

Übungen zu „Software-Entwicklung“

Zeichen und Zeichenketten

Profs. Dres. Rathke & Thies

Mit Ihrer Hilfe konnte Bodo Bank nun endlich erfolgreich die Klassen `Account`, `Bank` und `Transaction` des Bank Projektes ausführlich testen und diese anschließend in die Produktivumgebung der Spar Nix AG übernehmen. Es war ihm sogar etwas peinlich Ihre Hilfe für seine kleinen Leichtsinnfehler in Anspruch nehmen zu müssen. In der Testphase sind allerdings noch ein paar kleine Änderungen in die Klasse `Account` eingeflossen.

Damit nun auch die Kunden der Bank auch die Vorteile der überarbeiteten Kontenverwaltung nutzen können, müssen allerdings noch ein paar kleine Änderungen in die Anwendung einfließen.

Im letzten Meeting mit der Spar Nix Bank AG ist folgende ToDo Liste entstanden, die Sie nun Schritt für Schritt bearbeiten müssen:

- 1) Laden Sie die Datei 'IT1_Banking_09_1.zip' und importieren Sie das Projekt in Eclipse.
- 2) Überarbeiten Sie in der Klasse `Customer` (im Paket: `de.hdm.bankProject.data`) die Methode `toString()`. Diese soll als textuelle Repräsentation des Kunden den Vor- und Nachnamen als `String` zurückgeben.
- 3) Überarbeiten Sie in der Klasse `Customer` die Methode `nameLength()`. Diese Methode soll mit Hilfe der `toString()` Methode die Länge des Strings als `int` Wert zurückgeben. Überprüfen Sie Ihre Implementierung mit Hilfe der GUI. Wählen Sie hierzu zuerst einen Kunden aus und anschließend im Menü 'Übungen' den Eintrag 'Namenslänge' um sich die Länge des Strings ausgeben zu lassen.
- 4) Überarbeiten Sie in der Klasse `Account` die Methode `getAccountStatement()`. Diese Methode soll als Rückgabewert den Kontoauszug als `String` zurückgeben. Kopieren Sie hierzu den Code aus der Methode `printAccountStatement()` in die Methode `getAccountStatement()` und verändern sie diesen Code so, dass zur Erstellung der Ausgabe ein `StringBuffer` verwendet wird.

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'.

- 5) Verändern Sie die Methode `getAccountStatement()` in der Klasse `Account` so, dass der Buchungstext jeder Transaction, sollte dieser kürzer als 20 Zeichen sein, mit Leerzeichen verlängert wird, bis dieser String die Länge von 20 Zeichen hat. Überprüfen Sie Ihre Implementierung mit Hilfe der GUI (siehe Aufgabe 4).
- 6) Verändern Sie die Methode `getAccountStatement()` in der Klasse `Account` so, dass der Buchungstext jeder Transaction, sollte dieser länger als 20 Zeichen sein, auf die Länge von 20 Zeichen gekürzt wird. Überprüfen Sie Ihre Implementierung mit Hilfe der GUI (siehe Aufgabe 4).

- 7) Fügen Sie zur Klasse Account die Methode `normalizeDoubleValue(double number)` hinzu. Diese Methode soll aus der Zahl, welche sie per Methodenaufruf übergeben bekommt, einen `String` generieren und diesen so verändern, dass dieser genau zwei Ziffern nach dem letzten Punkt hat.

Beispiele:

632.0 → 632.00

632.764 → 632.76

- 8) Verändern Sie in der Klasse Account die Methode `getAccountStatement()` so, dass für die Darstellung der Beträge die Methode `normalizeDoubleValue()` verwendet wird. Überprüfen Sie Ihre Implementierung mit Hilfe der GUI (siehe Aufgabe 4).

Lösung:

Zu 2)

```
public String toString() {
    return this.firstName + " " + this.lastName;
}
```

Zu 3)

```
public int nameLength() {
    return this.toString().length();
}
```

Zu 4)

```
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.getAccountNumber() + "\n");
    result.append("\n");
    result.append("Kontostand zu Beginn: " + this.getBalance() + "\n");

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

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

Zu 5)

```
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.getAccountNumber() + "\n");
    result.append("\n");
```

```

result.append("Kontostand zu Beginn: " + this.getBalance() + "\n");

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

    result.append(tempTransaction.getDate());
    result.append(" ");
    String tempText = tempTransaction.getText();
    if(tempText.length() <= 20) {
        while(tempText.length() < 20) {
            tempText += " ";
        }
    }
    result.append(tempText);
    result.append(" ");
    result.append(tempTransaction.getAmount());
    result.append(" ");
    result.append(tempBalance);
    result.append("\n");
}
return result.toString();
}

```

Zu 6)

```

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.getAccountNumber() + "\n");
    result.append("\n");
    result.append("Kontostand zu Beginn: " + this.getBalance() + "\n");

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

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

```

Zu 7)

```

public String normalizeDoubleValue(double number) {
    StringBuffer buffer = new StringBuffer(String.valueOf(number));
    String substring = buffer.substring(buffer.lastIndexOf(".") + 1);

    if (substring.length() < 2) {
        return buffer + "0";
    } else {
        return buffer.substring(0, buffer.lastIndexOf(".") + 3);
    }
}

```

```

    }
}

```

Zu 8)

```

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.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();
}

```