

Praktikum zum Programmierkurs (WI)

Aufgabe 2 (Erste Basisfunktionen)

Hinweis zu Bonuspunkten: Zur Vergabe der Bonuspunkte werden wir in regelmäßigen Abständen den von Ihnen produzierten Quellcode begutachten. Informationen darüber, wann wir welche Praktikumsaufgaben begutachten, wie viele Bonuspunkte es gibt und wie Sie Ihren Quellcodestand einreichen, werden rechtzeitig bekanntgegeben.

Hinweis zum entstehenden Code: Verwenden Sie für den entstehenden Code das in Praktikum 1 in [GitLab](#) angelegte Projekt/Repository.

Fachliches Beispiel: Lernkarten-App

Unser fachliches Beispiel für das Praktikum ist eine App für *Lernkarten*. Diese App werden wir im Verlaufe des Semesters Stück für Stück aufbauen und um neue Funktionalitäten erweitern.

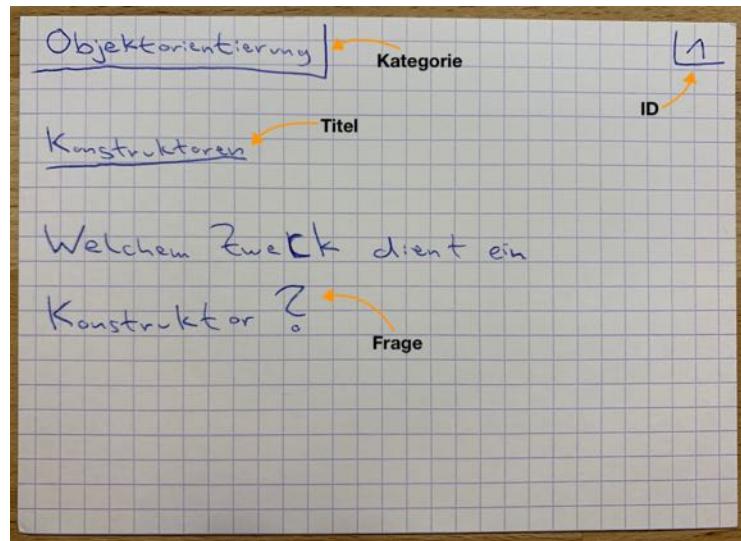
Lernkarten (englisch *flashcards*) unterstützen z.B. die Vorbereitung auf eine Prüfung und eignen sich besonders zur Verinnerlichung von Fakten (z.B. Antworten auf Wissensfragen, Vokabeln).

Das Prinzip ist einfach: Auf die Vorderseite der Lernkarte wird z.B. eine Frage notiert, auf die Rückseite eine entsprechende Antwort. Eine Sammlung solcher Lernkarten bildet eine *Lernkartei*. Beim Lernen zieht man eine zufällige Lernkarte aus der Lernkartei und betrachtet die Vorderseite. Man versucht dann, die Frage im Kopf zu beantworten. Danach betrachtet man die Rückseite und kontrolliert, ob man richtig lag. Dann packt man die Lernkarte wieder in die Kartei und zieht die nächste Karte. Grundidee ist es, dass man sich die richtigen Antworten durch ständige Wiederholung einprägt.

Weitere Varianten von Lernkarten, etwa mit mehreren Antworten (*multiple choice*), Stichworten etc. sind möglich. Die Lernkarten-App, die wir im Praktikum entwickeln, wird es ermöglichen, elektronische Lernkarten zu erstellen, zu verwalten und damit nach dem oben beschriebenen Prinzip zu lernen.

