3. Feuille de TP

Semaine 14 :

Contenant et gestion de contenants

Dans cet exercice, on va définir une interface `Contenant<T>` représentant la notion de quelque chose qui peut contenir des objets de type `T`.

Ainsi, un `Contenant<T>` est un objet qui a une méthode booléenne `contient` prenant en entrée un objet de type `T`.

1. Écrivez l'interface `Contenant<T>`.
2. Écrivez une classe `Couple` représentant un couple d'entiers et implémentant l'interface `Contenant<Integer>`. Le Couple *contient* l'entier x si l'une ou l'autre des deux valeurs du couple est égale à x.
3. Écrivez une classe `GestionContenants` ayant une méthode statique `contiennentTous` prenant en paramètre une List<Contenant<T>> conts et un T elem et renvoyant vrai si elem est contenu dans tous les Contenant de conts.
4. Écrivez une classe `Ensemble` représentant un ensemble d'entiers et implémentant l'interface `Contenant<Integer>`.

5. Dans un exécutable, créez une `List<Contenant<Integer>>` dans lequel vous ajouterez le couple (0, 1), l'ensemble {0, 1, 2, 3, 4} et le couple (0, 2). Vérifiez que la méthode `contiennentTous` renvoie vrai pour l'entier 0 et faux pour l'entier 1. Faites attention à l'appel de la méthode statique.

Des Cadeaux!!

1. Définir une interface **Cadeau** qui comporte une méthode **getPoids**().
2. Définir trois classes **Diamant**, **Argent** et **BouquetDeFleurs** implémentant cette interface pour que le programme principal suivant soit correct. Le poids d'une Rose est de 3g celui d'un Dahlia de 5g. Chaque pièce pèse 5g.

```
class Exec  {
   public static void main(String [] args) {
       List<Cadeau> sac = Arrays.asList(new Diamant(7),
                                        new BouquetDeFleurs(1, 3),
                                        new Argent(5));
       System.out.println(sac);
       // [Diamant de 7g, Bouquet de 1 Dalhia(s) et de 3 Rose(s), 5 pièces de 5 g␣
↪chacune]
  }
}
```

Ronins Yakuza et Maitre des clefs

Ces trois personnages participent à un jeu et ne peuvent faire que les actions suivantes :

1. entendre un message,
2. recevoir un cadeau,
3. interagir entre eux.

- Le maître des clefs connait un *mot de passe* qu'il peut transmettre à un personnage par l'intermédiaire de son action *interagir*. Il n'entend rien et ne reçoit rien. Quand il affiche son mot de passe, il le code!
- Le Ronin possède un *mot de passe* a priori inconnu. Seul le Maître des clefs peut lui en fournir un, s'il l'*entend* ; dans ce cas, il le retient. Il possède en outre un *sac de cadeaux* qui ne peut peser que *30g*. Lorsqu'il *reçoit* un cadeau, il ne peut donc pas dépasser ce poids (et peut donc perdre son cadeau). Enfin, lorsqu'il *interagit* avec un autre personnage, si son sac pèse plus de *20g*, il perd son *dernier* cadeau alors que l'autre personnage le *reçoit*.
- Enfin le Yakuza, quant à lui, possède un *sac de cadeaux* qui ne peut pas peser plus de *50g*. Lorsqu'il *entend* un message, il prend peur et *perd* tous les cadeaux de son sac. Lorsqu'il *interagit* avec un personnage, celui-ci reçoit le premier cadeau de son sac, qu'il perd donc.

On vous donne le petit scénario suivant :

```
public class Executable {
   public static void main(String [] args) {
       List<Cadeau> sac =
           Arrays.asList(new Diamant(7),
                         new BouquetDeFleurs(1, 3),
                         new Argent(5)
                         );
       System.out.println(sac);
       // [Diamant de 7g, Bouquet de 1 Dalhia(s) et de 3 Rose(s),
       //  5 pièce(s) de 5 g chacune]

       MaitreDesClefs maitre = new MaitreDesClefs("saloir");
       System.out.println(maitre);
       // Moi Maître des clefs je suis sourd
       // je refuse les cadeaux!
       // voici mon mot de passe codé
```

(suite sur la page suivante)

(suite de la page précédente)

```
        // 18638874811863887463186388747418638874771863887471186388748O

        Ronin ronin = new Ronin();
        ronin.recevoir(new Diamant(4));
        ronin.recevoir(new BouquetDeFleurs(2, 1));
        ronin.recevoir(new Argent(3));
        System.out.println(ronin);
        // Je suis un Ronin mais je n'ai pas encore de mot de passe!
        // Et mon sac : [Diamant de 4g,
        //               Bouquet de 2 dalhia(s) et de 1 rose(s)]


        maitre.interagir(ronin);
        System.out.println("Après interaction avec le Maître des clefs \n"
                           + ronin);
        // Après interaction avec le Maître des clefs
        // Je suis un Ronin mon mot de passe est saloir
        // Et mon sac : [Diamant de 4g,
        //               Bouquet de 2 dalhia(s) et de 1 rose(s)]

        Yakuza yakuza = new Yakuza();
        yakuza.recevoir(new Argent(2));
        yakuza.recevoir(new Diamant(3));
        yakuza.recevoir(new BouquetDeFleurs(5, 3));
        System.out.println(yakuza);
        // Je suis un Yakuza
        // Et mon sac : [2 pièce(s) de 5g chacune,
        //               Diamant de 3g,
        //               Bouquet de 5 dalhia(s) et de 3 rose(s)]

        yakuza.interagir(ronin);
        System.out.println("Après interaction avec un yakuza \n" + ronin);
        // Après interaction avec un yakuza
        // Je suis un Ronin mon mot de passe est saloir
        // Et mon sac : [Diamant de 4g,
        //               Bouquet de 2 dalhia(s) et de 1 rose(s),
        //               2 pièce(s) de 5g chacune]

        System.out.println("Après interaction avec un ronin \n" + yakuza);
        // Après interaction avec un ronin
        // Je suis un Yakuza Et mon sac :
        //      [Diamant de 3g,
        //       Bouquet de 5 dalhia(s) et de 3 rose(s)]

        yakuza.interagir(maitre);
        System.out.println("Après interaction avec un yakuza \n" + maitre);
        // Après interaction avec un yakuza
        // Moi Maître des clefs je suis sourd je refuse les cadeaux!
        // voici mon mot de passe codé
        // 18638874811863887463186388747418638874771863887471186388748O

        System.out.println("Après interaction avec le Maître des clefs \n" + yakuza);
        // Après interaction avec le Maître des clefs
        // Je suis un Yakuza Et mon sac :
        //      [Bouquet de 5 dalhia(s) et de 3 rose(s)]
```

(suite sur la page suivante)

(suite de la page précédente)

```
    }
}
```

Définissez les classes nécessaires.

Amélioration de code

Vous constatez que le Ronin et le Yakuza possèdent tous les deux un sac de cadeaux et qu'il doivent en calculer le poids.

Afin d'améliorer votre code, définissez une classe **abstraite** dont les héritiers seront Ronin et Yakuza et implémentant l'interface. Cette classe sera elle-même héritière d'une ArrayList de cadeaux. C'est elle qui calculera le poids du sac de cadeaux. Cette classe est abstraite dans la mesure où les actions (entendre, recevoir et interagir) ne peuvent pas être définies (pas de code). Il vous suffit d'ajouter le mot clef **abstract** devant la classe ou devant une méthode.

On vous rappelle alors que les héritiers de cette classe abstraite ont accès à toutes les méthodes de ArrayList, si nécessaire.

Assurez vous que le main ci-dessus reste valide!