Francisco Litvay – k01648066 – SKZ 526

Einführung in die Softwareentwicklung II 367.028 Gruppe 1 – Hausübung 4

Professor Ismail Khalil

13/04/2018

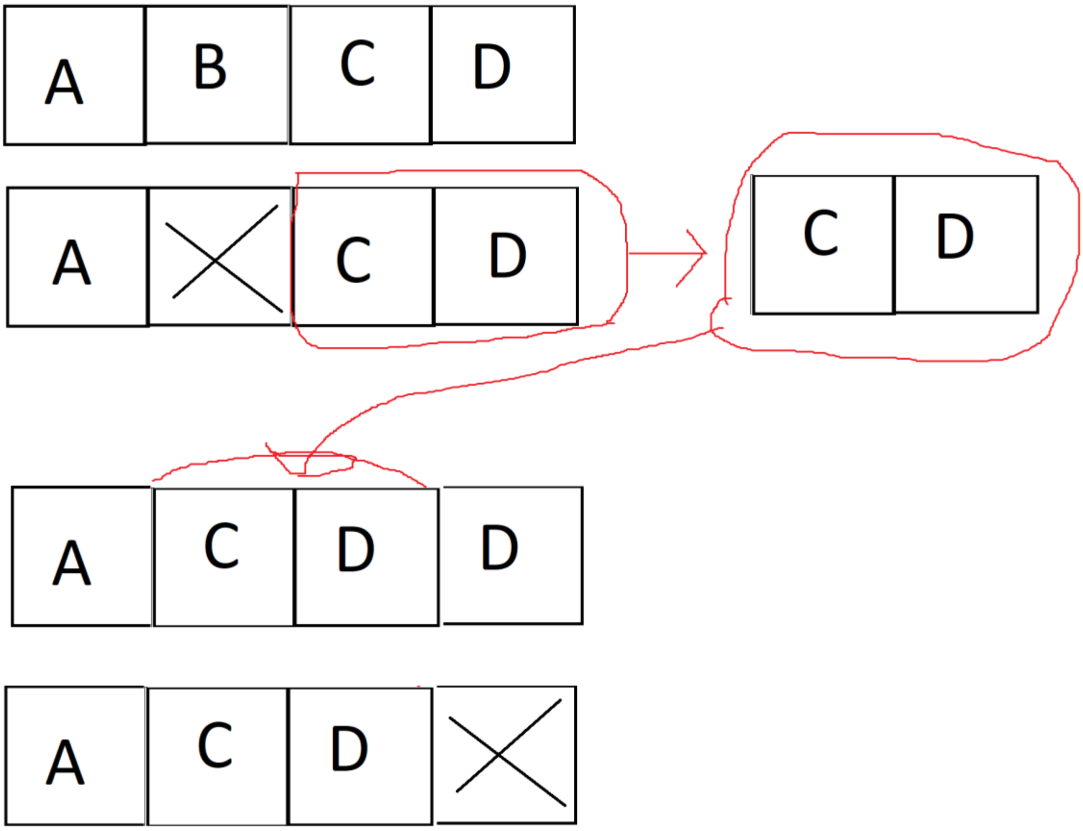
Dokumentation der Klassen in einem Textdokument:

Testfälle:

For this homework, 3 classes needed to be implemented: The arrayList, the EntryBasedMap and the Main. The ArrayList implements List and Iterable, has size and an array of generic type. The add first checks if there is space in the array, if not, it doubles it, then proceeds to adding the element on the next null spot. The get, contains, size and iterator are pretty standard, so i don't feel like there is a need to explain the methods. Remove however is a little bit more complicated: It will go through the array to look for the element that has to be removed. If found, it will remove the object, leaving a "null hole" in the middle of our list. This means that when we want to add some item, it would add it into this null hole, since it will start from the beginning and go to the end of the list, basically meaning it would change item order – an item would be inserted in the middle if a removal operation had happened before.

The solution for this issue is the following: We create a new array which encompasses the „hol" until the rest of the array. We then copy all elements after that hole to this new array, then transfer all these elements to the hole spot and beyond, effectively „moving" all items one field backwards. Then we delete the last item of the array to avoid duplicating the last item.

