Aufgabe 3 Klassen –abstrakte Klassen - Interfaces

TEIL A:

Sie sollen Arithmetik und Konvertierungen für Währungen entwerfen und implementieren.

Dazu soll ein Entwurf entstehen, der Gemeinsamkeiten an geeigneten Stellen der Klassen und Interfacehierarchie abstrahiert (implementiert).

Wenn Sie die Lösung nach Anleitung entwerfen, dann können Sie, ohne an den Interfaces und den abstrakten Klasse Änderungen vornehmen zu müssen, beliebige neue Währungen hinzufügen.

Im Einzelnen sind zu entwerfen / zu implementieren:

Schnittstelle WaehrungsGroesse

Die Schnittstelle ***WaehrungsGroesse*** für die Arithmetik von Währungen fordert die Implementierung der Operationen

- ***add, sub*** deren Parameter und Ergebnis von Typ ***WaehrungsGroesse*** sind.
- ***mult, div*** deren Parameter skalare Werte (***long***) und deren Ergebnis vom Typ ***WaehrungsGroesse*** sind

Weiterhin fordert die Schnittstelle die Konvertierungsmethoden

- ***toBase()***, das die Groesse in einen allen Größen gemeinsamen Basiswert (***long***) umrechnen soll
- ***fromBase(long baseVal)***, die aus einem Basiswert eine ***WaehrungsGroesse*** konstruiert
- ***convertTo(WaehrungsGroesse other)*** , die als Ergebnis ein Währungsobjekt von Typ des Objektes ***other*** zurückliefert.

Abstrakte Klasse AbstractWaehrungsGroesse

Diese Klasse enthält eine Reihe von **generischen Implementierungen** für alle konkreten Währungsklassen:

- liefert die Implementierung der arithmetischen Operationen der Schnittstelle ***WaehrungsGroesse*** für alle Währungen Der Ergebnistyp des neuen Währungsobjektes ist vom selben Typ wie ***this***. Verwendet dazu die Methoden Methoden ***toBase*** und ***fromBase.***
- liefert die Implementierung ***mult / div*** für alle Währungen. Der Ergebnistyp des neuen Währungsobjektes ist vom selben Typ wie ***this***. Verwendet dazu die Methoden Methoden ***toBase*** und ***fromBase.***
- implementiert die Methode ***convertTo,*** und verwendet dazu die Methoden ***toBase*** und ***fromBase*** verwendet.
- Implementiert die Methode ***toBase()*** für alle Währungen: Verwendet dazu die Methoden ***getInternalBase*** und ***getConversionFactor***

- liefert eine Implementierung ***toString()*** für alle konkreten Subklassen, die die Geldeinheit in Euro –Cent, Pfund-Pence, Dollar-Cent darstellen, wenn der Betrag keine vollen Euro's, Pfund oder Dollar's darstellt: Verwendet dazu die Methoden ***getInternalBase*** und ***getSymbols.***

Darüber hinaus fordert die Klasse von den Subklassen die Implementierung von 3 Methoden. Da diese Methoden nur für die Implementierung in der abstrakten Klasse relevant sind, werden diese als ***protected*** deklariert.

- ***getInternalBase***: Liefert für jede Währung die interne Darstellung des Wertes als long Wert (Basis für Euro und Dollar ist Cent, Basis für Pfund ist Pence)
- ***getConversionFactor***: Liefert den Umrechnungsfaktor in Euro
- **getSymbols**: liefert ein Objekt vom Typ **WaehrungsSymbole**, das die einzelnen Symbole für die Darstellung der jeweiligen Währung kennt z.B. **(€,ct)** für Euro **($,¢)** für Dollar **(£,p)** für britische Pfund**.**

Schnittstelle WaehrungsSymbole

- ***getFirst***: liefert die Darstellung für die größere Währungseinheit also eines aus **(€, $, £)**
- ***getSecond***: liefert die Darstellung für die kleiner Währungeinheit also eines aus (**ct,¢, p)**

Für jede konkrete Währungsgröße gibt es eine lokale Implementierung für diese Schnitsttelle.

Konkrete Klassen Euro, GBP, USD

Implementieren die Konvertierungsmethoden

- ***toBase***/ ***fromBase*** indem Sie in die Basis Euro-Cent umrechnen. Bei der Umrechnung entstehen in der Regel Beträge, die nicht in vollen Euro-Cent darstellbar sind. Die Nachkommastellen sollen gerundet werden.
- ***getInternalBase*** : Liefert für jede Währung die interne Darstellung des Wertes als long Wert (Basis für Euro und Dollar ist Cent, Basis für Pfund ist Pence)
- ***getConversionFactor***: Liefert den Umrechnungsfaktor in Euro
- **getSymbols**: liefert ein Objekt vom Typ **WaehrungsSymbole**, das die einzelnen Symbole für die Darstellung der jeweiligen Währung kennt z.B. **(€,ct)** für Euro **($,¢)** für Dollar **(£,p)** für britische Pfund**.**
- erzeugt werden die Geldeinheiten mit Konstruktoren, die den Wert in Cent/Pence oder Euro/Pfund/Dollar plus Cent/Pence übergeben.

Teil B:

Testen Sie Ihre Lösung in JUnit Tests, die eine vollständige Testabdeckung enthalten.

Teil C:

Lesen Sie aus Zeichenketten Objekte für Währungen und Längenangaben ein und nutzen Sie reguläre Ausdrücke, um den konkreten Typ der Objekte zu bestimmen.

5 € 20 Cent

5,69 €

800 Cent

8,89$

8 $ 89 ¢

70 £ 67p

70,67 £