

Le code Java donné pourra au choix être compatible avec la version 1.4 ou la version 1.5 du Jdk, vous indiquerez simplement en début de copie votre choix et vous y tiendrez dans toutes vos réponses.
 Comme toujours il est conseillé de lire une fois le sujet en entier avant de commencer à le traiter.
 L'évaluation portera sur les choix de conception objet et la correction du code proposé.

LA KERMESSE

C'est le mois de juin et les kermesses fleurissent dans les écoles. Une kermesse regroupe des activités très variées :

- des activités ludiques comme le jeu de massacres, la pêche à la ligne et le jeu d'adresse, pour lesquelles il faut atteindre un objectif pour gagner un lot,
- des ateliers comme le stand maquillage ou le stand de création de marionnettes en carton,
- des stands alimentaires comme la vente de pâtisseries ou la buvette,
- la tombola.

Soit 8 activités en tout (soulignées dans le texte précédent).

Le but d'une kermesse est avant tout de gagner de l'argent afin de financer des activités extra scolaires. Chaque activité a un coût et un tarif de participation. Il y a deux types d'activités : celles accessibles à l'aide d'un ou plusieurs tickets de kermesse pré-payés (dont la valeur en euros est fixée préalablement) et celles pour laquelle on paye directement en euros.

Dans le premier cas, le tarif de l'activité est exprimé en nombre de tickets de kermesse (monnaie K). Dans le second cas, le tarif de l'activité est exprimé en euros.

De plus, pour la plupart des activités, il a été nécessaire d'acheter des fournitures ou des lots ce qui induit un coût lié à chacune des activités.

Enfin, pour participer aux activités il faut parfois satisfaire certaines contraintes, celles-ci peuvent varier d'une activité à l'autre. Il existe cependant une politique par défaut qui est :

- ▷ tout le monde peut participer aux activités dont la monnaie est le ticket de kermesse,
- ▷ il faut que la personne qui participe à l'activité ait au moins 12 ans si le paiement se fait directement en euros.

Q 1. Les politiques de participation à une activité sont gérées par des classes implémentant l'interface `PolitiqueDeParticipation` du paquetage `kermesse` :

« interface »	kermesse::Personne
<i>kermesse::PolitiqueDeParticipation</i>	-age : int
	- nom : String
+peutParticiper(p : Personne) : boolean	getAge() : int
+ toString() : String	getNom() : String

De manière naturelle, la méthode `peutParticiper` renvoie `true` lorsque la personne passée en paramètre satisfait les contraintes de participation à l'activité et `false` sinon ; la méthode `toString` contient une chaîne de caractères décrivant la contrainte sous forme d'un texte court.

La classe `Personne` correspondant aux personnes participant à la kermesse est (au moins) de la forme indiquée ci-dessus.

- Q 1.1.** Expliquez concisément pourquoi le design pattern `SINGLETON` s'applique naturellement aux classes permettant de représenter les 2 politiques de participation par défaut. Vous appliquerez ce design pattern dans le codage des deux questions suivantes.
- Q 1.2.** Définir la classe permettant de représenter la politique de participation par défaut pour les activités dont le paiement se fait en tickets.
- Q 1.3.** Définir la classe permettant de représenter la politique de participation par défaut pour les activités dont le paiement se fait en euros.

Q 2. Donner un diagramme UML de classes¹ présentant votre choix de hiérarchie d'héritage pour la représentation des activités (**Attention** : chaque activité de la kermesse correspond à un objet mais pas à une classe, une réponse où à chaque activité correspondrait une classe serait **fausse**). On applique le design pattern *Strategy* pour gérer les politiques de participation.

