

Objet : modéliser des entités simples dans le cadre objet et utiliser leur modèle informatique.

Conseils : réalisez complètement les classes et testez-les, améliorez-les... au delà des séances de TP (si vous avez un ordinateur, vous pouvez installer java et netbeans qui sont des logiciels gratuits).

Ne vous arrêtez pas dès que vous avez à peu près compris ; il n'est pas suffisant de comprendre pour pouvoir produire soi-même des programmes, il faut se les approprier. Il est aussi plus gratifiant de faire fonctionner des réalisations complètes plutôt que des programmes non finalisés aux fonctionnalités approximatives.

Fractions

Fraction. Le résultat final devra respecter les modalités du TD mais ne recopiez pas directement ce que vous y avez fait ; suivez précisément les questions en tentant de comprendre les résultats intermédiaires obtenus (demandez éventuellement l'aide de l'enseignant responsable de la salle).

Créez un nouveau projet **fractions**. Par défaut, **netBeans** va créer une classe **Fractions** comportant une fonction **main** que vous utiliserez pour répondre aux questions et tester votre programme à chaque modification importante. Dans le même projet — **mais pas le même fichier !** —, créez aussi une classe **Fraction** (au singulier) qui contiendra toutes les caractéristiques d'une fraction (attributs, méthodes...).

1. Dans la classe **Fraction**, déclarez les attributs (variables d'instances) nécessaires.
2. Définissez les accesseurs (**getXXX** et **setXXX**) correspondant à ces attributs en faisant en sorte qu'il ne soit pas possible d'affecter au dénominateur 0 ni de valeur négative¹. À noter que, au lieu de considérer la donnée d'un dénominateur négatif comme erronée, il est possible de changer le signe du numérateur.
3. Placez les modificateurs d'accès **public** et **private** comme indiqué en cours. Exécutez à nouveau votre programme test, puis modifiez-le en fonction des nouvelles modalités.
4. Créez un *constructeur standard*² pour la classe, puis un constructeur qui permet de créer une fraction à partir d'un seul entier (son dénominateur vaut 1). Créez aussi un *constructeur de clonage*³. Testez vos modifications, par exemple avec les lignes suivantes :

```
Fraction f1 = new Fraction(8,4);
System.out.println(f1.getNum());
Fraction f2 = new Fraction(3), f3 = new Fraction(f2)...
```

5. Dans le constructeur d'une instance, il est possible d'utiliser un des autres constructeurs. Il a conventionnellement pour nom générique **this(...)**. Si ce n'est déjà fait, modifiez les constructeurs non standard pour qu'ils fassent appel au constructeur standard.
6. Écrivez la *méthode virtuelle*⁴ **getValeur** qui renvoie la valeur numérique issue du calcul de la fraction. Testez votre programme avec les fractions $\frac{8}{4}$, $\frac{3}{8}$, $\frac{3}{2}$... Expliquez les résultats obtenus s'ils vous paraissent erronés et modifiez éventuellement votre programme en consé-

¹ Par convention — et pour simplifier les traitements, le signe éventuel de la fraction sera porté par le numérateur.

² *Constructeur standard* : il donne une valeur à tous les attributs de base de l'instance — qui la définissent — et comporte donc un paramètre pour chacun de ces attributs.

³ *Constructeur de clonage* : il prend pour paramètre un objet du type de la classe et recopie ses caractéristiques dans l'instance courante. Ici, il prend un objet **Fraction** pour modèle.

⁴ Une fonction qui ne renvoie pas la valeur d'un attribut — comme le font les méthodes *classiques*. Elle caractérise une propriété dérivée de l'instance obtenue par calcul à partir des valeurs des attributs — qui représentent les propriétés de base, celles qui définissent l'instance.

quence.

- Écrivez la méthode `getFois` qui restitue la fraction obtenue en multipliant l'instance courante par un facteur numérique. Écrivez les méthodes `getPlus`, `getMoins`, `getFois`, `getSur` qui permettent respectivement d'ajouter, de soustraire, de multiplier et de diviser l'instance courante par une autre fraction en restituant le résultat sous forme d'un autre objet `Fraction`.
- Écrivez les lignes qui permettent d'effectuer le calcul suivant en créant les instances idoines :

$$\left(\frac{3}{2} - \frac{2}{5}\right) \times 2 - \frac{4}{5} \times \frac{-3}{4}$$

- Écrivez la fonction `static getPgcd(...,...)` qui renvoie le pgcd de deux nombres. Utilisez le code vu au premier semestre ou trouvez-le sur l'internet. Vérifiez son fonctionnement puis affectez-lui le modificateur d'accès `private`.
- Écrivez la méthode `getSimplification` qui restitue une nouvelle fraction obtenue en simplifiant la fraction courante (division du numérateur et du dénominateur par son pgcd). Expliquez pourquoi l'accès à la fonction `pgcd` est interdit à l'extérieur de l'instance et pourquoi il est ici fait mention de la *fonction* et pas de la *méthode* `pgcd`.
- Écrivez la méthode `toString` de telle manière que la fraction soit représentée de la manière la plus simple (sans dénominateur s'il vaut 1 et avec juste la chaîne "0" si le numérateur est nul).

Si tout est fini...

Créez un menu dans la méthode `main` de la classe principale pour effectuer des opérations sur les fractions. Par exemple :

- Entrée d'une fraction comme opérande principal (un seul élément de menu),

Quand cela est réalisé, le système affiche et peut exécuter en plus les opérations suivantes :

- Affichage de l'opérande principal,
- Multiplication de l'opérande principal par un facteur entier,
- Restitution de la valeur numérique correspondant à l'opérande principal,
- Simplification de l'opérande principal,
- Affichage du dernier calcul effectué,
- Le résultat du dernier calcul devient premier opérande,
- Entrée d'une fraction comme opérande secondaire,
- Effacement de l'opérande principal, ce qui a pour effet d'effacer aussi l'opérande secondaire et de retourner au seul choix 1.
- Fin.

Le résultat des opérations est affecté à une variable qui sera utilisée pour les choix 6 et 7. L'exécution du choix 8 permet d'ouvrir aux fonctionnalités supplémentaires suivantes :

- Affichage de l'opérande secondaire,
- Inversion des opérandes principaux et secondaires,
- /13./14./15. Addition/Soustraction/Multiplication/Division des deux opérandes,
- Effacement de l'opérande secondaire. Retour au calcul sur un seul opérande.

Testez votre programme en chaînant des opérations sur des fractions. Vous pouvez aussi ajouter la mise en mémoire d'une ou plusieurs valeurs.