

Programmieraufgabe “Lagerverwaltung”

Erstellen sie ein Programm zur modernen Verwaltung eines Lagers.

Allgemeines

Die gesamte Applikation soll in Scala programmiert werden.
Erste Schritte können sie dem offiziellen “Tour of Scala” unter <https://docs.scala-lang.org/tour/tour-of-scala.html> entnehmen.

Anforderungen

Die Applikation soll folgende Funktionen abdecken:

- Hinzufügen neuer Artikel
- Löschen von Artikeln
- Suchen von Artikeln
- Berechnen des Werts aller Artikel aktuell im Lager

1. Class “Item”

Ein Artikel in der Lagerverwaltungssoftware muss die folgenden Werte verwalten können:

```
var id: Int  
var name: String  
var value: Int
```

Definieren sie dazu ein geeignetes Format.

2. Trait “Printer”

Der Trait “Printer” dient zum Ausgeben von Informationen auf die Console.
Implementieren sie den Trait in der Datei “Printer.scala” so, dass er folgende Methode zur Verfügung stellt:

```
def printAction(actionName: String, name: String): Unit = { }
```

Der Trait soll folgende beispielhafte Ausgabe

PC gespeichert.

in der Console darstellen können. Definieren sie dazu ein geeignetes Format.

3. Class “StoreItem”

Implementieren sie eine Class “StoreItem” in einer Datei “StoreItem.scala”.
Die Klasse soll die zuvor implementierten Traits “Printer” und “Item” implementieren und damit die darin definierten Methoden und Variablen erben.

4. Trait “StockManagement”

Implementieren sie den Trait “StockManagement” in der Datei StockManagement.scala mit folgenden Methoden:

1. `def delete(id: Int): Array[StoreItem]`
2. `def search(name: String): Array[StoreItem]`
3. `def store(item: StoreItem): Array[StoreItem]`
4. `def sumUp(): Int`

Die Class “Database” wird die Funktionalität dieses Traits implementieren.

5. Class “Database”

Implementieren sie eine Class “Database” in der Datei “Database.scala” die eine private Variable `storedItems` definiert.

Die Variable soll vom Typ `Array` sein und mehrere items vom Type `StoreItem` aufnehmen können.

Hinweis:

Die Werte in der Variablen sind natürlich flüchtig und werden nur zur Laufzeit der Applikation gespeichert. In einer produktiven Applikation ist es natürlich notwendig die einzelnen Werte in einer Datenbank zu persistieren.

In der Class "Database" implementieren sie die im Trait "StockManagement" definierten Methoden:

1. Delete

Die Methode "delete" bekommt als Parameter eine `id` vom Typ `Int` übergeben. Suchen sie im Array in der Variablen `storedItems` nach dem `StoreItem` mit der entsprechenden `id` und löschen sie diese aus dem Array.

Rufen sie die Methode `printAction` des `StoreItems` auf um die Action auf der Console auszugeben.

Beispielausgabe:

```
> Laptop gelöscht
```

Wird der Artikel nicht gefunden, geben sie bitte folgende Fehlermeldung aus:

```
> Id 1 nicht gefunden
```

2. Search

Die Methode "search" bekommt als Parameter einen `name` vom Typ `String` übergeben. Suchen sie im Array in der Variablen `storedItems` nach den `StoreItems` mit dem entsprechenden `name` und geben die gefundenen items vom Type `StoreItem` zurück.

Rufen sie die Methode `printAction` des `StoreItems` auf um die Action auf der Console auszugeben.

Beispielausgabe:

```
> PC gefunden
```

Wird der Artikel nicht gefunden, geben sie bitte folgende Fehlermeldung aus:

```
> PC nicht gefunden
```

3. Store

Die Methode "store" bekommt als Parameter ein item vom Typ `StoreItem` übergeben.

Fügen sie das übergebene item in das Array `storedItems` und retournieren sie das neue Array als Ergebnis der Methode.

Rufen sie die Methode `printAction` des `StoreItems` auf um die Action auf der Console auszugeben.

Beispielausgabe:

```
> Laptop gespeichert
```

4. SumUp

Die Methode “sumUp” retourniert ein Ergebnis vom Typ `Int`. Summieren sie alle `values` der `StoreItems` in dem Array `storedItems` und retournieren das Ergebnis.

Applikation

Implementieren Sie die Hauptapplikation `MainApp` in der Datei `MainApp.scala` in der sie ein Object mit der geeigneten Methode als Haupteinstiegpunkt in das Programm implementieren. Dann instanziiieren sie zunächst die Datenbank:

```
val db = new Database()
```

Instanziiieren sie weiters vier `StoreItems`:

```
val item1 = new StoreItem(1, "Laptop", 1000)
val item2 = new StoreItem(2, "PC", 800)
val item3 = new StoreItem(3, "Smartphone", 1200)
val item4 = new StoreItem(4, "PC", 1400)
```

Speichern sie nun die instanziierten items in der Database Class.

Bei korrekter Umsetzung bekommen sie folgende Ausgabe in der Console angezeigt:

```
> Laptop gespeichert
> PC gespeichert
> Smartphone gespeichert
> PC gespeichert
```

Löschen sie nun das Item mit der id 1.

Bei korrekter Umsetzung bekommen sie folgende Ausgabe in der Console angezeigt:

```
> Laptop gelöscht
```

Suchen sie nun nach dem Namen “PC”.

Bei korrekter Umsetzung bekommen sie folgende Ausgabe in der Console angezeigt:

```
> PC gefunden
> PC gefunden
```

Abgabemodalitäten

Abgabetermin: **Mittwoch, 11.12.2019 12:00** auf der Online-Abgabeplattform.

Verwenden Sie bei ihrer Abgabe keine packages sondern definieren Sie alle Klassen und Objekte und die App im Default package.

Für eine positive Abgabe ist das lauffähige (Scala) Programm mit allen hier genannten Anforderungen rechtzeitig vor dem Abgabetermin auf der Online-Plattform abzugeben. Weitere Informationen dazu in den VU-Einheiten/Tutorien und auf Moodle.