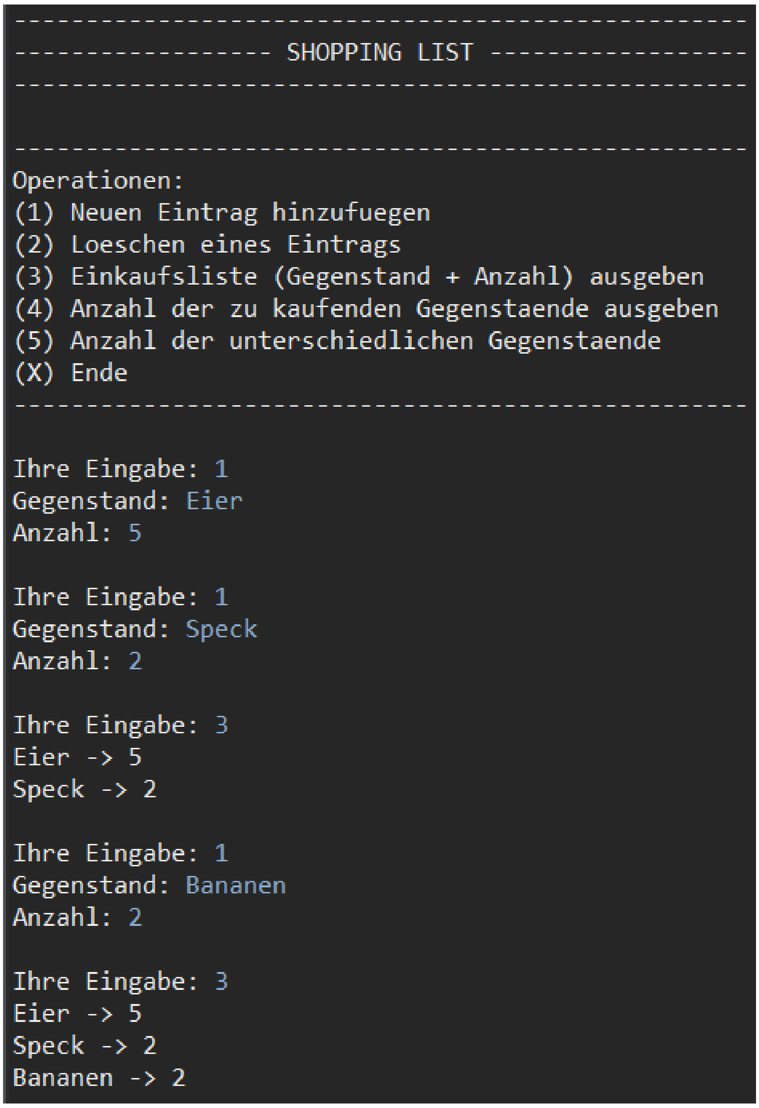
This simple scheme demonstrates the process:



