Anmerkungen

- **Abgabe:** Die Abgabe erfolgt online auf ILIAS. Der Quellcode zu den Aufgaben 1-1 und 1-2 soll als `*.java` Datei und die Lösung von Aufgabe 1-3 soll als `*.pdf` Datei abgegeben werden. Andere Formate werden nicht akzeptiert.
- Quellcode-Dateien, welche wir nicht kompilieren können, werden nicht akzeptiert.
- Arbeit in Zweiergruppen: Geben Sie jeweils nur ein Exemplar der Lösung pro Gruppe ab. Geben Sie in der Quellcode-Datei die **Namen und Matrikelnummern** beider Gruppenmitglieder in den ersten beiden Zeilen als Kommentar an.
- Einzelarbeit: Geben Sie ebenfalls Ihren Namen und Ihre Matrikelnummer in der ersten Zeile der Quellcode-Datei als Kommentar an.

Aufgabe 6-1

Anmerkung In den folgenden Teilaufgaben werden die Matrizen A, B, C als 2-dimensionale Arrays dargestellt (z.B. int[][] A = new int[m][n]) . Des Weiteren ist es **nicht verlangt**, dass der Benutzer die Matrix eingibt.

Aufgabe 6-1a

Sei A eine $n \times n$ Matrix. Schreiben Sie einen Algorithmus, welcher A als Input erhält und A^T (die transponierte Matrix A) ausgibt. Bei dem Transponieren einer Matrix spiegelt man einen Matrixeintrag a_{ij} an der Diagonalen von A. Einfach gesagt, aus a_{ij} wird a_{ji}.

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \text{ und } A^T = \begin{pmatrix} a_{00} & a_{10} & a_{20} \\ a_{01} & a_{11} & a_{21} \\ a_{02} & a_{12} & a_{22} \end{pmatrix}$$

Ihr Algorithmus sollte alle möglichen Werte von n berücksichtigen. Schreiben Sie ausserdem eine Demonstration `Transpose.java` Ihrer Methode mit der folgenden Matrix A und geben Sie A^T aus.

$$\text{Input: } A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \text{ Output: } A^T = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

Aufgabe 6-1b

Sei A eine $n \times m$ Matrix und B eine $m \times l$ Matrix. Schreiben Sie einen Algorithmus der das Produkt AB berechnet. Ist $C = AB$, dann berechnet sich c_{ij} wie folgt

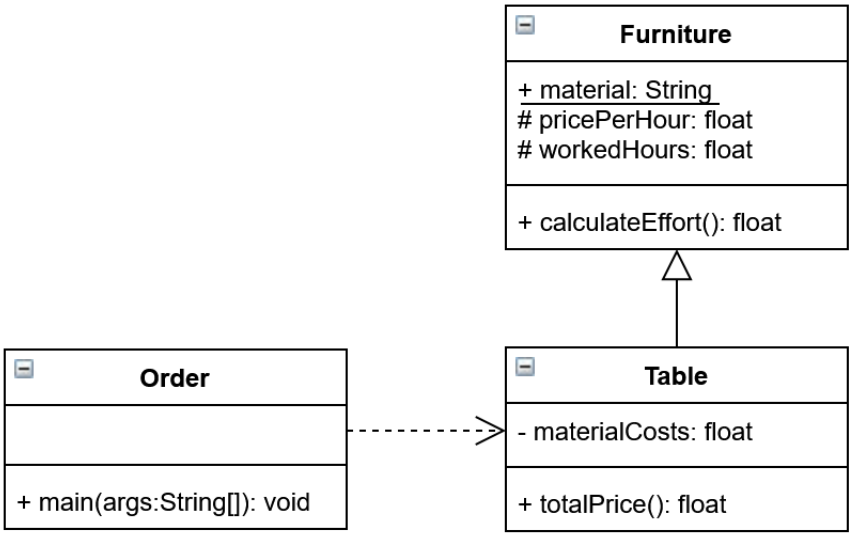
$$c_{ij} = \sum_{k=0}^{n-1} a_{ik} b_{kj}$$

Ihr Algorithmus sollte alle möglichen Werte von m, n und l berücksichtigen.Schreiben Sie ausserdem eine Demonstration `Product.java` Ihrer Methode mit den Matrizen A, B

$$\text{Input } A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}, B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \qquad \text{Output: } C = \begin{pmatrix} 9 & 12 & 15 \\ 19 & 26 & 33 \\ 29 & 40 & 51 \end{pmatrix}$$

Aufgabe 6-2

Sie sollen für eine Firma, welche Möbel herstellt, ein System entwickeln. Die Firma will sehen, ob sich dieses System bewährt, deshalb soll zunächst nur die Preisberechnung für die Tische implementiert werden. Falls die Firmenleitung zufrieden ist, sollen mehr Möbelstücke und mehr Funktionalitäten integriert werden. Implementieren Sie für dieses System `Furniture.java` und `Table.java` nach dem folgenden UML-Diagramm (`Order.java` ist auf Ilias verfügbar):



Anweisungen:

1. Die Klassen sollen nach diesem UML-Diagramm implementiert werden ohne zusätzliche Variablen oder Methoden zu verwenden.
2. Alle Möbel, welche die Firma herstellt, bestehen aus *weisser Eiche*.
3. Erstellen Sie einen passenden Konstruktor für `Furniture` und `Table`. Der Konstruktor von `Table` soll dabei den Konstruktor von `Furniture` verwenden.
4. Die Variable `pricePerHour` gibt die Kosten pro Stunde an für die Anfertigung des Möbelstückes. `workedHours` gibt die Anzahl Stunden an, welche nötig waren um das Möbelstück fertigzustellen. Die Methode `calculateEffort()` soll nun den Aufwand für die Herstellung eines Möbelstückes berechnen.
5. Da in der Aufwandsberechnung noch nicht der Materialpreis inbegriffen ist, soll in der Methode `totalPrice()` zunächst mithilfe der Methode aus `Furniture` den Preis für den Aufwand berechnet werden, dann der Materialpreis (`materialCosts`) dazu addiert werden.
6. Ist es in diesem Fall sinnvoll Inheritance zu verwenden? Begründen Sie Ihre Antwort.
7. Sind die Visibility Modifiers hier sinnvoll gewählt? Begründen Sie Ihre Antwort.
8. Was bedeutet Overloading? Wie könnte man Overloading in diesem Beispiel verwenden?

Hinweis: Verwenden Sie die `super` Referenz.

6-3

Welches ist die Ausgabe des folgenden Programms? Überlegen Sie sich dies, ohne das Program auszuführen.

```
1  public class A{
2      protected String bla;
3      public A(){ this.bla = "Hello from A"; }
4      public void bla(){ System.out.println(this.bla); }
5      public void foo(){ System.out.println("A.foo"); }
6      public void bar(){ this.foo(); }
7  }
8  public class B extends A{
9      public B(){
10         super();
11         this.bla = "Hello from B";
12     }
13 }
14 public class C extends B{
15     public C(){ super(); }
16     public void foo(){ System.out.println("C.foo"); }
17 }
18 public class Test{
19     public static void main(String [] args) {
20         new A().bla();
21         new A().foo();
22         new A().bar();
23         new B().bla();
24         new B().foo();
25         new B().bar();
26         new C().bla();
27         new C().foo();
28         new C().bar();
29     }
30 }
```