



Abgabe 2 Prüfungsvorleistung

Zoo Besichtigung

下面我们来尝试怎么来更新我们第一次的答案，我们主要之研究这个关于动物园的答案的一部分，使用一个新的角度来实现。所以你们可以使用你们第一次的答案，在这个基础上部分修改。

Im Folgenden betrachten wir erneut den Zoo aus Abgabe 1. Es werden allerdings nur Teile des Zoos betrachtet und es sind neue Aspekte zu Implementieren. Ihnen ist es daher freigestellt, Ihre Lösung oder Teile Ihrer Lösung aus Abgabe 1 erneut zu verwenden.

描述：你的任务是编写一个来动物园的游览者的游览路线。在游览结束之后（或者游览之中）游览者必须列出一张表，他们在什么笼子里面都看见了什么动物。

Beschreibung: Ihre Aufgabe ist den Rundgang eines Besuchers durch den Zoo zu programmieren. Am Ende/Während des Rundgangs soll der Besucher auflisten, welche Tiere er in welchem Gehege gesehen hat.

Im Zoo sind folgende Gehege mit folgenden Tieren (Namen dürfen Sie selbst vergeben):

- Vogelgehege 1 动物园里有如下的动物和如下的笼子，你们要给笼子和动物自己取名：
 - 3 Papageien · 鸟笼1
 3 鸚鵡
- Vogelgehege 2 · 鸟笼2
 - 2 Pinguine 2企鵝
- Wüste · 爬虫类
 - 17 Skorpione 17 蝎子
- Fluss · 河流类
 - 1 Krokodil 1 鳄鱼
 - 2 Flusspferde 2 河马
- Erde · 地面类
 - 212 Mülle 212 鼯鼠

Lesen Sie bitte alle Aufgabenteile und beachten Sie, dass die Aufgabenteile voneinander abhängen. 仔细阅读所有的作业要求，他们是相互相关的。

Aufgabe 1: Implementieren Sie eine einfach verkettete Liste. Nennen Sie die Implementierung (Klasse) `MyLinkedList`. Mittels einer Instanz von `MyLinkedList` speichern Sie alle Gehege. Verwenden Sie weitere Instanzen von `MyLinkedList` um die Tiere je Gehege zu speichern. Um den Rundgang des Besuchers zu realisieren, Implementieren Sie einen `MyListIterator`, der das Interface `Iterator` (<https://docs.oracle.com/javase/8/docs/api/java/util/Iterator.html>) erfüllt. Verwenden Sie den `MyListIterator` als `Iterator` um alle Gehege zu besuchen, geben Sie in jedem Gehege aus, welche Tiere Sie sehen. Verwenden Sie hierzu ebenfalls einen `Iterator`. `MyLinkedList` und der `MyListIterator` sind selbst zu implementieren!

作业1：实现一个链表 `LinkedList`。这个链表的名字叫做 `MyLinkedList`。使用一个链表的实体，来存储所有的笼子（Gehege）。使用另一个链表的实体，来存储每个笼子里住的动物（Tier je Gehege）。为了实现游览者的游览列表，你需要创建一个 `MyListIterator`，他的接口（Interface）要符合给出链接的要求（这个链接就是普通的JDK-API里面给出的页面，迭代器的页面）。使用这个 `MyListIterator` 作为迭代器来浏览所有的笼子。最后给出每一个笼子以及笼子里的动物。（动物和笼子都要使用迭代器）`MyLinkedList` 和 `MyListIterator` 要自己写（就是不许倒用 `util` 包）。

最后输出的格式是：笼子类型，笼子名称，动物类型，动物名称。（链表套链表）

Aufgabe 2: Implementieren Sie einen **Binärbaum** in dem alle Gehege gespeichert werden. Der Baum muss **nicht sortiert** sein aber weitgehend balanciert sein (Inbalance ist nur in der tiefsten Ebene erlaubt). Realisieren Sie einen **MyTreeIterator**, der das **Interface Iterator** (<https://docs.oracle.com/javase/8/docs/api/java/util/Iterator.html>) erfüllt. Verwenden Sie die realisierten Bäume zum Speichern der Gehege und Tiere je Gehege (analog zu Aufgabe 1). **Tauchen** Sie hierzu die Datenstruktur Ihrer Implementierung aus Aufgabe 1 aus. Halten Sie die Änderungen so gering wie möglich. Der Baum ist selbst zu implementieren!

作业2：实现一个二叉树Binärbaum, 来存储所有的笼子（Gehege）。这个二叉树不需要是有序的，但是一定要平衡的。（这里老师推荐用AVL树）你需要创建一个MyTreeIterator，他的Interface（接口）要符合给出链接的要求（这个链接就是普通的JDK-API里面给出的页面，迭代器的页面）。使用这个MyListIterator作为迭代器来浏览所有的笼子。最后给出每一个笼子以及笼子里的动物。（动物和笼子都要使用迭代器），要给出和第一题一样的结果。二叉树要自己写（就是不许倒util包）。

最后输出的格式是：笼子类型，笼子名称，动物类型，动物名称。（链表套链表）

- Ausgabe des Programms als .txt oder PDF
- Liste der Gruppenmitglieder inklusive Matrikelnummern

Abgabe bis spätestens 28. Mai 23:00 MEZ!

Anmerkung: Programme, die lediglich die Textausgaben erzeugen, aber keine Objekte anlegen und verknüpfen, werden mit nicht bestanden bewertet.