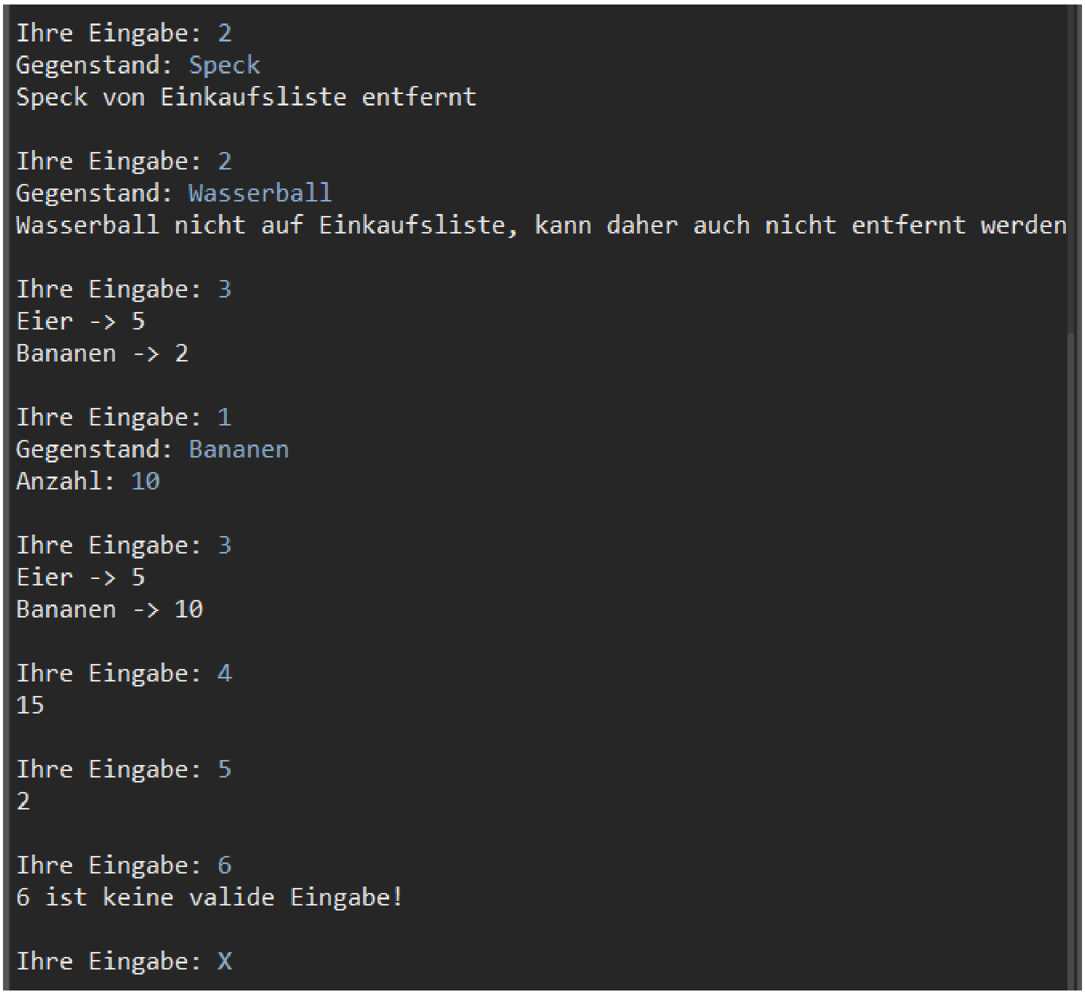
Now for the EntryBasedMap. The put will basically iterate through the items to see if the specified key already exists, if it is, it will update the value, otherwise will call the ArrayList add method. Get, contains, size and iterators are again trivial, the remove will basically call remove if it finds the key, otherwise just return false. Important here is the Entry inner class that implements the Tuple, allowing us to use all operations that needs to check keys and values.

Finally, the main. We create a EntryBasedMap. At 1, we put the item. At 2, we try to remove it, reacting accordingly if it succeeds or fails. 3 Iterates through all items, printing the key and the value. 4 Iterates through all items, summing their values. 5 Iterates through all items, adding 1 to the counter for every distinct item. Default for the switch is to say that the number is not permitted, and a X quits the program. That’s about it 😉