Les activités devront au moins (ce n'est sûrement pas limitatif !) disposer des méthodes :

¹ Vous soulignerez les éventuels éléments "abstraits" et préciserez "static" pour les éventuels éléments statiques

▷

```
/** Une personne prend part à l'activité (si elle en a le droit)
 * Dans ce cas, la méthode se contente d'afficher un message
 * tel que "nom a participé à l'activité ...".
 * @param p la personne qui participe à l'activité
 * @exception PasAutoriseException si la personne ne satisfait
 * pas la contrainte de participation à l'activité
 */
public void participe(Personne p) throws PasAutoriseException
```

▷ (attention pour certaines activités la connaissance du prix du ticket est nécessaire, trouvez une solution...)

```
/** Renvoie la recette estimee en euros pour cette activité
 * en fonction d'un nombre donné de participants prévus (on
 * suppose qu'ils participent tous effectivement)
 * @param nbParticipantsPrevus le nombre de participants prevus pour cette activité
 */
public float getRecetteEstimee(int nbParticipantsPrevus)
```

▷ public String toString()

qui renvoie une chaîne de caractères reprenant le nom de l'activité, son tarif et la contrainte de participation. Par exemple, pour le stand "jeu de massacres", la chaîne résultat de l'invocation de cette méthode pourrait être : "Jeu de massacre, tarif : 1 K, ouvert à tous."

Q 3. Donner le code Java pour les classes permettant la gestion des activités et **conforme** au diagramme donné ci-dessus.

Q 4. Donner le code d'une classe `Kermesse` qui gère un ensemble d'activités. Le prix du ticket est fixé à la création de la kermesse et ne peut changer par la suite. On doit pouvoir connaître :

- ▷ les activités offertes par la kermesse,
- ▷ le coût de la kermesse,
- ▷ la recette estimée de la kermesse en fonction d'un nombre moyen de personnes participant à chaque activité (sans tester si elles y ont accès ou pas, il s'agit ici d'avoir une recette "estimative"),
- ▷ le solde (bénéfice ou déficit) estimé qui serait réalisé lors de la kermesse pour un nombre donné moyen de participants par activité.

Q 5. Donner le code Java nécessaire pour une classe `Test`, disposant d'un `main` dans lequel on crée la kermesse décrite ci dessus et ses 8 activités, sachant que

- ▷ la valeur du ticket de kermesse a été fixée à 50 centimes d'euros,
- ▷ l'activité stand de maquillage est réservée aux personnes de moins de 12 ans.
- ▷ Tous les lots distribués pour les activités ludiques ont été achetés ensemble pour un montant total de 120 euros. On répartira ce coût sur chaque activité (le nombre d'activités ludiques est supposé connu avant la création des activités ludiques, le coût pour chacune est facilement calculable). Le tarif de participation à une activité ludique est de 1 ticket de kermesse (1 K).
- ▷ L'atelier de maquillage coûte 150 euros en nécessaire de maquillage et une participation de 2 K est demandée.
- ▷ L'atelier de marionnettes en carton coûte 30 euros et une participation de 1 K est demandée.
- ▷ Les pâtisseries sont faites par les parents d'élèves (coût 0) et sont vendues 2 euros.
- ▷ Les boissons coûtent 120 euros et sont vendues à un prix de 2 euros.
- ▷ La tombola coûte 600 euros et chaque enveloppe coûte 1 euro.

Ce "main" affichera le prix du ticket, l'ensemble des activités offertes, le coût de la kermesse ainsi que les recettes et soldes estimés pour 50 et 200 participants.

Exemple de trace obtenue à l'exécution :

```
Le ticket vaut 0.5 euro
- vente de patisseries, tarif : 2.0 Euro, activite reservee au plus de 12 ans
- buvette, tarif : 2.0 Euro, activite reservee au plus de 12 ans
- tombola, tarif : 1.0 Euro, activite reservee au plus de 12 ans
- marionnettes en carton, tarif : 1.0 K, activite ouverte à tous
- jeu de massacres, tarif : 1.0 K, activite ouverte à tous
- pêche à la ligne, tarif : 1.0 K, activite ouverte à tous
- jeu d'adresse, tarif : 1.0 K, activite ouverte à tous
- stand maquillage, tarif : 2.0 K, activite reservee au moins de 12 ans
Cout de la kermesse 900.0
Recette estimée pour 50 = 400.0 Solde estimé pour 50 = -500.0
Recette estimée pour 200 = 1600.0 Solde estimé pour 200 = 700.0
```