

Übung 4

UML

Beim Software Engineering sollte, so haben wir es in der Vorlesung gelernt, zunächst die Analyse stattfinden, bevor mit dem Design und der Programmierung begonnen wird. Hier fangen wir trotzdem einmal umgekehrt an...

1. Schreiben Sie ein JAVA-Programm namens **Multiplikation.java**, das mit Hilfe der Funktion **double readDouble(String prompt)**; aus der **Eingabe**-Klasse (s. **Eingabe.java**) zwei **double**-Zahlen von der Tastatur einliest, diese miteinander multipliziert, und das Ergebnis wieder am Bildschirm ausgibt. Tipp: Wenn Sie statt Aggregation eine Vererbung verwenden, wird das in den nächsten Aufgaben zu erstellende UML-Diagramm interessanter!

Ein vielleicht etwas größeres Beispiel als notwendig:

```
public class Multiplikation { // uses public class Eingabe
    private double d1, d2;

    public double getD1() { return d1; }

    public double getD2() { return d2; }

    public Multiplikation() {
        d1 = Eingabe.readDouble("Bitte erste double-Zahl eingeben: ");
        d2 = Eingabe.readDouble("Bitte zweite double-Zahl eingeben: ");
    }

    public double ergebnis() {
        return d1 * d2;
    }

    public static void main(String[] args) {
        Multiplikation m = new Multiplikation();
        System.out.println( m.getD1() + " * " + m.getD2() + " = "
            + m.ergebnis() );
    }
}
```

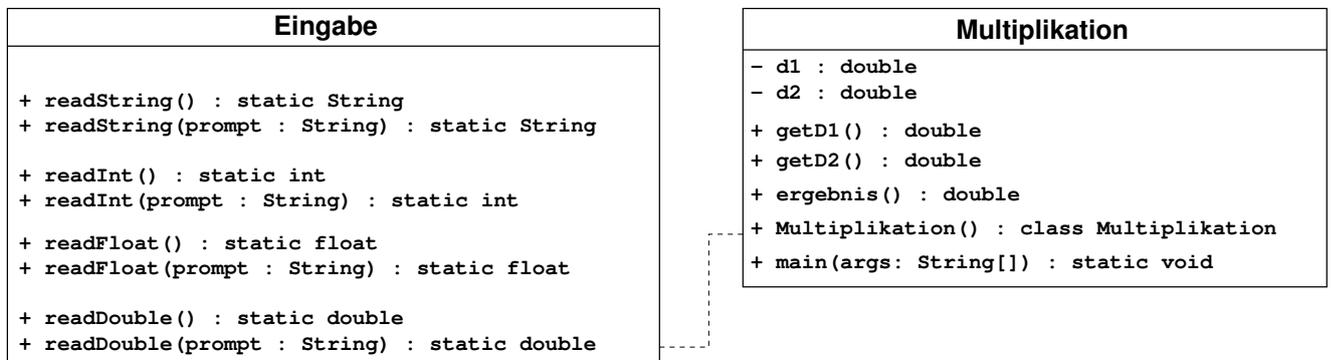
2. Beginnen Sie, was z.B. bei einem „Real Life“ Projekt der Fall sein könnte, eine technische Dokumentation für das Multiplikations-Programm, für die Sie zunächst nur die Beziehung der in Ihrem Programm explizit verwendeten Klassen und Objekte graphisch darstellen.

- (a) Welche(n) UML 2.0 Diagrammtyp(en) würden Sie verwenden, und warum?

Im obigen Beispiel ist, konform mit dem Aufgabentext, weder Vererbung noch Aggregation/Komposition verwendet worden. Daher würde sich evtl. das Verteilungsdiagramm anbieten, um den Datenfluss (Eingabedaten, Ausgabedatum) darzustellen. Nun sollen aber dessen ungeachtet die Klassen bzw. die verwendeten Objekte dargestellt werden, daher hier entsprechend eine Darstellung als Klassendiagramm.

NB: Man könnte den Aufruf der statischen Methode `readDouble(String)` aus der Klasse `Eingabe` innerhalb von `Multiplikation` (gestrichelte Linie im Diagramm unten) auch als Aggregation auffassen, dann kommen entsprechende Pfeile hinzu.

- (b) Stellen Sie die Beziehung zwischen den von Ihnen verwendeten Klassen nun in der von Ihnen gewählten Diagrammform dar.

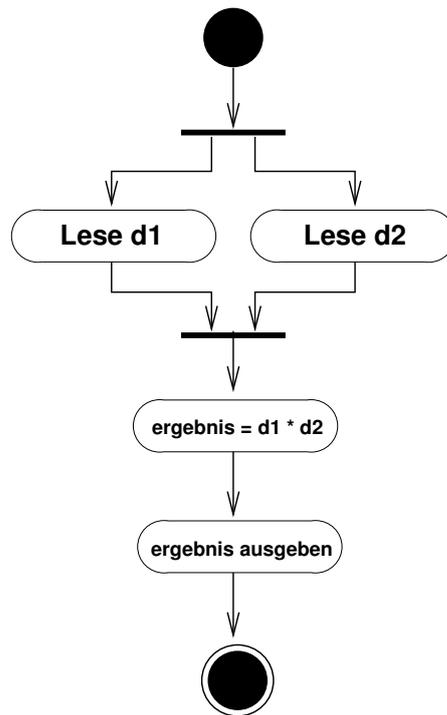


3. Der Ablauf des sehr einfachen Multiplikationsprogramms aus der ersten Aufgabe dieser Übung soll nun ebenfalls in Form einer geeigneten Skizze dokumentiert werden.

- (a) Welche(n) UML 2.0 Diagrammtyp(en) würden Sie verwenden, und warum?

Um einen Ablauf zu dokumentieren, bieten sich die Verhaltensdiagrammtypen an (Aktivitätsdiagramm, Anwendungsfalldiagramm, Interaktionsubersichtsdiagramm, Kommunikationsdiagramm, Sequenzdiagramm, Zeitverlaufsdiagramm, Zustandsdiagramm) an. In diesem Fall würde voraussichtlich das Aktivitätsdiagramm am meisten Sinn machen.

- (b) Stellen Sie den Ablauf des Programms auf die gewählte Art und Weise graphisch dar.



4. Stellen Sie den **Aufbau einer TCP-Internetverbindung**, z.B. beim Aufruf einer Webseite, als **UML-Sequenzdiagramm** dar.