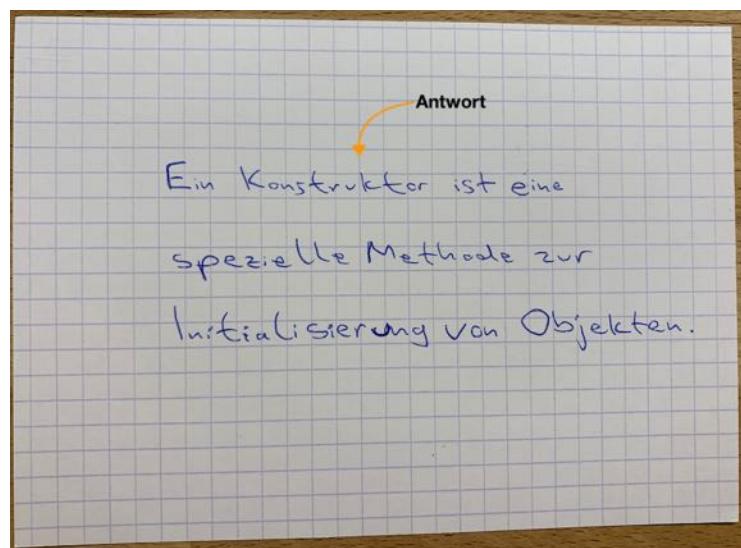
Hier ein Beispiel einer Lernkarte:

Vorderseite der Lernkarte:



Auf der Vorderseite der Karte findet sich eine Kategorie (es können mehrere Lernkarten zu einer Kategorie existieren), eine ID, ein Titel sowie die Frage.

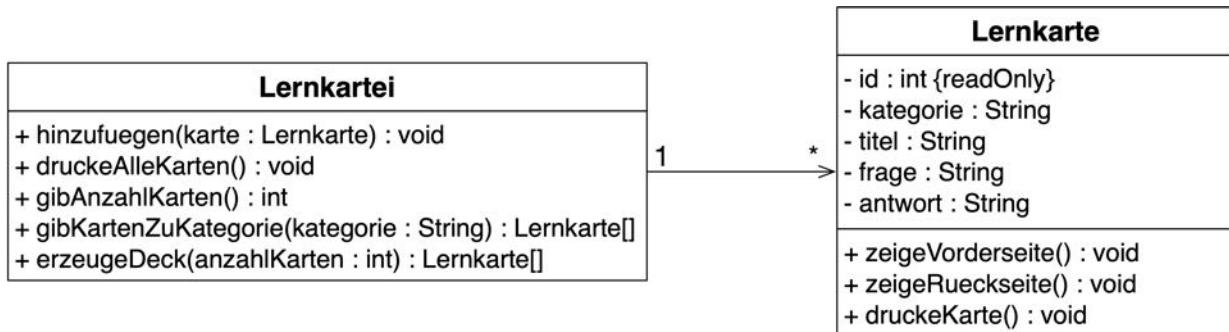
Rückseite der Lernkarte:



Auf der Rückseite steht die Antwort auf die Frage.

1. Lernkarte und Lernkartei implementieren

Implementieren Sie die Klassen `Lernkarte` und `Lernkartei` gemäß folgendem Klassendiagramm:



Bitte berücksichtigen Sie bei der Implementierung die folgenden Punkte:

- Legen Sie für den entstehenden Code im Package `pk` ein neues Package namens `lkarten` an.
- Lernkarten werden über die Klasse `Lernkarte` modelliert. Jede Lernkarte besitzt eine eindeutige ID (Attribut `id`), welche beim Anlegen eines Eintrages *automatisch vergeben* wird. Die IDs sollen fortlaufend sein, beginnend bei 1 (der zweite Eintrag bekommt dann die ID 2, der dritte Eintrag die ID 3, usw.).
- Implementieren Sie die folgenden Methoden in der Klasse `Lernkarte` :
 1. `zeigeVorderseite` : Gibt die Vorderseite der Lernkarte auf der Konsole aus.
Die Ausgabe soll in folgender Form erfolgen:

```
[ID, Kategorie] Titel:  
Frage
```

Beispiel:

```
[1, Objektorientierung] Konstruktoren:  
Welchem Zweck dient ein Konstruktor?
```

2. `zeigeRueckseite` : Gibt die Rückseite der Lernkarte (enthält lediglich die Antwort) auf der Konsole aus.

