

## Praktikumsaufgabe 4

### Thema: Objektsammlungen: Array, (Set) und Hash

Die Aufgaben **A 4.1-A 4.3** müssen zum Praktikum bearbeitet sein. **A 4.4** wird abgenommen, wenn Sie es zum Termin schaffen, wird aber Bestandteil von Aufgabe 5 werden.

### Umfang der Bearbeitung

**A 4.1:** 7 von 11 Aufgaben, A 4.1.11 muss enthalten sein

**A 4.2:** vollständig.

**A 4.3:** vollständig. Dies ist eine Modifikation von A 3.2. Das Erzeugen von Objekten des Modells wird hier durch einen geschachtelten Hash ersetzt.

**A 4.4:** vollständig wird **A 5.1**.

## 4.1 1-dimensionale Arrays

1. Schreiben Sie die Methode `erzeuge_1D_inhalt_index_reverse(n)`, die ein 1D Array der Länge `n` erzeugt und die Werte mit den umgedrehten Indizes belegt. Die Methode gibt das Array als Ergebnis zurück. Bsp.: Ein Array der Länge 5 hat den Inhalt: `[4,3,2,1,0]`
2. Schreiben Sie die Methode `erzeuge_1D_inhalt_random_intervall(n, range)`, die ein 1D Array der Länge `n` erzeugt und die Werte mit Zufallswerten im Intervall `range` belegt. Die Methode gibt das Array als Ergebnis zurück.
3. Schreiben Sie die Methode `inverses?(ary1, ary2)`, die prüft, ob `ary2` das inverse Array zu `ary1` ist. Bsp.:

```
puts inverses?(ary, ary.reverse()) # true
puts inverses?(ary, [])           # false
puts inverses?(ary, ary)          # false wenn ary nicht symmetrisch ist
puts inverses?([], [])            # true
puts inverses?([1], [1])          # true
```

4. Schreiben Sie die Methode `tausche(ary, i, j)`, die die Elemente an Position `i` und `j` in `ary` vertauscht.
5. Schreiben Sie die Methode `schiebe_auf_position(ary, index)`, die beginnend mit dem Index `index`, das Element an Position `i` mit dem Element an Position `i-1` tauscht, wenn das Element an Position `i` kleiner als das Element an Position `i-1` ist. Die Methode wird beendet, wenn das Element an Position `i` größer oder gleich dem Element an Position `i-1` ist. Verwenden Sie die Methode aus 4'tens. Bsp.:

Für `ary = [4, 3, 2, 1, 0]` und `index = 3` wird die 1 mit der 2, dann mit der 3, dann mit der 4 getauscht.

```
[4, 3, 2, 1, 0] #Original
[4, 3, 1, 2, 0] # 1'ter Tausch
[4, 1, 3, 2, 0] # 2'ter Tausch
[1, 4, 3, 2, 0] # 3'ter Tausch
```

6. Schreiben Sie die Methode `sortiere_einfach(ary)`, die beginnend mit dem Index `i=1` bis zum Ende des Arrays alle Element auf ihre Position schiebt. Verwenden Sie dazu die Methode aus 5'tens. Das Sortier-Verfahren, das Sie hier implementiert haben, heißt InsertionSort. Bsp.:

Für `ary = [4, 3, 0, 1, 2]` wird zuerst die 3, dann die 0, dann die 1 und dann die 2 auf die korrekte Position geschoben.

```
i=1: [3, 4, 0, 1, 2] # schiebt 3 auf die korrekte Position
i=2: [0, 3, 4, 1, 2] # schiebt 0 auf die korrekte Position
i=3: [0, 1, 3, 4, 2] # schiebt 1 auf die korrekte Position
i=4: [0, 1, 2, 3, 4] # schiebt 2 auf die korrekte Position
```

7. Schreiben Sie die Methode `loesche_alle(ary, wert)`, die alle Vorkommen von `wert` in `ary` löscht. Die Methode soll destruktiv arbeiten. Das Verwenden von

`delete_if`, `replace` und `clear` ist nicht erlaubt. Lösen Sie die Aufgabe mit einer Zählschleife (einen geeigneten Iterator für Zahlen).

