

Demande de changement #2

Votre client vous demande de lui livrer ces fonctionnalités pour le 30 mars 2017, avant 17h30. La date de livraison n'est pas négociable.

Le contenu de ce document s'ajoute à toutes les demandes précédentes.

Exigences fonctionnelles

Maximum mensuel

Nous ajoutons un nouveau concept à notre logique d'affaire : le maximum mensuel. Certains types de soin ont un montant maximal mensuel de remboursement. Autrement dit, en plus de toutes les autres règles déjà connues, si la somme des réclamations pour un soin en particulier dépasse le maximum mensuel permis pour ce soin, on rembourse jusqu'au montant maximal mensuel et l'excédent n'est pas remboursé.

Supposons qu'il existe un montant mensuel maximal de 200\$ pour le soin 175 (Médecin généraliste privé). En considérant le fichier d'entrée suivant :

```
{
  "dossier": "A100323",
  "mois": "2017-01",
  "reclamations": [
    {
      "soin": 175,
      "date": "2017-01-11",
      "montant": "130.00$"
    },
    {
      "soin": 175,
      "date": "2017-01-14",
      "montant": "130.00$"
    },
    {
      "soin": 175,
      "date": "2017-01-15",
      "montant": "130.00$"
    },
    {
      "soin": 175,
      "date": "2017-01-17",
      "montant": "130.00$"
    }
  ]
}
```

Nous obtiendrions le fichier de sortie suivant :

```
{
  "dossier": "A100323",
  "mois": "2017-01",
  "remboursements": [
    {
      "soin": 175,
      "date": "2017-01-11",
      "montant": "65.00$"
    },
    {
      "soin": 175,
      "date": "2017-01-14",
      "montant": "65.00$"
    },
    {
      "soin": 175,
      "date": "2017-01-15",
      "montant": "65.00$"
    },
    {
      "soin": 175,
      "date": "2017-01-17",
      "montant": "5.00$"
    }
  ],
  "total": "200.00$"
}
```

Voici les maximums mensuels pour les types de soin qui en possèdent :

Numéro de soin	Catégorie de soin	Maximum mensuel
100	Ostéopathie	250.00\$
175	Médecin généraliste privé	200.00\$
200	Psychologie individuelle	250.00\$
500	Chiropratie	150.00\$
600	Physiothérapie	300.00\$

Manipulation monétaire sécuritaire

Il est généralement connu des développeurs qu'il ne faut jamais manipuler des valeurs monétaires avec des types de données réels (ex. float, double). Ces types sont des approximations qui permettent une grande précision pour les nombres fractionnaires mais qui représentent mal les valeurs monétaires. Quelques manipulations monétaires simples sur un type réel suffisent à faire apparaître ou disparaître quelques sous, ce qui n'est définitivement pas souhaitable pour un système qui manipule de l'argent.

Une façon courante de régler ce problème est de créer une classe «Dollar» qui contient toute la logique de manipulation de la monnaie. Souvent, la donnée interne est la valeur monétaire en cents. Ainsi, 120.21\$ est représenté comme 12021 cents. Une division entière par 100 nous permet d'obtenir le nombre de dollars et un modulo 100 nous permet d'obtenir les cents. Comme la valeur monétaire est toujours représentée avec une valeur entière (les cents), nous n'avons plus les

problèmes occasionnés par les points flottants.

Vous devrez donc créer une classe «Dollar» qui permettra d'effectuer toutes vos manipulations monétaires de façon sécuritaire. Chaque calcul fait sur une valeur monétaire devra passer par la classe «Dollar». Vous devrez également écrire des tests unitaires sur chacune des fonctionnalités de cette classe pour vous assurer qu'elle fonctionne parfaitement bien.

Statistiques

Le logiciel devra maintenant accumuler des statistiques lors de chaque exécution. Vous devrez conserver les statistiques dans un fichier. Le fichier peut être de la nature que vous voulez (texte, JSON, XML ou autre, tant que ça ne nécessite aucune installation).

À chaque exécution, vous devrez calculer les statistiques suivantes :

- le nombre de réclamations valides traitées;
- le nombre de réclamations rejetées;
- le nombre de soins déclarés pour chaque type de soin.

Pour produire la liste des statistiques, l'utilisateur exécutera le logiciel en spécifiant, en paramètres, l'option "-S" et les statistiques seront affichées à la console. Par exemple :

```
java -jar Refund -S
```

L'option "-SR" doit réinitialiser les statistiques, c'est-à-dire les remettre à 0. Un message de confirmation doit être affiché à la console.

```
java -jar Refund -SR
```