

Java Grundlagen

Arrays

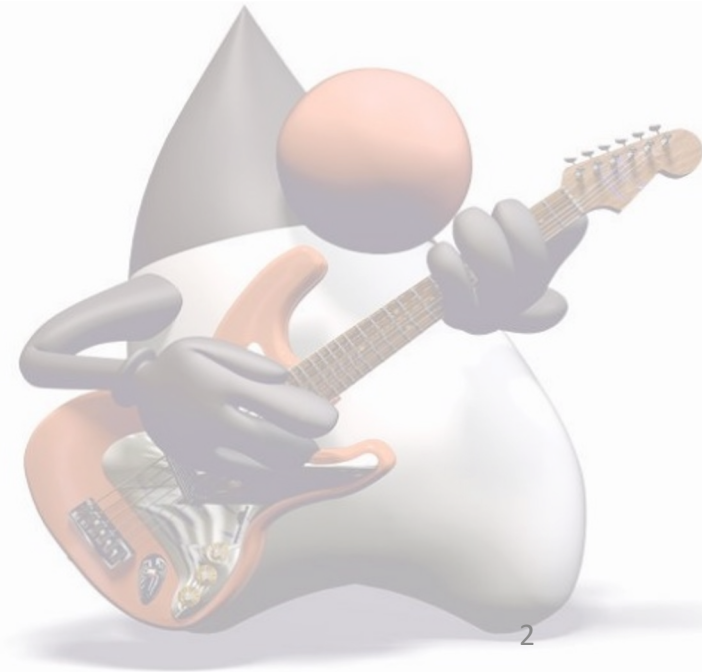


Was ist ein Array

- In einem Array kann man eine große Anzahl von Datenfeldern des selben Typs speichern
- Zum Beispiel: 100 Integer, 20 Strings, 15 Klassen Objekte (z.B Kunden)

- Ein Array deklarieren

```
int[] numbers;  
char[] letters;  
Customer[] customers;  
String[] names;
```



Arrays sind Referenz Datentypen

- Sie müssen instanziiert werden bevor sie verwendet werden können
- Bei der Instanziierung muss die Größe mitgegeben werden
- Instantiating arrays:

```
int[] numbers = new int[100];  
char[] letters = new char[20];  
Customer[] customers = new Customer[10];  
String[] names = new String[10000];
```



Arrays mit primitiv Datentypen

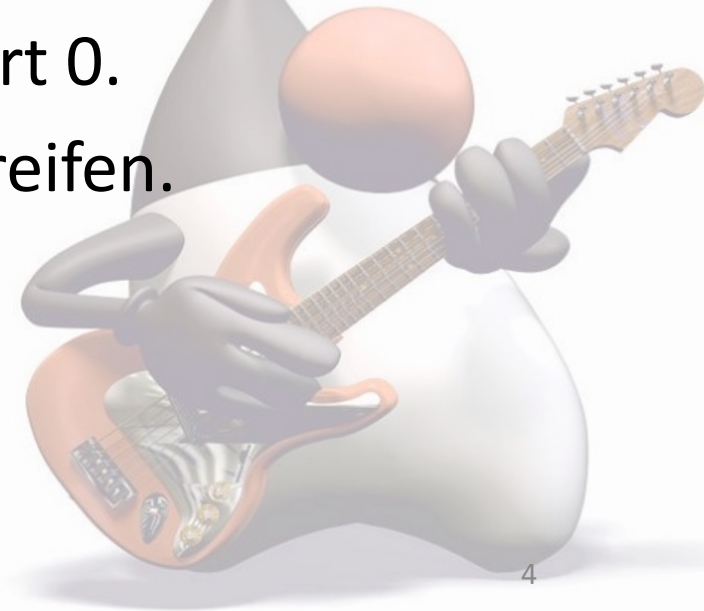
- `int[] numbers = new int[5];`

- Nach dem Initialisieren haben alle Elemente den Wert 0.
- Auf die einzelnen Elemente kann man per Index zugreifen.

