

Übungen zu „Software-Entwicklung“

Klassen

Profs. Dres. Rathke & Thies

Bei der letzten Besprechung mit der Spar Nix Bank AG gab es sehr viele positive Anmerkungen von Bodo Bank und seinen Kollegen zu Ihrer überarbeiteten Version der Bankverwaltung, welche umgehend den Kunden zur Verfügung gestellt wurde. Nun können sich die Kunden der Bank ihre Kontoauszüge sehr übersichtlich ausdrucken.

Bei dieser Besprechung sind weitere Ideen aufgekommen, wie sich die Bankverwaltung weiterentwickeln lässt, damit sie die zukünftigen internationalen Bestimmungen für Bankensoftware einhalten kann. Bei dieser Besprechung ist folgende ToDo Liste entstanden:

- 1) Laden Sie sich die Datei 'IT1_Banking_10_1.zip' und importieren Sie das Projekt in Eclipse.
- 2) Fügen Sie zur Klasse `Transaction` die Instanzvariable `transactionId` vom Datentyp `long` hinzu und weisen Sie ihr den Initialwert `0L` zu.
- 3) Fügen Sie zur Klasse `Transaction` die Methode `getTransactionId()` hinzu, die als Rückgabewert den Wert der Variable `transactionId` zurück gibt.
- 4) Fügen Sie zur Klasse `Transaction` die Klassenvariable `transactionIdCounter` vom Datentyp `long` hinzu und weisen Sie ihr den Initialwert `100000000L` zu.
- 5) Fügen Sie zur Klasse `Transaction` die Methode `getTransactionIdCounter()` hinzu, die als Rückgabewert den Wert der Variable `transactionIdCounter` zurück gibt.
- 6) Erstellen Sie einen Konstruktor für die Klasse `Transaction`: weisen Sie der Variable `transactionId` den Wert der Variable `transactionIdCounter` zu. Erhöhen Sie anschließend den Wert der Variable `transactionIdCounter` um `1`.
- 7) Überarbeiten Sie in der Klasse `Account` die Methode `getAccountStatement()`. Es soll bei der Erstellung des Kontoauszugs zusätzlich noch die `transactionId` vor dem Datum der `Transaction` ausgegeben werden.
- 8) Überprüfen Sie Ihre Implementierung mit Hilfe der GUI. Erstellen Sie hierzu ein neues Konto für Erna Sparbier und führen Sie folgende Buchungen durch:

Buchungswert:	Buchungstext:
€ 2500.-	„Gehalt für Dezember 2010 inkl. Weihnachtsgeld“
€ - 660.-	„Miete für 12/10“
€ - 113.-	„Monatsmarke VVS Dezember 2010“
€ - 67.-	„Einzugsservice Telekommunikation & Internet 12/10“

Lassen Sie sich nach der Durchführung der Buchungen den Kontostand anzeigen: Wählen Sie hierzu im Menü 'Konto' den Menüeintrag 'Kontoauszug anzeigen'.

- 9) Fügen Sie zur Klasse `Account` konstante Klassenvariablen vom Datentyp `int` für folgende Währungen hinzu: EUR, CHF, USD und YEN. Jede Währung soll hierbei durch einen `int` Wert repräsentiert werden. Andere Klassen sollen auf diese Variablen zugreifen können. Dies soll dazu dienen, dass in Zukunft eine Währung hinterlegt werden kann, in der das Konto geführt wird.
- 10) Fügen Sie zur Klasse `Account` die Instanzvariable `currency` vom Datentyp `int` mit dem Initialwert `0` hinzu. Hier soll die Währung hinterlegt werden, in der dieses Konto geführt wird.

- 11) Fügen Sie zur Klasse `Account` die Methode `getCurrency()` hinzu, welche die Währung, in der das Konto geführt wird, als `int` Wert zurück gibt.
- 12) Überarbeiten Sie in der Klasse `Account` den Konstruktor `public Account(customer owner, int currency)`, so dass beim Erstellen eines neuen Kontos die Währung, in der das Konto geführt werden soll, mit übergeben werden kann.
- 13) Fügen Sie zur Klasse `Account` die statische Methode `currencyToString(int currency)` hinzu, die die textuelle Repräsentation der dazugehörigen Währung als `String` zurück gibt.
- 14) Überarbeiten Sie in Klasse `Account` die Methode `getCurrencyAsString()`, welche die Währung, in der das Konto geführt wird, als `String` zurück gibt. Verwenden Sie als Hilfe hierzu die Methode `currencyToString(int currency)`.
- 15) Modifizieren Sie die Methode `getAccountStatement()` der Klasse `Account` so, dass zu Beginn des Kontoauszuges die Währung in der das Konto geführt wird, ausgegeben wird.
- 16) Überprüfen Sie Ihre Implementierung mit Hilfe der GUI. Erstellen Sie hierzu ein neues Konto für Erna Sparbier und führen Sie folgende Buchungen durch:

