Programmieraufgabe 1 (1. KL, 1inf1)

# Grundanforderung

Erstelle ein Programm zur Verwaltung von Hausaufgaben mit Hilfe einer einfach verketteten Liste auf Basis des Kompositum-Entwurfsmusters. Die Hausaufgabe muss mindestens die Attribute *abgabeDatum[[1]](#footnote-1), inhalt, fach, prioritaet*[[2]](#footnote-2) enthalten und über eine sinnvolle GUI eingegeben werden können. Hierbei muss das MVC-Entwurfsmuster berücksichtigt werden.

# Grundfunktionalität

Eingabe von einer Hausaufgabe und Anzeige aller zu machenden Hausaufgaben für einen bestimmten Tag (->Datum) mit Angabe der Gesamtanzahl der zu machenden Hausaufgaben für diesen Tag.

# Weitere mögliche Funktionen

* Anzeige aller Hausaufgaben einer Tages nach Priorität sortiert.
* Anzeige aller nicht erledigter und erledigter Hausaufgaben
* Eingabe der Daten über Dropdown-Menüs etc. (Nur 1 Menü genügt nicht)
* Fenster zum Anpassen und nachträglichen Korrigieren der Daten

*Hinweise zur Arbeitszeit:*

Die Aufgabe darf nur während der normalen Unterrichtszeit bearbeitet werden. Lösungsansätze dürfen zwar zu Hause überlegt werden, programmiert wird aus Gründen der Fairness nur in der Schule während der normalen Unterrichtszeiten.

Folgende Zeitangabe liegen zugrunde:

1. Stunde: Handschriftlicher Klassentwurf

2. Stunde: Entwerfen der GUI mit dem JavaEditor

3./4. Stunde: Implementierung des Backends

5. Stunde: Implementierung der Kommunikation zwischen Front- und

Backend

6. Stunde: Fehlerkorrektur

7./8. Stunde Verbesserung

*Hinweis zur Benotung:*

Eine sehr gute Leistung geht grundsätzlich über die Grundfunktionalität hinaus (s.o.).

Bei der Benotung wird die Einhaltung der Java-Programmierrichtlinien berücksichtigt.

Es darf ausschließlich Unterrichtsstoff benutzt werden – keine ArrayLists oä.

*Hinweis zum Speichern von Daten:*

Speichern der Daten ist über das Interface Serializable möglich (s. Rückseite). Diese Funktionalität wird nicht mitbewertet.

# Klasse speicherbar machen

import java.io.Serializable;

public class Hausaufgabe implements Serializable{}

# Datei einlesen bzw. schreiben

import java.io.\*;

import java.util.\*;

public class Test {

public Test() {

Hausaufgabe h1 = new Hausaufgabe();

//schreiben

try {

//Die Datei wird ha\_data.txt heissen und im

//gleichen Ordner liegen

FileOutputStream fs = new FileOutputStream("ha\_data.txt");

ObjectOutputStream os = new ObjectOutputStream(fs);

os.writeObject(h1);

os.close();

} catch (IOException e) {

System.err.println(e.toString());

}

//einlesen

try {

//Die Datei heisst wieder ha\_data.txt und muss

//im gleichen Ordner liegen

FileInputStream fs = new FileInputStream("ha\_data.txt");

ObjectInputStream is = new ObjectInputStream(fs);

//Das eingelesene Objekt ist nicht als Objekt der Klasse

//Hausaufgabe erkennbar und wird daher von

//Hand dazu gemacht (-> casten)

h1 = (Hausaufgabe)is.readObject();

is.close();

} catch (ClassNotFoundException e) {

System.err.println(e.toString());

} catch (IOException e) {

//Diese Zeile kann auskommentiert werden,

//damit keine Fehlermeldung kommt,

//wenn keine Datei vorliegt.

System.err.println(e.toString());

}

}

}

Alle Klassen, die gespeichert werden, müssen als Serializable gekennzeichnet werden.

1. Ein String der Form yyyymmdd [↑](#footnote-ref-1)
2. 1 ist die höchste Priorität. 4 die niedrigste. [↑](#footnote-ref-2)