`numbers[0] = 7;`

`numbers[4] = 12;`

| |
|---|
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |



Durch ein Feld iterieren

```
public static void main(String[] args) {  
    int[] numbers = new int[20];  
  
    for (int i=0;i<numbers.length;i++){  
        numbers[i] = i*2;  
    }  
}
```

Kapazität des Arrays



Arrays mit Referenz Datentypen

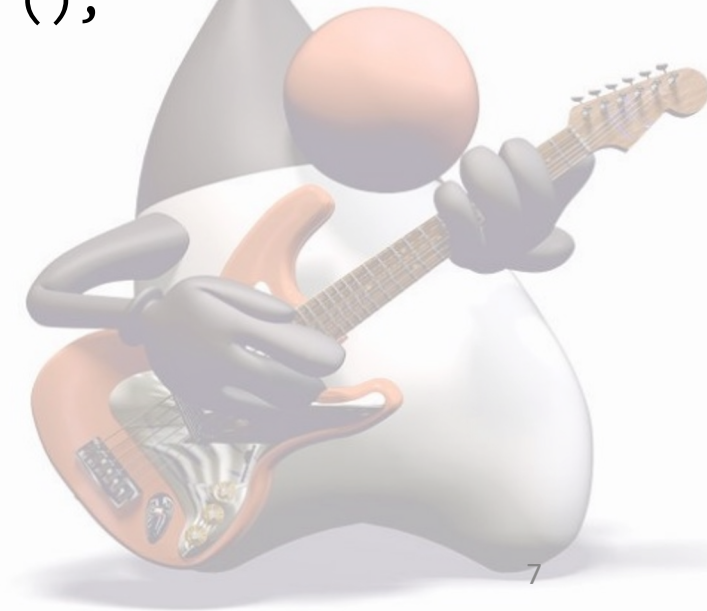
- `Customer[] customers = new Customer[5];`
- Nachdem das Array instanziiert wurde sind alle Elemente null

| |
|------|
| null |
| null |
| null |
| null |
| null |



Initialisieren eines Arrays von Referenzdatentypen

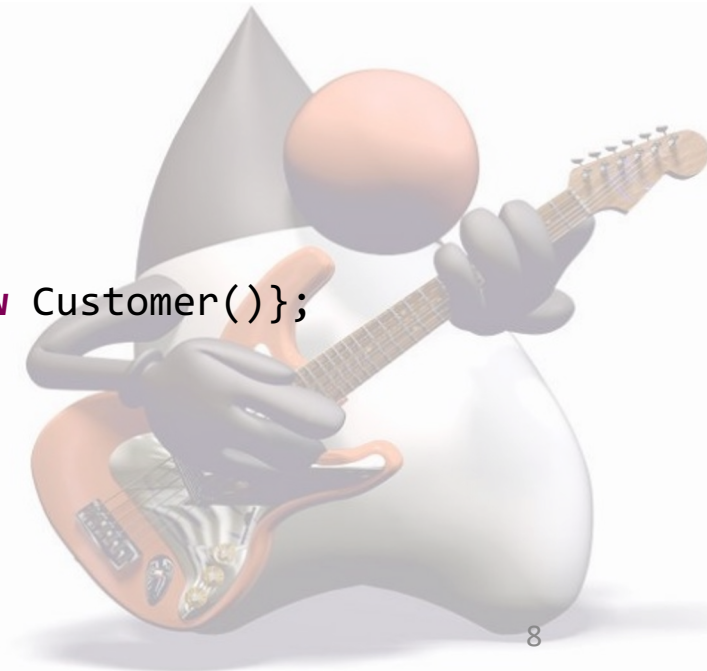
```
public static void main(String[] args) {  
    Customer[] customers = new Customer[5];  
  
    for (int i = 0; i < customers.length; i++) {  
        customers[i] = new Customer();  
    }  
}
```



