Fragen, Antworten und Kommentare zur aktuellen Vorlesung

Die Online-Befragung zur genutzten alternativen Veranstaltungsform und zur Lehrevaluation ist online. Bitte ausfüllen: https://forms.gle/8HLbXfVaQLjbSeo5A. Sie werden eventuell aufgefordert sich bei Google anzumelden, das ist nur notwendig, wenn Sie in der Bearbeitung eine Pause machen wollen und das Teilergebnis zwischenspeichern wollen. Die Befragung endet am 20.12., die Ergebnisse stehen in einem nachfolgenden Fragen&Antworten-Dokument auf der Webseite der Veranstaltung.

Frage: Wir haben Schwierigkeiten ein konkretes Hausarbeitsthema zu finden.

(s. auch FA06) Die Aufgabenstellung ist bewusst offen formuliert, damit sie ein Thema finden, dass sie interessiert. Die Idee ist dann, dieses Framework oder die Bibliothek auszuprobieren und dann eine kleine Aufgabenstellung zu definieren, die sie dann mit OOAD-Mitteln von der Analyse bis zur Implementierung durchziehen. Alternativ entwickeln Sie ein großes Projekt und setzen dann nach der Use Case-Modellierung nur das erste Inkrement um. Generell steht die komplette Durchführung aller Entwicklungsschritte im Mittelpunkt.

Wem das noch zu unkonkret ist, für den ist der Vorschlag sich eine Oberflächentechnologie in Java anzusehen, dann auf einzelne Steuerungselemente wie Knöpfe und Tabellen zu konzentrieren und dann dazu eine Beispielaufgabe zu definieren. Beispiele für Technologien sind: Swing (alt, etabliert, fester Java-Bestandteil), JavaFX (genauer OpenJFX, war mal Teil von Java, muss jetzt selbst ergänzt werden; ist aber in der Java-Version in der KleukerSEU enthalten), Google Web Toolkit (GWT, Web-Applikationen mit Java), Spring (Web-Applikationen, da einen sehr kleinen Einstieg). Der Tipp ist, wenn Sie kein interessantes Framework kennen, etwas Zeit in Google investieren, bei ein zwei Technologien schauen, ob man das Konzept versteht und Hello-World-artiges zum Laufen bekommt und sich dann eine kleine Aufgabenstellung auszudenken.

Eine einfachere Aufgabenstellung ist die Neuimplementierung z. B. von Teilen des Collection-Frameworks von Java und ein Vergleich mit existierenden Implementierungen. Bei solcher einer Aufgabenstellung ist es eher unwahrscheinlich ein sehr gutes aber auch ein sehr schlechtes Ergebnis zu erreichen.

Frage: Es gibt keine Präsentation?

Antwort: Richtig, anders als beim Projektbericht gibt es bei der Hausarbeit keine Präsentation, so dass mehr Zeit zur Erstellung des Textes zur Verfügung steht. Am Ende existiert ein Textdokument, dass der Prof von vorne nach hinten liest und bewertet. Der Arbeitsaufwand ist identisch für alle Module mit Leistungspunkten.

Frage: Wieviele Aufgabenblätter gibt es?

Antwort: 11 bewertete, Blatt 9 ist das letzte (gab zwei b-Blätter).

Frage: Müssen wir Sequenzdiagramme in der Hausarbeit nutzen und sollen wir sie generieren lassen.

Meine Standardantwort bei „müssen“ ist meist: Sie müssen das nicht machen und ich muss ihnen auch keine sehr gute Note geben. Konkret hängt es hier später von der von Ihnen gewählten Aufgabenstellung ab, ob Sequenzdiagramme und wie viele sinnvoll sind. Wenn der Fokus z. B. auf der Kommunikation von GUI-Elementen liegt, gibt es oft ein zentrales Prinzip, so dass ein Diagramm reicht. Da im Sequenzdiagramm nur alle relevanten Klassen sichtbar sein sollen, kann man das Diagramm generieren oder auch von Hand zeichnen. Ein Beispiel ist die Klasse Messages in der Aufgabe 21, die für das Verständnis der MVC-Idee sicherlich nicht relevant ist und weggelassen wird. Ähnliches gilt für die EinUndAusgabe-Klasse, wobei die eigentliche Ausgabe wieder sehr wichtig ist, aber auch einfach mit einem Pfeil von der View-Klasse zu Extern mit der Ausgabe auf dem Pfeil dargestellt werden könnte.

Beachten Sie, dass meine Vorgaben nur das enthalten können, bei deren vollständiger Abarbeitung ein Ergebnis herauskommt, dass die durchschnittlichen Erwartungen erfüllt. Das entspricht der Basisnote 3.0.

Das ist nebenbei der Grund, warum ich nicht noch detaillierter in die Aufgabenstellung einsteige. Würde ich das machen, wäre es immer schwerer eine (sehr) gute Note zu erreichen. Sehr gute Leistungen entstehen nie durch Pflichterfüllung.

Frage: In Java gibt es die Klassen Oberserver und Observable, die sind aber deprecated, sollen wir sie nutzen? Hat das was mit dem Flow-API zu tun?

Antwort: Den genauen Grund für „deprecated“ kenne ich nicht. Da es aber üblich ist den Observer-Observable-Ansatz in die eigenen Klassen und Interfaces zu integrieren, spielen diese Klassen in der Umsetzung in der Praxis kaum eine Rolle. Also integrieren und implementieren Sie den Ansatz selbst.

