

Lösungsvorschlag zur Sitzung am 25.01.2021 – Programmiersprachen: Objektorientierung und Datenstrukturen

Abgabe auf Ilias bis 01.02.2021, 10.00 Uhr

Aufgabe 1

Erläutern Sie sowohl das LIFO-, als auch das FIFO-Prinzip und referieren Sie jeweils eine Datenstruktur, die nach dem FIFO- und eine Datenstruktur, die nach dem LIFO-Prinzip arbeitet.

Lösung:

FIFO ist eine Abkürzung aus dem Englischen und steht für „First In, First Out“. Das Prinzip beschreibt eine Art der Speicherung, bei der zuerst eingespeicherte Daten auch zuerst wieder aus dem Speicher gelesen werden. Eine häufig verwendete Analogie, ist die der Warteschlange. Die Person, die als erstes angekommen ist, wird auch als erstes den Laden betreten.

LIFO ist ebenfalls eine Abkürzung und steht für „Last In, First Out“. Das Prinzip beschreibt die umgekehrte Form, des FIFO-Prinzips. Zuletzt gespeicherte Daten, werden zuerst ausgelesen. Man kann es sich wie einen vertikalen Stapel vorstellen, bei dem man zum Beispiel Bücher übereinanderlegt. Das zuletzt draufgelegte Buch, ist dann am einfachsten herunterzunehmen.

Die Queue ist eine häufig verwendete Datenstruktur in der Informatik, die nach dem FIFO-Prinzip arbeitet. Sie ermöglicht die Datenübergabe bei asynchronen Prozessen. Meistens handelt es sich dabei um langsamere Geräte, wie Drucker oder ähnliches.

Eine Datenstruktur, welche nach dem LIFO-Prinzip arbeitet, ist der Stack.

Aufgabe 2

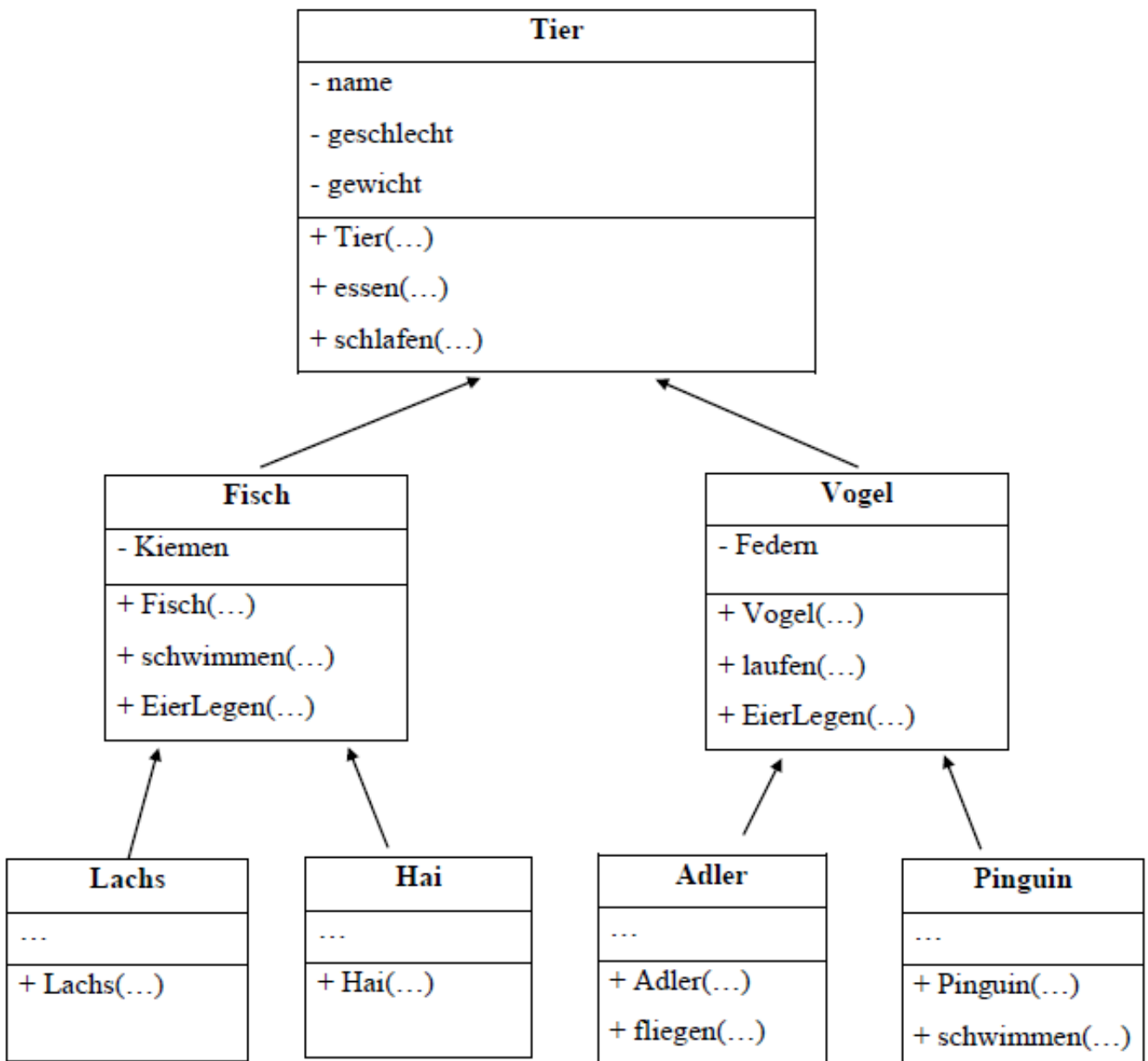
Erstellen Sie beispielhaft ein UML-Diagramm zum Thema "Tiere". Gehen Sie dazu explizit auf die Kapselung von Daten und Funktionen sowie die Vererbung ein.