Andere Möglichkeiten ein Array zu erstellen

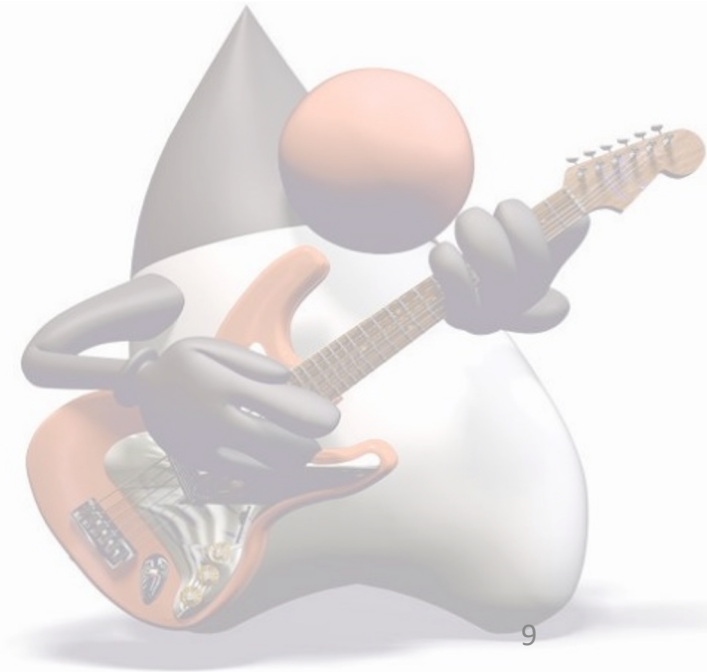
```
public static void main(String[] args) {  
    String[] months = { "January", "February", "March", "April", "May",  
                        "June", "Juli", "August", "September", "October", "November",  
                        "December" };  
  
    for (int i = 0; i < months.length; i++) {  
        System.out.println(months[i]);  
    }  
}
```

```
public static void main(String[] args) {  
    Customer[] customers = { new Customer(), new Customer(), new Customer()};  
  
    //...  
}
```



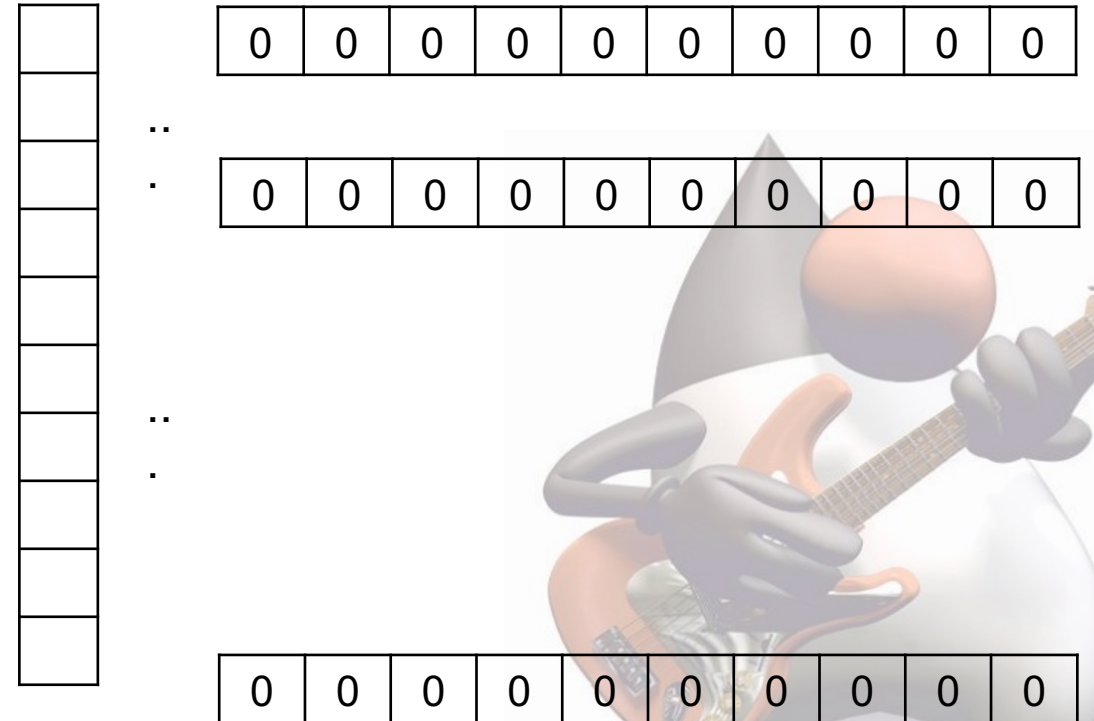
For-Each Loop, für Arrays

```
public static void main(String[] args) {  
    Customer[] customers = { new Customer(), new Customer(), new Customer()};  
  
    for (Customer c:customers){  
        c.setName("Anonymous");  
    }  
}
```