Buchungswert:	Buchungstext:
€ 2500.-	„Gehalt für Dezember 2010 inkl. Weihnachtsgeld“
€ - 660.-	„Miete für 12/10“
€ - 113.-	„Monatsmarke VVS Dezember 2010“
€ - 67.-	„Einzugsservice Telekommunikation & Internet 12/10“

Lassen Sie sich nach der Durchführung der Buchungen den Kontostand anzeigen: Wählen Sie hierzu im Menü 'Konto' den Menüeintrag 'Kontoauszug anzeigen'.

Lösung:

Zu 2)

```
private long transactionId = 0L;
```

Zu 3)

```
public long getTransactionId() {
    return transactionId;
}
```

Zu 4)

```
private static long transactionIdCounter = 100000000L;
```

Zu 5)

```
public long getTransactionIdCounter() {
    return transactionIdCounter;
}
```

Zu 6)

```
public Transaction(){
    transactionId=transactionIdCounter;
    transactionIdCounter++;
}
```

Zu 7)

```

public String getAccountStatement() {
    StringBuffer result = new StringBuffer();
    Transaction tempTransaction = null;
    double tempBalance = this.getBalance();

    result.append("Kontoauszug fuer : " + this.getOwner() + "\n");
    result.append("Kontonummer : " + this.getId() + "\n");
    result.append("\n");
    result.append("Kontostand zu Beginn: "
        + this.normalizeDoubleValue(this.getBalance()) + "\n");

    for (int i = 0; i < this.getTransactionPointer(); i++) {
        tempTransaction = this.getTransactions()[i];
        tempBalance = tempBalance + tempTransaction.getAmount();

        result.append(tempTransaction.getTransactionId());
        result.append(": ");

        result.append(tempTransaction.getDate());
        result.append(" ");

        String tempText = tempTransaction.getText();
        if (tempText.length() <= 20) {
            while (tempText.length() < 20) {
                tempText += " ";
            }
        } else {
            tempText = tempText.substring(0, 20);
        }
        result.append(tempText);

        result.append(" ");
        result.append(
            this.normalizeDoubleValue(tempTransaction.getAmount()));
        result.append(" ");
        result.append(this.normalizeDoubleValue(tempBalance));
        result.append("\n");
    }
    return result.toString();
}

```

Zu 9)

```

public final static int EUR = 0;
public final static int CHF = 1;
public final static int USD = 2;
public final static int YEN = 3;

```

Zu 10)

```

private int currency = 0;

```

Zu 11)

```

public int getCurrency() {
    return this.currency;
}

```

Zu 12)

```

public Account(Customer owner, int currency) {
    this.owner = owner;
    this.currency = currency;
}

```

Zu 13)

```
public static String currencyToString(int currency) {
    switch (currency) {
        case EUR:
            return "EUR";
        case CHF:
            return "CHF";
        case USD:
            return "USD";
        case YEN:
            return "Yen";
        default:
            return Integer.toString(currency);
    }
}
```

Zu 14)

```
public String getCurrencyAsString() {
    return currencyToString(this.currency);
}
```

Zu 15)

```
public String getAccountStatement() {
    StringBuffer result = new StringBuffer();
    Transaction tempTransaction = null;
    double tempBalance = this.getBalance();

    result.append("Kontoauszug fuer : " + this.getOwner() + "\n");
    result.append("Kontonummer : " + this.getId() + "\n");
    result.append("Waehrung : " + this.getCurrencyAsString());
    result.append("\n");
    result.append("Kontostand zu Beginn: "
        + this.normalizeDoubleValue(this.getBalance()) + "\n");

    for (int i = 0; i < this.getTransactionPointer(); i++) {
        tempTransaction = this.getTransactions()[i];
        tempBalance = tempBalance + tempTransaction.getAmount();

        result.append(tempTransaction.getTransactionId());
        result.append(": ");

        result.append(tempTransaction.getDate());
        result.append(" ");

        String tempText = tempTransaction.getText();
        if (tempText.length() <= 20) {
            while (tempText.length() < 20) {
                tempText += " ";
            }
        } else {
            tempText = tempText.substring(0, 20);
        }
        result.append(tempText);

        result.append(" ");
        result.append(
            this.normalizeDoubleValue(tempTransaction.getAmount()));
        result.append(" ");
        result.append(this.normalizeDoubleValue(tempBalance));
        result.append("\n");
    }
    return result.toString();
}
```