Lösung:

Hinweis: Dies ist nur ein Lösungsvorschlag, welcher das Prinzip eines UML-Diagramms andeuten soll und somit nicht vollständig ist. Andere Lösungen sind möglich.

„+“ entspricht public (öffentliche Attribute/ Funktionen)

„-“ entspricht private (private Attribute/ Funktionen)



Aufgabe 3

Welche Vorteile bringen die Zugriffsmodifikatoren in Programmiersprachen, wie zum Beispiel C/C++ mit sich? Nennen Sie hierfür mindestens zwei Modifier und erläutern Sie ihre Eigenschaften.

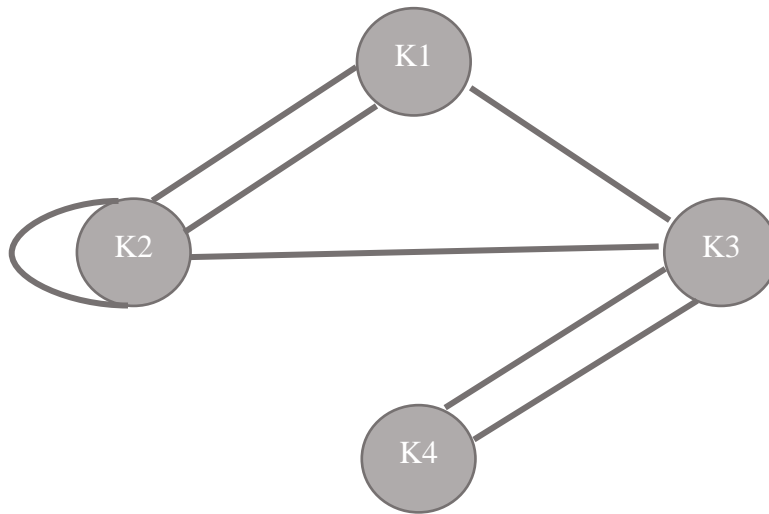
Lösung:

Zugriffsmodifikatoren definieren, welche Teile eines Programms Zugriff auf zum Beispiel eine Funktion oder Variable haben. Zugriffsmodifikatoren sorgen für eine bessere Wiederverwendbarkeit von Code. Wenn man zum Beispiel nicht möchte, dass man eine Variable von außerhalb der Klasse verändern kann, so lässt sich dies damit umsetzen. Außerdem kann man durch die Verwendung von Gettern und Settern zusätzliche Logik einbauen, die beispielsweise die Eingabe validieren kann, bevor sie gesetzt wird.

Public	Protected	Private	Friend
Alle Klassen haben Zugriff	Nur untergestellte Klassen besitzen Zugriff (Vererbung)	Klassenintern	Eine bestimmte weitere Klasse hat Zugriff

Aufgabe 4

Bestimmen Sie die Adjazenzmatrix für den folgenden Graphen:



Lösung:

	K1	K2	K3	K4
K1	0	2	1	0
K2	2	1	1	0
K3	1	1	0	2
K4	0	0	2	0