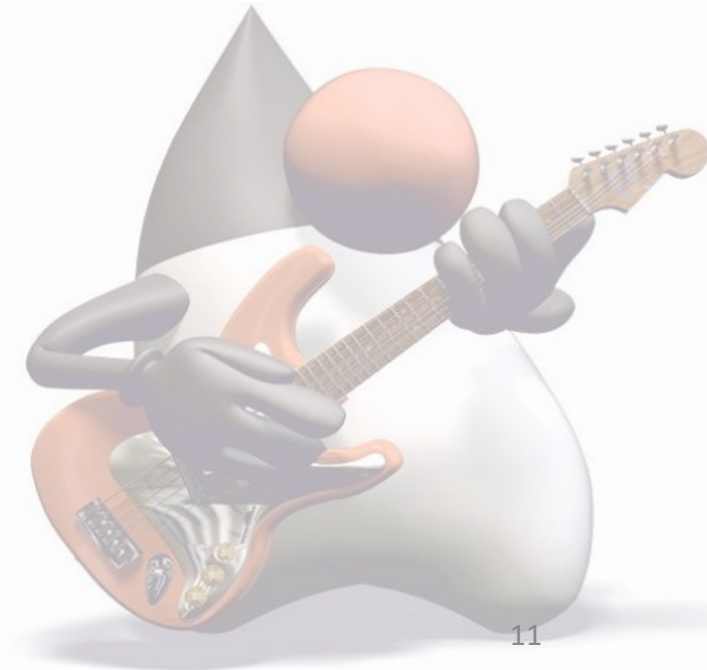
Multi-Dimensionale Arrays

```
public static void main(String[] args) {  
    int[][] table = new int[10][10];  
  
    for (int i = 0; i < table.length; i++) {  
        for (int j = 0; j < table[i].length; j++) {  
            table[i][j] = i * j;  
        }  
    }  
}
```



Multi-Dimensionale Arrays

```
public static void main(String[] args) {  
    int hypercube[][][][] = new int[5][5][5][5];  
    for (int i = 0; i < hypercube.length; i++) {  
        for (int j = 0; j < hypercube[i].length; j++) {  
            for (int k = 0; k < hypercube[i][j].length; k++) {  
                for (int l = 0; l < hypercube[i][j][k].length; l++) {  
                    hypercube[i][j][k][l] = i * j * k * l;  
                }  
            }  
        }  
    }  
}
```



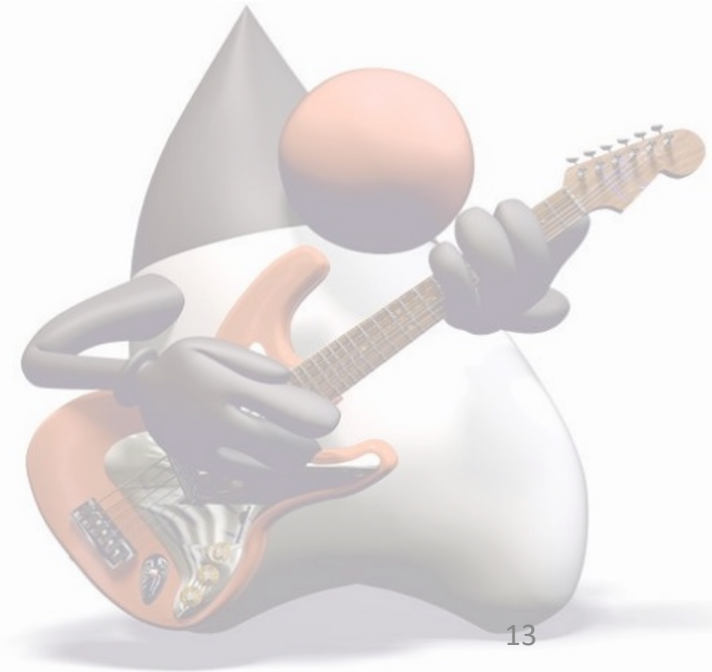
Resizing of Arrays

- Es ist nicht möglich die Größe eines Arrays zu ändern
- Wenn ein Array seine maximale Kapazität erreicht hat, muss man ein neues, größeres Array erzeugen und alle Elemente vom in das neue Array umkopieren
- Um Alle Elemente von einem Array ins andere zu übernehmen, verwendet man `System.arraycopy()`;



