Inhalt

[(A) Einsteigeraufgaben 4](#_Toc25653824)

[A1) 4](#_Toc25653825)

[A2) 4](#_Toc25653826)

[A3) 4](#_Toc25653827)

[A4) 4](#_Toc25653828)

[(B) Einfache if-Verzweigungen 4](#_Toc25653829)

[B1) 4](#_Toc25653830)

[B2) 5](#_Toc25653831)

[B3) 5](#_Toc25653832)

[(C) Einfache Schleifen 5](#_Toc25653833)

[(C1) For-Schleifen 5](#_Toc25653834)

[C1-1) 5](#_Toc25653835)

[C1-2) 5](#_Toc25653836)

[C1-3) 5](#_Toc25653837)

[(C2) While-Schleifen 6](#_Toc25653838)

[C2-1) 6](#_Toc25653839)

[C2-2) 6](#_Toc25653840)

[C2-3) 6](#_Toc25653841)

[C2-4) 6](#_Toc25653842)

[(D) Einfache foreach-Schleifen 6](#_Toc25653843)

[D2) Kundenliste 6](#_Toc25653844)

[D2-1) 6](#_Toc25653845)

[D2-2) 7](#_Toc25653846)

[D3) Reichster Mann 7](#_Toc25653847)

[D3-1) 7](#_Toc25653848)

[D3-2) 7](#_Toc25653849)

[(E) Kombination: Schleife/if - 7](#_Toc25653850)

[(F) Anspruchsvollere Aufgaben 7](#_Toc25653851)

[F1) 8](#_Toc25653852)

Aufgaben Struktogramme + Kontrollstrukturen

Durchzuführen am Beispiel der Lösung zu Abiaufgabe 2014, Aufg. 2.3 (Girokonten, Kundenbetreuer …)

**Alle Aufgaben sind in der Klasse Kundenbetreuer durchzuführen**. Beachten Sie, dass im Klassendiagramm der Kundenbetreuer eine Liste "kunden" hat, die im Projekt der besseren Verständlichkeit als kontoinhaberListe benannt wurde.

**Bitte IMMER ZUERST ein Struktogramm anfertigen, bevor Sie zu programmieren beginnen!**

Alle programmierten Methoden auf korrekte Lauffähigkeit prüfen (via Startklasse).

Sie benötigen folgende Dateien:

**zusammenfasssende\_aufgaben\_struktogramme\_kontrollstrukturen.docx** (diese Datei – hier kommen die Lösungen rein)

**191126\_bank** – eclipse-Projekt

**191126\_bank/assets/KLASSENDIAGRAMM\_loesungsvorschlag\_abi2014\_aufgabe23.png** – Klassendiagramm zum Projekt (mal von einer Schülerin gemacht)



##  (A) Einsteigeraufgaben

### A1)

Programmieren Sie eine Methode datenAusgeben(kontoinhaber:Kontoinhaber):void, die die Daten eines Kunden ausgibt in Form

Heini Müller - Kontostand: 3000 Euro

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

### A2)

Wie A1, nur wird der String nicht AUSGEGEBEN, sondern ZURÜCKGEGEBEN. Die Methode heißt dann datenZurueckgeben(kontoinhaber:Kontoinhaber):String

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

### A3)

Programmieren Sie eine Methode anzahlInhaber():int, die die Anzahl der von einem Betreuer betreuten Kontoinhaber zurückgibt.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

### A4)

Programmieren Sie eine Methode listeMeinerBetreutenKunden():ArrayList<Kontoinhaber>, die eine Liste aller betreuten Kunden ZURÜCKGIBT.

(Wenn Sie das in der Startklasse ausprobieren, benötigen Sie für die Anzeige eine foreach-Schleife.)

