Aufgabe 1

Konto

-float kontostand

-String kontoinhaber

-int kontonummer

- getKontostand

-void setKontostand

-float abheben

-float einzahlen

-float überweisen

Aufgabe 2

**package** de.hdm.bankProject.data;

/\*\*

\* Klasse der Bankkonten

\* **@author** Thies

\*/

**public** **class** Account {

/\*\*

\*

\*/

**private** **float** balance = 0;

/\*\*

\* Kontonummer

\*/

**private** **int** id = 0;

/\*\*

\* Kontoinhaber

\*/

**private** Customer owner = **null**;

/\*\*

\* Auslesen der Kontonummer.

\* **@return** Kontonummer, ganzzahlig

\*/

**public** **int** getId() {

**return** **this**.id;

}

/\*\*

\* Auslesen des Kontoinhabers.

\* **@return** Kontoinhaber-Instanz

\* **@see** Customer

\*/

**public** Customer getOwner() {

**return** owner;

}

/\*\*

\* Setzen der Kontonummer. <br>

\* <b>Hinweis:</b> Sollte nur von Verwaltungsklassen genutzt werden!

\* **@param** i

\* **@see** setOwner(Customer customer)

\*/

**public** **void** setId(**int** i) {

**this**.id = i;

}

/\*\*

\* Setzen des Kontoinhabers. <br>

\* <b>Hinweis:</b> Sollte nur von Verwaltungsklassen genutzt werden!

\* **@param** c

\* **@see** setId(int i)

\*/

**public** **void** setOwner(Customer c) {

**this**.owner = c;

}

/\*\*

\* Eine Einzahlung bzw. Gutschrift fuer das Konto buchen.

\* **@param** amount

\* **@see** makeWithdrawal(int amount)

\*/

**public** **void** makeDeposit(**float** amount) {

**this**.balance = **this**.balance + amount;

}

/\*\*

\* Eine Auszahlung bzw. Belastung des Kontos buchen.

\* **@param** amount

\* **@see** makeDeposit(int amount)

\*/

**public** **void** makeWithdrawal(**float** amount) {

**this**.balance = **this**.balance - amount;

}

/\*\*

\* Feststellen der Gleichheit zweier Account-Objekte. Die Gleichheit wird in

\* diesem Beispiel auf eine identische Kontonummer beschraenkt.

\*/

@Override

**public** **boolean** equals(Object o) {

/\*

\* Abfragen, ob ein Objekt ungl. NULL ist und ob ein Objekt gecastet werden

\* kann, sind immer wichtig!

\*/

**if** (o != **null** && o **instanceof** Account) {

Account c = (Account) o;

**if** (c.getId() == **this**.id) {

**return** **true**;

}

}

**return** **false**;

}

@Override

**public** **int** hashCode() {

**int** hash = 7;

hash = 79 \* hash + **this**.id;

hash = 79 \* hash + Float.*floatToIntBits*(**this**.balance);

hash = 79 \* hash + **this**.owner.getId();

**return** hash;

}

**public** **float** balance() {

**return** balance;

}

**public** **void** setBalance (**float** balance){

**this**.balance = balance ;

}

}