

Hinweis: Diese Lernnotiz enthält einen sehr sinnvollen Vorschlag um den Lehrstoff der 14. Woche der Veranstaltung zu erlernen. Er ist gegliedert in die generellen Ziele und die Arbeitsschritte. Es ist notwendig, dass Sie die in dieser Lernnotiz genannten Videos bis zum Ende der offiziellen Vorlesungszeit (Fr 10.12, 11:30) durchgearbeitet haben. Zu den Vorlesungszeiten besteht die Möglichkeit in Zoom Fragen zu stellen und weitergehende Themen zu Aufgaben oder auch dem Studium zu diskutieren.

<https://hs-osnabrueck.zoom.us/my/kleuker>

Denken Sie daran, dass ich auch über E-Mail erreichbar bin und Fragen beantworte.

Ziele VL 26

- Verständnis und Fähigkeit zur Strukturierung von Programmen auf Basis von Paketen
- Wissen, dass es sinnvolle Java-Programme ohne Paketstrukturen nicht gibt
- Wunsch generieren, sich in der veranstaltungsfreien Zeit intensiver mit weiteren Entwicklungsumgebungen, zumindest Eclipse, zu beschäftigen

Arbeitsschritte VL 26

- *Laden Sie sich die folgenden Videos zuerst herunter, wenn Sie die HS-Plattform nutzen und schauen Sie sich diese an. Es ist sinnvoll die Folien danach nochmals durchzugehen.*

Folien 655 – 673: Einführung von Paketen zur Strukturierung von Klassen

<http://kleuker.iui.hs-osnabrueck.de/Videos/Prog1/Prog1Pakete2.mp4> (22:35), auch <https://youtu.be/86qYLbaiWGo>

Ausführung von mit Paketen strukturierten Programmen

<http://kleuker.iui.hs-osnabrueck.de/Videos/Prog1/Prog1Pakete3.mp4> (4:24), auch <https://youtu.be/D48nZwM6bH4>

Übergang von BlueJ nach Eclipse

<http://kleuker.iui.hs-osnabrueck.de/Videos/Prog1/Prog1Entwicklungsumgebungen1.mp4> (38:42), auch <https://youtu.be/QOttfUTYmkE>

Folien 674 – 684: Ideen von Entwicklungswerkzeugen, am Beispiel von Eclipse

<http://kleuker.iui.hs-osnabrueck.de/Videos/Prog1/Prog1Entwicklungsumgebungen2.mp4> (23:03), auch <https://youtu.be/d5fCMTriu8Q>

- Bearbeiten Sie das Quiz unter http://kleuker.iui.hs-osnabrueck.de/quiz/prog26_11477.html und merken Sie sich die oben angegebenen Lösungsbuchstaben für das Praktikum.
- Prüfen Sie, ob Sie die angegebenen Lernziele erreicht haben.

Durch die Feiertage, entfällt eine Vorlesung, die damit auch nicht Teil der Klausur ist. Die Erarbeitung des für nachfolgende Veranstaltungen wichtigen Stoffes ist damit Teil Ihrer Veranstaltungsnachbereitung. Natürlich können Sie dann auch Fragen stellen.

Ziele VL 27

- Fähigkeit zur schnellen Generierung typischer Standardelemente einer Java-Klasse (Konstruktoren, get- und set-Methoden, hashCode und equals sowie toString)
- Grundsätzliches Verständnis wie aus Java bekannte Klassen in C++ umgesetzt werden können.

- Grundsätzliches Verständnis des Begriffs „Zeiger“ und dessen Äquivalent in Java.

Arbeitsschritte VL 27

- *Laden Sie folgendes Video zuerst herunter, wenn Sie die HS-Plattform nutzen und schauen Sie sich diese an.*

Ein erster Schritt von Java nach C++

<http://kleuker.iui.hs-osnabrueck.de/Videos/Prog1/Prog1JavaNachCpp.mp4> (85:14),
auch <https://youtu.be/p6NjupdrsZM>

Es gibt diesmal keine Folien, da der Beispielcode genutzt und damit experimentiert werden soll.

Anmerkung: Die Vorlesung zeigt, dass alle wichtigen OO-Ideen aus Java fast vollständig sich in C++ wiederfinden. Neu ist der Unterschied zwischen Objekten und Zeiger auf Objekten. Java arbeitet aus C++-Sicht ausschließlich mit „Zeigern“, die vereinfacht Referenzen genannt werden. In C++ sind die Begriffe Zeiger und Referenz klar zu trennen, meinen aber verwandte Themen. Genauer kann man zu einem Objekt immer nur eine Referenz bekommen, die damit nicht null sein kann. Eine Objektvariable mit Zeiger-Typ kann aber den bekannten null-Wert enthalten.

```
// new gibt die Referenz auf ein neues Objekt zurueck,  
// diese muss eine Variable mit passenden Zeigertyp  
// zugewiesen werden. (generell genau wie Java)  
Punkt* puz = new Punkt(1,2);  
puz->setX(42); //entspricht der Schreibweise puz.setX(42) in Java  
  
// Anders als in Java kann man auch direkt ein Objekt  
// erzeugen und einer Variable dieses Typs zuweisen.  
// Hierzu gibt es kein Java-aequivalent!!  
Punkt puo(3,41);  
puo.setX(-3); //diese Schreibweise werden sie seltenst in C++ sehen  
  
// man kann zu einem Objekt ueber den &-Operator eine  
// Referenz auf das Objekt erhalten.  
puz = &puo;  
  
// man kann durch Dereferenzieren mit *-Operator das  
// zugehoerige Objekt erhalten (sauberer mit null-Pruefung)  
puo = *puz;
```

- Bearbeiten Sie das Quiz unter http://kleuker.iui.hs-osnabrueck.de/quiz/prog27_11477.html und merken Sie sich die oben angegebenen Lösungsbuchstaben für das Praktikum.
- Prüfen Sie, ob Sie die angegebenen Lernziele erreicht haben.
- Bearbeiten Sie für sich selbst das Aufgabenblatt 14 in der vorlesungsfreien Zeit. Denken Sie daran, dass ich bei Fragen oder für Diskussionen von Lösungen meist kurzfristig erreichbar bin.