8. Schreiben Sie das Prädikat `alle_elemente_groesser?(ary, wert)`, das prüft, ob alle Elemente in `ary` größer als `wert` sind. Das Verwenden von `all?` ist nicht erlaubt.
9. Schreiben Sie die Methode `kopie_von(ary)`, die eine Kopie von `ary` Element-weise erzeugt und die Kopie als Ergebnis zurückgibt.
10. Schreiben Sie die Methode `kopie_von_plus(ary, plus)`, die in ein Array der Länge `ary.length + plus` die Elemente von `ary` kopiert und die zusätzlichen Elemente mit 0 belegt.
11. Schreiben Sie die Methode `addiere_ary1D(ary1, ary2)`, die 2 Arrays unterschiedlicher Länge „addiert“, in dem es die Elemente der beiden Arrays „addiert“. Die Länge des Ergebnisarrays ist die Länge des längsten Arrays. Die Elemente des längsten Arrays, die nicht im gemeinsamen Bereich liegen, sollen in das Ergebnisarray übertragen werden. Bsp.:

```
addiere_ary1D([0, 1, 2, 3, 4, 5, 6],
              [3, 2, 3, 2, 4, 2, 4, 4, 2, 4])
```

berechnet:

```
[3, 3, 5, 5, 8, 7, 10, 4, 2, 4]
addiere_ary1D([3, 2, 3, 2, 4, 2, 4, 4, 2, 4],
              [0, 1, 2, 3, 4, 5, 6])
```

berechnet ebenfalls:

```
[3, 3, 5, 5, 8, 7, 10, 4, 2, 4]
```

## 4.2 2D-Arrays

1. Schreiben Sie die Methode `erzeuge_matrix_random_in_range(n,m,range)`, die eine `nxm` Matrix erzeugt, deren Elemente zufällig aus dem Intervall `range` gewählt sind.
2. Schreiben Sie die Methode `erzeuge_unregelmässiges_random_in_range(n,min,max,range)`, die ein unregelmäßiges 2D Array erzeugt, dessen 1D Arrays eine zufällige Länge im Bereich `min` bis `max` haben und dessen Elemente zufällig aus dem Intervall `range` gewählt sind.
3. Schreiben Sie die Methode `transponierte?(m1,m2)`, die prüft, ob `m2` die transponierte Matrix zu `m1` ist. Prüfen Sie, bevor Sie die Elemente vergleichen, ob die Anzahl der Zeilen von `m1` gleich der Anzahl der Spalten von `m2` und die Anzahl der Spalten von `m1` gleich der Anzahl der Zeilen von `m2` sind. Für eine Transponierte von `m1` muss für alle `i, j` gelten, dass `m1[i][j]==m2[j][i]`. Sie dürfen voraussetzen, dass `m1` und `m2` Matrizen sind.
4. Schreiben Sie die Methode `addiere_ary2D(ary1, ary2)`, die zwei unregelmäßige 2D Arrays unterschiedlicher Länge miteinander „addiert“, in dem sie die enthaltenen 1D Arrays „addiert“ (siehe 4.1. letzter Aufgabenteil). Die Länge des Ergebnisarrays ist die Länge des längsten 2D Arrays. Die 1D Arrays des längsten 2D Arrays, die nicht im

gemeinsamen Bereich liegen, sollen Elementweise in das Ergebnisarray übertragen werden. Bsp.:

```
ary2D_1: [[4, 6], [7, 6, 1, 7, 1, 7, 1, 8, 5], [1, 2, 3, 6], [1, 8, 2, 1], [1, 1]]
```

```
ary2D_2: [[5, 4, 5], [9, 6, 5, 2], [6, 2, 3, 4, 7, 9, 8]]
```

dann ist das Ergebnis von `addiere_ary2D(ary2D_1,ary2D_2):`

```
[[9, 10, 5], [16, 12, 6, 9, 1, 7, 1, 8, 5], [7, 4, 6, 10, 7, 9, 8], [1, 8, 2, 1], [1, 1]]
```

### 4.3 Symboltabellen Dictionaries Hashes erzeugen

Modifizieren Sie die Klasse `KasseLeser` aus Aufgabe 3 derart, dass die Zeilen der Datei in einen geschachtelten Hash eingetragen werden.

Auf der ersten Ebene bilden die Bestellnummern auf Hashes von Positionsdaten ab.

Auf der zweiten Ebene bilden die Produktnamen auf ein 2-elementiges Array ab, dessen 1'tes Element die Anzahl und dessen 2'tes Element der Produktpreis ist.