Resizing of Arrays

```
public static void main(String[] args) {  
    int[] numbers = new int[100];  
  
    int[] morenumbers = new int[200];  
  
    //...  
  
    System.arraycopy(numbers, 0, morenumbers, 0, numbers.length);  
  
}
```

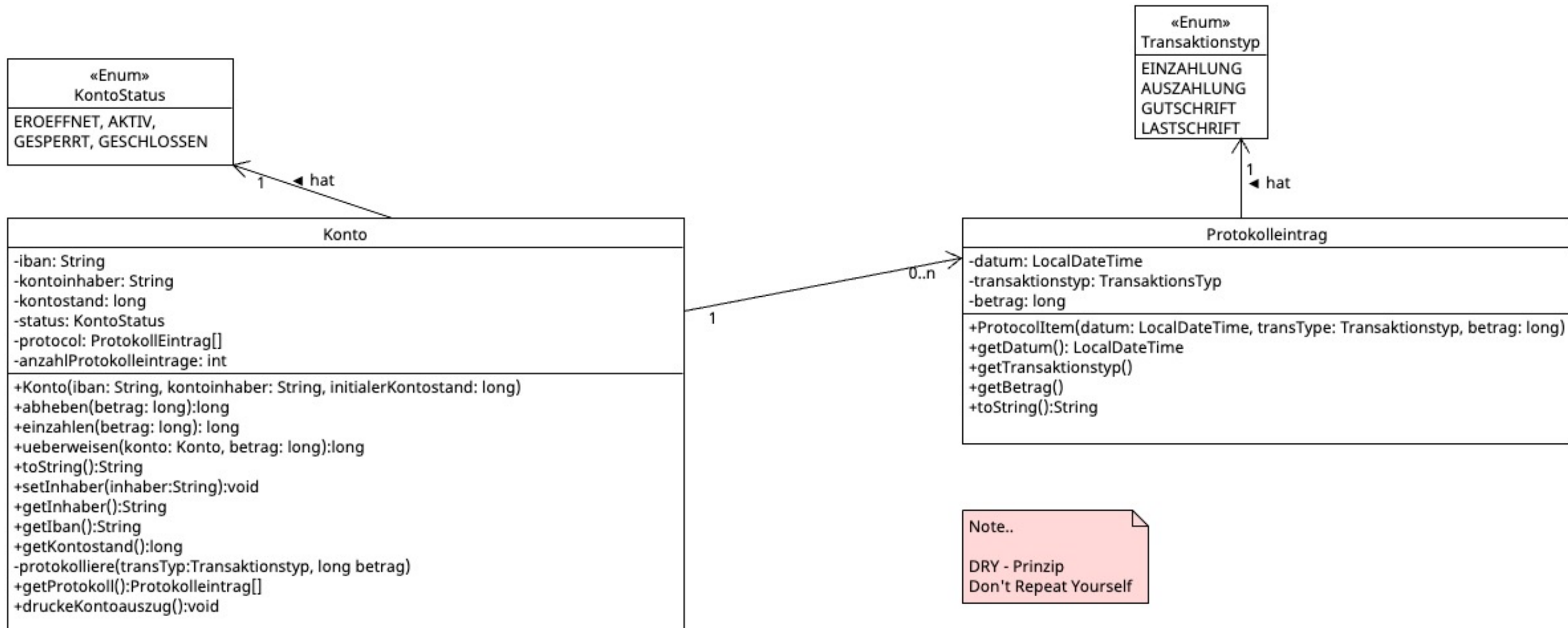


Übung

- Ändern sie ihre Account Klasse so, dass für sie ein Protokoll der Transaktionen mitschreibt
- Verwenden sie ein Array von „Protocol Items“. Das Protokoll ist eine selbst definierte Klasse mit Attributen
 - Date
 - Transaction Type (deposit, withdrawel, transfer)
 - Amount
- Implementieren sie in Ihrer Account Klasse eine getProtocol und im ProtocolItem eine toString Methode
- Jede Transaktion im Account soll im Protocol erfasst werden



UML Diagramm (Beispiel)



Übung

- Ergänzen sie in die Klasse Customer indem sie ein Array von Accounts halten kann
- Customer sollte zusätzliche Attribute wie Name, Geburtsdatum halten können
- Implementieren sie eine Methode Customer.openAccount and Customer.closeAccount

