avec remise dans un sac de 20 boules (10 blanches et 10 rouges) et le tirage de 5 boules **sans** remise cette fois dans un sac de 20 boules (10 blanches et 10 rouges).

## 3 Test unitaires

(3 pts)

Ecrivez des tests pour la classe `Boule`. Vous testerez les méthodes

↪ `isochrome` et

↪ `equals`