Wenn der geschachtelte Hash mit der Methode `pp_nested_hash(a_hash)` (siehe `utils.rb`) ausgegeben wird, dann erhalten wir folgende Ausgabe (Auszug):

```
1=>
  Vitello tonnato=>[1, 15]
  Antipasto misto=>[1, 15]
  Riso col nero di seppie=>[1, 16]
  Ossobuco=>[1, 25]
  Panna cotta=>[1, 7]
  Aqua=>[1, 10]
  Te nero=>[2, 8]
  Spuma=>[1, 4]
  Coca=>[1, 3]
2=>
  Insalata frutti di mare=>[1, 15]
  Cozze al marinara=>[1, 16]
  Saltimboca=>[1, 17]
  Panna cotta=>[1, 7]
  Tirami su=>[1, 8]
  Spuma=>[2, 8]
  Cappuccino=>[2, 8]
  Espresso=>[2, 8]
3=>
  Antipasto misto=>[1, 15]
  Vitello tonnato=>[1, 15]
  Insalata frutti di mare=>[1, 15]
  Saltimboca=>[1, 17]
  Arrosto di manzo=>[2, 50]
  Tirami su=>[1, 8]
  Caffè freddo=>[1, 5]
  Marsala-Zabaione=>[1, 6]
  Cappuccino=>[2, 8]
  Aqua=>[2, 20]
  Coca=>[1, 3]
...
```

## 4.4 Symboltabellen Dictionaries Hashes auswerten

### Lösen Sie die Aufgaben wenn möglich mit einer geeigneten Methode von Enumerable!

Die Klasse `Auswerter` liest mit Hilfe des Lesers aus Aufgabe 4.3 den Inhalt der Datei `rechnungen.csv` in einen Hash und speichert diesen in die Instanz-Variable `@kasse`. Des Weiteren hat die Klasse eine Methode, um zu einer Bestellnummer den Hash mit den Produkten zu lesen. Sie sollen die Klasse um Methoden erweitern, die den Hash für die Kasse für verschiedene Aufgabenstellungen auswertet.

Implementieren Sie in der Klasse `Auswerter` die folgenden Methoden:

1. `in_flaches_array()`: Bildet den Hash auf ein flaches Array ab. Sie dürfen die Methoden `flatten` und `to_a` verwenden.
2. `alle_produkte()`: Berechnet die Menge aller Produkte aller Bestellungen.
3. `wert(produkt_hash)`: Berechnet für einen Produkt-Hash den Gesamtbestellwert.
4. `in_bestellung_mit_wert()`: Bildet den Hash auf ein Array mit 2-elementigen Arrays ab, dessen 1'tes Element die Bestellnummer und dessen 2'tes Element der Wert der Bestellung ist.
5. `gesamt_bestellwert()`: Berechnet den Gesamtbestellwert der Kasse.
6. `hoechste_bestellung()`: Berechnet die Bestellung mit dem größten Wert.
7. `anzahl_produkte(produkt_hash)`: Berechnet für einen Produkt-Hash die Anzahl der bestellten Produkte.
8. `umfangreichste_bestellung()`: Berechnet für die Kasse die Bestellung mit dem meisten bestellten Produkten.
9. `in_produkte_mit_gesamt_bestellwert()`: Bildet die Kasse in einen Hash ab, dessen Schlüssel die Produkte und dessen Werte der gesamte Bestellwert eines Produkts in der Kasse ist.
10. `in_produkte_mit_gesamt_bestellanzahl()`: Bildet die Kasse in einen Hash ab, dessen Schlüssel die Produkte und dessen Werte die gesamte Bestellanzahl eines Produkts in der Kasse ist.
11. `sortiere_produkte_nach_anzahl_name()`: Sortiert den Hash aus 4.4.10 zuerst nach Anzahl und dann nach Produktname.
12. `sortiere_produkte_nach_anzahl_name_absteigend()`: Wie 4.4.11 nur absteigend. Methode `reverse` von `Array` ist nicht erlaubt!

Wird das Skript `auswerter_szenario` mit einer korrekten Lösung ausgeführt, dann erzeugt das Skript folgende Ausgaben.