Beispiel:

```
Ein Konstruktor ist eine spezielle Methode zur  
Initialisierung von Objekten.
```

- druckeKarte : Gibt die gesamte Karte (also Vorderseite und Rückseite) der Karte auf der Konsole aus.

Beispiel:

```
[1, Objektorientierung] Konstruktoren:  
Welchem Zweck dient ein Konstruktor?  
Ein Konstruktor ist eine spezielle Methode zur  
Initialisierung von Objekten.
```

- Beachten Sie die Assoziation zwischen Lernkartei und Lernkarte . Realisieren Sie die Verwaltung der Lernkarten mit Hilfe eines Arrays. Beim Erzeugen einer Lernkartei -Instanz soll die Kapazität dieses Arrays (also die maximale Anzahl an Lernkarten) als Parameter mitgegeben werden.
- Implementieren Sie die folgenden Methoden in der Klasse Lernkartei :
 1. hinzufuegen : Fügt eine neue Lernkarte in der Lernkartei hinzu. Die neue Lernkarte wird dabei als Parameter übergeben. Falls das Array bereits voll ist, erfolgt eine Fehlermeldung auf der Konsole und die Lernkarte wird nicht hinzugefügt.
 2. druckeAlleKarten : Gibt alle in der Lernkartei enthaltenen Karten auf der Konsole aus.
 3. gibAnzahlKarten : Gibt zurück, wie viele Lernkarten aktuell in der Lernkartei enthalten sind.
 4. gibKartenZuKategorie : Gibt alle Lernkarten zu einer gegebenen Kategorie (Parameter kategorie) als Array zurück. Wurde zu einer Kategorie keine Lernkarte gefunden, so wird ein leeres Array zurückgegeben.
 5. erzeugeDeck :
 - Erzeugt eine zufällige Auswahl von Lernkarten (ein Deck) aus der Lernkartei und gibt diese als Array zurück.
 - Über den Parameter anzahlKarten kann bestimmt werden, wie viele Lernkarten das erzeugte Deck enthalten soll.
 - Entgegen der physikalischen Gesetze ist es in unserer Lernkarten-App erlaubt, dass ein und dieselbe Lernkarte mehrfach in einem Deck vorkommt - schließlich erhoffen wir uns durch Wiederholung einen besseren Lernerfolg.

 - Nutzen Sie zur Realisierung des Zufalls die Klasse java.util.Random aus der Java-Standardklassenbibliothek. Finden Sie durch Recherche in der [Java 17 API-Dokumentation](#) heraus, wie Sie die Klasse zu diesem Zweck einsetzen können.

Hinweis: Das Klassendiagramm ist unvollständig. Fügen Sie bei Bedarf weitere Methoden

hinzugefügt werden. z.B. private Hilfsmethoden oder solche Methoden, die Sie z.B. zum Zugriff auf den Zustand ihrer Objekte benötigen.

2. Java-Anwendung

Implementieren Sie (ebenfalls im Package `pk.lkarten`) eine ausführbare Java-Anwendung. Beim Ausführen der Java-Anwendung soll Folgendes geschehen:

1. Es soll eine Instanz der Klasse `Lernkartei` erzeugt werden.
2. Es sollen mindestens drei Lernkarten erzeugt und der Lernkartei hinzugefügt werden.
3. Rufen Sie *alle weiteren Methoden* der Klasse `Lernkartei` auf und geben Sie das Ergebnis jeweils auf der Konsole aus.

3. Commit und Push

1. Schreiben Sie den entstandenen Code per Commit in Ihrem lokalen Repository fest. Verwenden Sie als Commit-Message "Aufgabe 2: Erste Basisfunktionen".
2. Bringen Sie die Änderungen dann per Push auf den GitLab-Server. Kontrollieren Sie in GitLab, dass Ihre Änderungen angekommen sind.