

Wintersemester 2024/25  
Prof. Dr. J. Rexilius

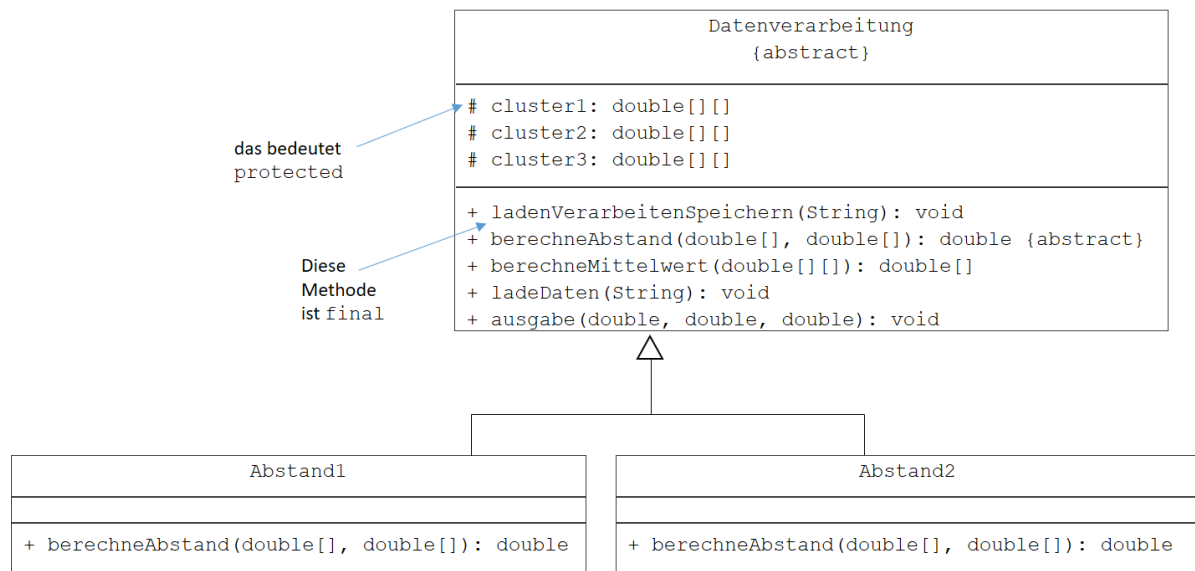
Abgabetermin: 16.01.2025, 08:00

## Aufgabe 12 (9 Punkte)

Beachten Sie die aktuelle Version zu den allgemeinen Abgabehinweisen (Praktikumsordner im ILIAS).

### (a) Lilien (6 Punkte)

Ein bekannter Datensatz ist der sogenannte Iris-flower-Datensatz – siehe Praktikumsordner zu diesem Aufgabenzettel. Der Datensatz besteht aus jeweils 50 Beobachtungen für drei Arten von Schwertlilien. Jede Beobachtung entspricht dabei einer Zeile in der Datei. Für jede Beobachtung wurden jeweils vier Merkmale erhoben. Ziel dieser Aufgabe ist es, den Mittelwert für die jeweilige Lilienart zu berechnen (also ein Vektor der Länge vier), und danach den Abstand zwischen jeweils zwei Mittelwerten als Ergebnis der Verarbeitung zu berechnen. Implementieren Sie dazu die Klassen und Methoden wie im Klassendiagramm unten angegeben.



Die Arbeitsschritte in der Methode `ladenVerarbeitenSpeichern` sind wie folgt:

1. Datensatz aus Datei laden (Name & Verzeichnis wird als String übergeben). Jede Lilienart wird in eine eigene Matrix geschrieben. Gehen Sie davon aus, dass die Anzahl der Datenpunkte und die Positionen der Inhalte vorab bekannt sind. Methode: `ladeDaten`
2. Mittelwert für jede Art über die zugehörigen Werte berechnen. Insgesamt also 3 Mittelwerte. Methoden: `berechneMittelwert`
3. Abstände zwischen den drei Mittelwerten berechnen. Methoden: `berechneAbstand`
4. Ausgabe der berechneten Abstände auf der Konsole. Methode: `ausgabe`

Wintersemester 2024/25  
Prof. Dr. J. Rexilius

Abgabetermin: 16.01.2025, 08:00

Die Abstandsberechnung ist in den Klassen `Abstand1` und `Abstand2` wie folgt definiert:

- Für die Berechnung in `Abstand1`:  
Abstand  $d(a,b)$  für zwei Vektoren  $a=(a_1,\dots,a_n)$ ,  $b=(b_1,\dots,b_n)$ :  $d(a,b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$
- Für die Berechnung in `Abstand2`:  
Abstand  $d(a,b)$  für zwei Vektoren  $a=(a_1,\dots,a_n)$ ,  $b=(b_1,\dots,b_n)$ :  $d(a,b) = \sum_{i=1}^n |a_i - b_i|$

Testen Sie Ihre Implementierung mit den beiden Klassen `Abstand1` und `Abstand2`. Die Ausgabe soll die unten gegebene Form haben (nutzen Sie die Methode `System.out.printf` für die Formatierung). Beispielausgabe für `Abstand1`:

```
Ergebnis für Abstand1:  
d(cluster1, cluster2) = 3,21  
d(cluster1, cluster3) = 4,75  
d(cluster2, cluster3) = 1,62
```

### (b) Adapter (3 Punkte)

Implementieren Sie eine Datenstruktur, in der Elemente und ihre Priorität gespeichert werden. Nutzen Sie das Entwurfsmuster Adapter und implementieren Sie eine Adapter-Klasse `PrioAdapter` für eine generische Datenstruktur. Die Datenstruktur verwendet intern eine `ArrayList` für die Elemente und eine für die zugehörigen Prioritäten. Entwerfen Sie für die generische Datenstruktur ein Interface mit folgenden Methoden:

- `boolean empty()`  
liefert `true`, wenn die Datenstruktur leer (=keine Elemente) ist, ansonsten `false`.
- `void push(T item, int p)`  
legt ein Element ab bestehen aus einem `item` mit Priorität `p`. Der Inhalt von `item` wird dabei so in die `ArrayList` eingefügt, dass alle anderen Werte in der Liste mit einem kleineren Index eine kleinere oder gleiche zugehörige Priorität haben.
- `T pop()`  
löscht das Element mit höchster Priorität und liefert es zurück. Wenn keine Elemente vorhanden sind wird eine Exception vom Typ `RuntimeException` ausgelöst.
- `T peek()`  
liefert das Element mit der höchsten Priorität. Wenn keine Elemente vorhanden sind wird eine Exception vom Typ `RuntimeException` ausgelöst.

Testen Sie Ihren Entwurf.