4.4.1 `in_flaches_array`-----

```
[ "1", "Vitello tonnato", 1, 15, "Antipasto misto", 1, 15, "Riso col nero di
seppie", 1, 16, "Ossobuco", 1, 25, "Panna cotta", 1, 7, "Aqua", 1, 10, "Te nero",
2, 8, "Spuma", 1, 4, "Coca", 1, 3, "2", "Insalata frutti di mare", 1, 15, "Cozze al
marinara", 1, 16, "Saltimboca", 1, 17, "Panna cotta", 1, 7, "Tirami su", 1, 8,
"Spuma", 2, 8, "Cappuccino", 2, 8, "Espresso", 2, 8, "3", "Antipasto misto", 1, 15,
"Vitello tonnato", 1, 15, "Insalata frutti di mare", 1, 15, "Saltimboca", 1, 17,
"Arrosto di manzo", 2, 50, "Tirami su", 1, 8, "Caffe freddo", 1, 5, "Marsala-
Zabaione", 1, 6, "Cappuccino", 2, 8, "Aqua", 2, 20, "Coca", 1, 3, "4", "Insalata
frutti di mare", 2, 30, "Insalata mista", 1, 8, "Cozze al marinara", 1, 16, "Tirami
su", 1, 8, "Coca", 3, 9, "Espresso", 1, 4, "Cappuccino", 1, 4, "Spuma", 1, 4, "5",
"Insalata mista", 1, 8, "Insalata frutti di mare", 1, 15, "Antipasto misto", 1, 15,
"Saltimboca", 1, 17, "Cozze al marinara", 1, 16, "Arrosto di manzo", 1, 25, "Tirami
su", 1, 8, "Marsala-Zabaione", 1, 6, "Panna cotta", 1, 7, "Cappuccino", 1, 4,
"Spuma", 1, 4, "Espresso", 1, 4, "Aqua", 3, 30, "6", "Insalata Caprese", 2, 20,
"Insalata frutti di mare", 1, 15, "Riso col nero di seppie", 1, 16, "Caffe freddo",
1, 5, "Marsala-Zabaione", 1, 6, "Espresso", 3, 12, "Te nero", 1, 4, "Coca", 1, 3,
"7", "Insalata frutti di mare", 1, 15, "Cappon magro", 1, 28, "Riso col nero di
seppie", 1, 16, "Tirami su", 1, 8, "Marsala-Zabaione", 1, 6, "Caffe freddo", 1, 5,
"Coca", 1, 3, "Te nero", 1, 4, "Spuma", 1, 4, "8", "Insalata frutti di mare", 1,
15, "Vitello tonnato", 2, 30, "Riso col nero di seppie", 1, 16, "Pizza Margerita",
1, 12, "Pizza Funghi&Salsiccia", 1, 15, "Tirami su", 2, 16, "Aqua", 3, 30,
"Espresso", 1, 4, "Te nero", 1, 4, "9", "Insalata frutti di mare", 1, 15, "Insalata
Caprese", 1, 10, "Ossobuco", 1, 25, "Gamberoni alla griglia", 1, 30, "Panna cotta",
1, 7, "Caffe freddo", 1, 5, "Espresso", 1, 4, "Spuma", 1, 4, "Te nero", 1, 4, "10",
"Insalata frutti di mare", 1, 15, "Antipasto misto", 1, 15, "Saltimboca", 1, 17,
"Pizza Margerita", 1, 12, "Caffe freddo", 1, 5, "Panna cotta", 1, 7, "Coca", 2, 6,
"Te nero", 1, 4, "Spuma", 1, 4]
```

4.4.2 `alle_produkte`-----

```
["Vitello tonnato", "Antipasto misto", "Riso col nero di seppie", "Ossobuco",
"Panna cotta", "Aqua", "Te nero", "Spuma", "Coca", "Insalata frutti di mare",
"Cozze al marinara", "Saltimboca", "Tirami su", "Cappuccino", "Espresso", "Arrosto
di manzo", "Caffe freddo", "Marsala-Zabaione", "Insalata mista", "Insalata
Caprese", "Cappon magro", "Pizza Margerita", "Pizza Funghi&Salsiccia", "Gamberoni
alla griglia"]
```

```
4.2.3 wert({'Insalata frutti di mare'=>[1, 15], 'Cozze al marinara'=>[1, 16],
'Saltimboca'=>[1, 17], 'Panna cotta'=>[1, 7], 'Tirami su'=>[1, 8], 'Spuma'=>[2, 8],
'Cappuccino'=>[2, 8], 'Espresso'=>[2, 8]})
```