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

## (B) Einfache if-Verzweigungen

### B1)

Programmieren Sie eine Methode heisstKundeMetz(kontoinhaber:Kontoinhaber):boolean, die prüft, ob der Nachname eines Kunden "Metz" ist. Zur Erinnerung: Verwenden Sie bei String-Vergleichen nicht "==", sondern "equals(…)", also bspw. kundename.equals("Metz")

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

### B2)

Programmieren Sie eine Methode kundePruefungAusgabe(kontoinhaber:Kontoinhaber):void, die abhängig vom Kontostand ausgibt:

Heini Müller hat mehr als 5000 Euro auf dem Konto.

oder

Heini Müller hat 5000 Euro oder weniger auf dem Konto.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

### B3)

Programmieren Sie eine Methode kundePruefung(kontoinhaber:Kontoinhaber):boolean, die abhängig vom Kontostand eines Kunden true (bei mehr als 5000 Euro) oder false (bei 5000 oder weniger) zurückgibt.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

##  (C) Einfache Schleifen

###  (C1) For-Schleifen

C1-1)

Programmieren Sie eine Methode zaehleBis(ende:int), die von 0 bis ende zählt und die Zahlen ausgibt.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

C1-2)

Programmieren Sie eine Methode zaehleVonBis(start:int, ende:int), die von start bis ende zählt und die Zahlen ausgibt.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

C1-3)

Programmieren Sie eine Methode zaehleVonBisAbstand(start:int, ende:int, abstand:int), die von start bis ende zählt und die Zahlen ausgibt.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

###  (C2) While-Schleifen

C2-1)

Programmieren Sie eine Methode whileZaehlen(), die von 0 bis 20 zählt und die Zahlen ausgibt. Verwendung einer while-Schleife.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

C2-2)

Programmieren Sie eine Methode whileZaehlenVon(start:int), die von start bis 20 zählt und die Zahlen ausgibt. Verwendung einer while-Schleife.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

C2-3)

Programmieren Sie eine Methode whileZaehlenVonBis(start:int, ende:int), die von start bis ende zählt und die Zahlen ausgibt. Verwendung einer while-Schleife.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

C2-4)

 Programmieren Sie eine Methode whileZaehlenVonBisSchritt(start:int, ende:int, abstand:int), die von start bis ende in Schrittweite abstand zählt und die Zahlen ausgibt. Verwendung einer while-Schleife.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

##  (D) Einfache foreach-Schleifen

### D2) Kundenliste

D2-1)

Programmieren Sie eine Methode listeMeinerBetreutenKundenAlsString():void, die eine Liste aller betreuten Kunden AUSGIBT (name, vorname, kontonummer, Kontostand). Spalten mit \t erzeugen.

Name Vorname KtoNr. KtoStand

Schmitt Heini 3321314 45 Euro

Müller Bilal 587373 28348 Euro

(Es kann dabei zu Verschiebungen kommen, wenn Elemente besonders lang oder kurz sind)

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

D2-2)

Programmieren Sie eine Methode listeMeinerBetreutenKundenAlsStringRueckgabe():String, die eine Liste wie oben ZURÜCKGIBT. Sie müssen dazu D2-1 nur geringfügig umarbeiten.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

### D3) Reichster Mann

D3-1)

Programmieren Sie eine Methode reichsterMannKontostand():double, die den Kontostand des Kontoinhabers mit dem höchsten Kontostand zurückgibt.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

D3-2)

Programmieren Sie eine Methode reichsterMann():Kontoinhaber, die den Kontoinhaber mit dem höchsten Kontostand zurückgibt.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

##  (E) Kombination: Schleife/if -

Programmieren Sie eine Methode listeMeinerBetreutenReichenKunden():String, die eine Liste wie in Aufgabe D2-1 ZURÜCKGIBT (als String!), allerdings sind dort nur die Kontoinhaber mit einem Kontostand von mehr als 500 Euro enthalten.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren

## (F) Anspruchsvollere Aufgaben

### F1)

Programmieren Sie eine Methode reichsteKunden():ArrayList<Kontoinhaber>, die eine Liste aller Kunden zurückgibt, deren Kontostand über dem durchschnittlichen Kontostand aller Kunden liegt.

*LÖSUNG: Struktogramm*

Bild hier einfügen

*LÖSUNG: Code*

Code hier einfügen, als code formatieren