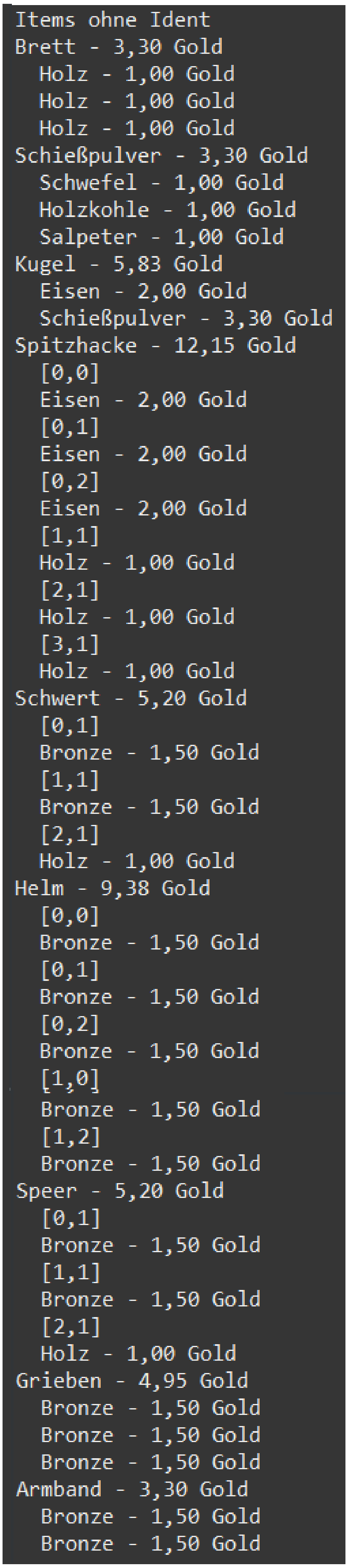
These Test demonstrations cover the example testcases:



```
---------------------------------------------------
------------------ SHOPPING LIST ------------------
---------------------------------------------------


---------------------------------------------------
Operationen:
(1) Neuen Eintrag hinzufuegen
(2) Loeschen eines Eintrags
(3) Einkaufsliste (Gegenstand + Anzahl) ausgeben
(4) Anzahl der zu kaufenden Gegenstaende ausgeben
(5) Anzahl der unterschiedlichen Gegenstaende
(X) Ende
---------------------------------------------------

Ihre Eingabe: 1
Gegenstand: Eier
Anzahl: 5

Ihre Eingabe: 1
Gegenstand: Speck
Anzahl: 2

Ihre Eingabe: 3
Eier -> 5
Speck -> 2

Ihre Eingabe: 1
Gegenstand: Bananen
Anzahl: 2

Ihre Eingabe: 3
Eier -> 5
Speck -> 2
Bananen -> 2
```



```
Ihre Eingabe: 2
Gegenstand: Speck
Speck von Einkaufsliste entfernt

Ihre Eingabe: 2
Gegenstand: Wasserball
Wasserball nicht auf Einkaufsliste, kann daher auch nicht entfernt werden

Ihre Eingabe: 3
Eier -> 5
Bananen -> 2

Ihre Eingabe: 1
Gegenstand: Bananen
Anzahl: 10

Ihre Eingabe: 3
Eier -> 5
Bananen -> 10

Ihre Eingabe: 4
15

Ihre Eingabe: 5
2

Ihre Eingabe: 6
6 ist keine valide Eingabe!

Ihre Eingabe: X
```



```
Items ohne Ident
Brett - 3,30 Gold
  Holz - 1,00 Gold
  Holz - 1,00 Gold
  Holz - 1,00 Gold
Schießpulver - 3,30 Gold
  Schwefel - 1,00 Gold
  Holzkohle - 1,00 Gold
  Salpeter - 1,00 Gold
Kugel - 5,83 Gold
  Eisen - 2,00 Gold
  Schießpulver - 3,30 Gold
Spitzhacke - 12,15 Gold
  [0,0]
  Eisen - 2,00 Gold
  [0,1]
  Eisen - 2,00 Gold
  [0,2]
  Eisen - 2,00 Gold
  [1,1]
  Holz - 1,00 Gold
  [2,1]
  Holz - 1,00 Gold
  [3,1]
  Holz - 1,00 Gold
Schwert - 5,20 Gold
  [0,1]
  Bronze - 1,50 Gold
  [1,1]
  Bronze - 1,50 Gold
  [2,1]
  Holz - 1,00 Gold
Helm - 9,38 Gold
  [0,0]
  Bronze - 1,50 Gold
  [0,1]
  Bronze - 1,50 Gold
  [0,2]
  Bronze - 1,50 Gold
  [1,0]
  Bronze - 1,50 Gold
  [1,2]
  Bronze - 1,50 Gold
Speer - 5,20 Gold
  [0,1]
  Bronze - 1,50 Gold
  [1,1]
  Bronze - 1,50 Gold
  [2,1]
  Holz - 1,00 Gold
Grieben - 4,95 Gold
  Bronze - 1,50 Gold
  Bronze - 1,50 Gold
  Bronze - 1,50 Gold
Armband - 3,30 Gold
  Bronze - 1,50 Gold
  Bronze - 1,50 Gold
```