111

4.2.4 `in_bestellung_mit_wert`-----

```
[["1", 111], ["2", 111], ["3", 240], ["4", 131], ["5", 219], ["6", 125], ["7", 89],
["8", 248], ["9", 104], ["10", 91]]
```

4.2.5 `gesamt_bestellwert`-----

1469

4.2.6 `hoechste_bestellung`-----

"8"

```
4.2.7 anzahl_produkte({"Insalata mista"=>[1, 8], "Insalata frutti di mare"=>[1, 15], "Antipasto misto"=>[1, 15], "Saltimboca"=>[1, 17], "Cozze al marinara"=>[1, 16], "Arrosto di manzo"=>[1, 25], "Tirami su"=>[1, 8], "Marsala-Zabaione"=>[1, 6], "Panna cotta"=>[1, 7], "Cappuccino"=>[1, 4], "Spuma"=>[1, 4], "Espresso"=>[1, 4], "Aqua"=>[3, 30]})
```

15

```
4.2.8 umfangreichste_bestellung-----
```

"5"

```
4.2.9 in_produkte_mit_gesamt_bestellwert-----
```

```
{Vitello tonnato=>90
Antipasto misto=>60
Riso col nero di seppie=>64
Ossobuco=>50
Panna cotta=>35
Aqua=>230
Te nero=>36
Spuma=>40
Coca=>51
Insalata frutti di mare=>180
Cozze al marinara=>48
Saltimboca=>68
Tirami su=>72
Cappuccino=>40
Espresso=>68
Arrosto di manzo=>125
Caffe freddo=>25
Marsala-Zabaione=>24
Insalata mista=>16
Insalata Caprese=>50
Cappon magro=>28
Pizza Margerita=>24
Pizza Funghi&Salsiccia=>15
Gamberoni alla griglia=>30}
```

```
4.2.10 in_produkte_mit_gesamt_bestellanzahl-----
```

```
{Vitello tonnato=>4
Antipasto misto=>4
Riso col nero di seppie=>4
Ossobuco=>2
Panna cotta=>5
Aqua=>9
Te nero=>7
Spuma=>8
Coca=>9
Insalata frutti di mare=>10
Cozze al marinara=>3
Saltimboca=>4
Tirami su=>7
Cappuccino=>6
Espresso=>9
Arrosto di manzo=>3
Caffe freddo=>5
Marsala-Zabaione=>4
Insalata mista=>2}
```

```
Insalata Caprese=>3
Cappon magro=>1
Pizza Margerita=>2
Pizza Funghi&Salsiccia=>1
Gamberoni alla griglia=>1}
```

#### 4.2.11 sortiere\_produkte\_nach\_anzahl\_name-----

```
{Cappon magro=>1
Gamberoni alla griglia=>1
Pizza Funghi&Salsiccia=>1
Insalata mista=>2
Ossobuco=>2
Pizza Margerita=>2
Arrosto di manzo=>3
Cozze al marinara=>3
Insalata Caprese=>3
Antipasto misto=>4
Marsala-Zabaione=>4
Riso col nero di seppie=>4
Saltimboca=>4
Vitello tonnato=>4
Caffe freddo=>5
Panna cotta=>5
Cappuccino=>6
Te nero=>7
Tirami su=>7
Spuma=>8
Aqua=>9
Coca=>9
Espresso=>9
Insalata frutti di mare=>10}
```

#### 4.2.12 sortiere\_produkte\_nach\_anzahl\_name\_absteigend-----

```
{Insalata frutti di mare=>10
Espresso=>9
Coca=>9
Aqua=>9
Spuma=>8
Tirami su=>7
Te nero=>7
Cappuccino=>6
Panna cotta=>5
Caffe freddo=>5
Vitello tonnato=>4
Saltimboca=>4
Riso col nero di seppie=>4
Marsala-Zabaione=>4
Antipasto misto=>4
Insalata Caprese=>3
Cozze al marinara=>3
Arrosto di manzo=>3
Pizza Margerita=>2
Ossobuco=>2
Insalata mista=>2
Pizza Funghi&Salsiccia=>1
Gamberoni alla griglia=>1
Cappon magro=>1}
```