Java-Grundlagenschulung
Tag 6

SHD Einzelhandelssoftware GmbH

Agenda

- Enums
- Aufgabe 1 - Enums benutzen
- JUnit - Was ist das?
- Aufgabe 2 - Herunterladen und konfigurieren einer Java-Lib
- Verwendung des Frameworks
- Aufgabe 3 - JUnit in der Praxis
- JUnit - Before und After
- Aufgabe 4 - Before und After
- Hausaufgabe

Enums

- Aufzählungstyp mit keyword ‚enum‘ statt ‚class‘
- Bildet eine fest definierte Anzahl von unterschiedlichen Werten ab (z.b. Wochentage, Monate, Status, Tierart, ..)
- Jeder Enum-Eintrag existiert genau einmal als Objekt (Singleton)
- Einer Enum-Klasse kann man beliebig viele Attribute mitgeben
- Jede Enum kann direkt ‚losbenutzt‘ werden
- Jeder Enum-Eintrag wird kommasepariert groß geschrieben

```
public enum Farbe
{
 private int r;
 private int g;
 private int b;

 BLAU(0,0,255),
 ROT(255,0,0),
 GRUEN(0,255,0);
 .
 .
 .
}
```

Aufgabe 1 – Enums benutzen

1. Erzeuge ein Enum Tierart, mit allen Tierarten die wir in den vergangen Lektionen benutzt haben.
2. Füge dem Enum die Felder ‚fressZunhame‘ und ‚schlachtGewicht‘ hinzu und lege entsprechende getter an. Passe dabei auch, den Konstruktor an.
3. Rufe in dem Futter-/bzw. –Schlachtautomaten die Entsprechende Tierart-Enum auf und hole dir aus diesen die entsprechenden Fress/Schlachtattribute raus um die willkürlichen Zahlen aus deiner bisherigen Implementierung zu ersetzen.
4. Erzeuge ein weiteres Enum ‚Geschlecht‘, nehme es als Attribut in die Klasse Stalltier auf, schreibe dieses mit der name() Methode in die CSV Datei und setze dies auch wieder beim auslesen.

JUnit - Was ist das?

- Testframework zum Testen von Java-Code
- Dient dem automatisierten Testen
- Ist nicht im JDK vorhanden und muss heruntergeladen werden.

Anlegen einer Teststruktur in IntelliJ


src


automaten


FutterAutomat


test


automaten


AutomatTest

Aufgabe 2

1. Lade die neueste JUnit Bibliothek aus dem Internet runter.

2. Binde diese in IntelliJ in den ProjektEinstellungen unter Libraries ein.

3. Lege unter Modules einen neuen Ordner an und markiere diesen als Test

Verwendung des Frameworks

- Testklasse erstellen, z.B. AutomatTest mit selber package Struktur
- ***@Test*** Annotation oberhalb der Testmethode
- Assert Klasse zum festlegen des Testergebnisses
- Ausführung der Testklasse mit Run Befehl

Aufgabe 3 - JUnit in der Praxis

1. Programmiere eine Testmethode für die Fütter-Methode in der Klasse AutomatTest.

2. Programmiere eine Testmethode für die Schlacht-Methode in der Klasse AutomatTest.

3. Erstelle eine Testklasse für die DAO Schicht.

4. Programmiere eine Testmethode für das Schreiben der CSV-Datei, indem du den Inhalt der Datei auf Richtigkeit prüfst.

JUnit - Before und After

- Jeweils eine eigene Methode
- @Before Annotation dient dem definieren von Vorgaben und wird vor jeder Test-Methode aufgerufen
- @After Annotation wird nach jeder Test-Methode aufgerufen und dient zum Aufräumen

Aufgabe 4 - Before und After

- Erstelle eine Before Methode um den Stall zu erstellen, damit anschließend eine Testmethode diesen verwenden kann.
- Erstelle eine After Methode in der DAO-Schicht Testklasse, um die CSV Datei nach jedem Testdurchlauf zu löschen.

Hausaufgabe

Das Testframework JUnit noch einmal zuhause Anwenden:

1. Füge dem Stalltier das Attribut Geschlecht als enum hinzu.
   (Informiere dich im Internet über enums in Java)

2. Erstelle einen Zuchtautomat. Dieser soll prüfen ob von einer Tierart zwei unterschiedliche Geschlechter vorhanden sind.
   Falls ja, füge dem Stall ein neues Tier der gleichen Tierart hinzu.

3. Teste den Zuchtautomat