Das Flow-API hat eine gewisse Verwandtschaft, dient aber einen eigen Architekturansatz, wie er z. B. mit dem Framework RxJava umgesetzt werden kann. Wenn Sie sich für interessante Architekturansätze interessieren, ist es sehr empfehlenswert, aber definitiv kein „Muss“.

Generell gibt es immer wieder neue Ansätze, wie Java-Streams, Flow-API oder das Play-Acca-Framework, die gerne gehypt werden. Jeder dieser Ansätze ist sinnvoll, bietet mehrere Vorteile, ist aber kein ultimativer Game-Changer. Man kann die Vorteile nutzen, muss es aber nicht und wird dadurch auch nicht zu schlechteren oder weniger performanten SW-Lösungen kommen.

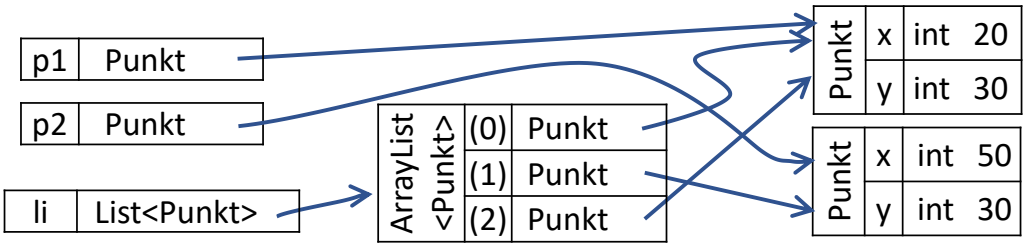
Zum Clonen hier noch ein kleines weiteres Beispiel, warum eine einfache rekursive Nutzung von clone() nur unter bestimmten Randbedingungen zum Ergebnis führt.

```java
public class Beispiel {

  public static List<Punkt> kopieren(List<Punkt> liste){
    List<Punkt> ergebnis = new ArrayList<>();
    for(Punkt e:liste) {
      if (e == null) {
        ergebnis.add(null);
      } else {
        ergebnis.add(e.clone());
      }
    }
    return ergebnis;
  }

   public static void main(String[] s) {
     List<Punkt> li = new ArrayList<>();
     Punkt p1 = new Punkt(20,30);
     Punkt p2 = new Punkt(50,30);
     li.add(p1);
     li.add(p2);
     li.add(p1);
     System.out.println("" + (li.get(0) == li.get(2)) );
     List<Punkt> li2 = kopieren(li);
     System.out.println("" + (li2.get(0) == li2.get(2)) );
   }
}
```

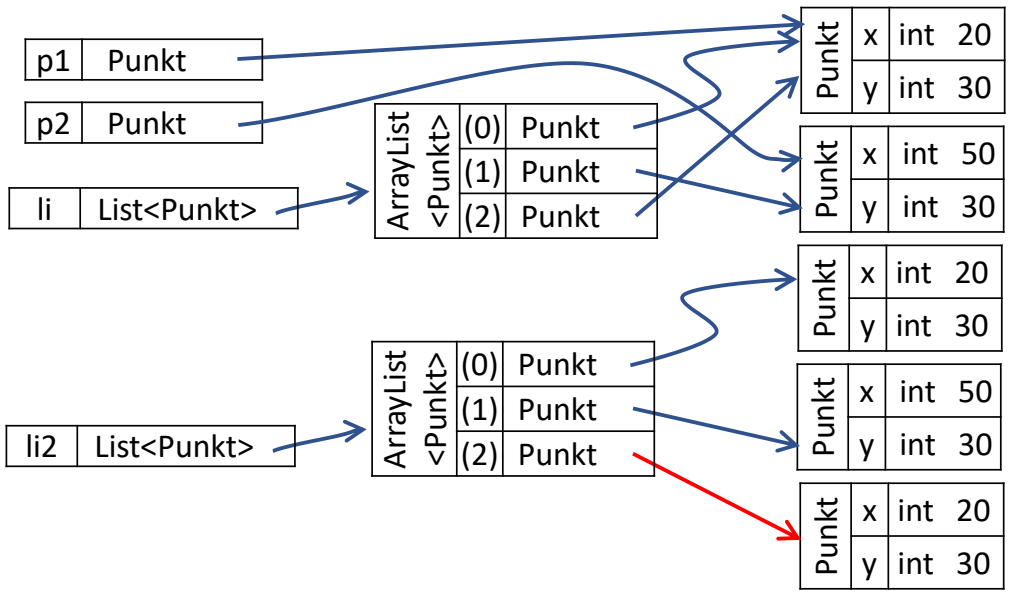
Problems @ Javadoc Console Declaration Coverage

```
<terminated> Beispiel [Java Application] C:\kleukersSEU\java\bi
true
false
```

Die Methode kopieren() kopiert einfach jedes existierende Element. In der Liste li befinden sich identische Elemente an den Positionen 0 und 2, wie auch das folgende Objektspeicherdiagramm zeigt.



Beim Erstellen der Kopie stehen dann nur gleiche Objekte an den Positionen 0 und 2.



Als Lösung kann man sich schon geclonte Objekte z. B. in einer zu übergebenden HashMap merken und vor jedem Clonen prüfen, ob das Objekt schon geclonet wurde und ggfls. dieses als Ergebnis liefern. Dies HashMap muss dann bei jedem Clone()-Schritt als Parameter mitgegeben werden. Der XML-Ansatz stellt dies durch Referenzen in XML sicher, wie der folgende Ausschnitt zeigt. Alternativ können hier Binär-Kopien zu schnellen sauberen Clone-Ergebnissen führen.

```
<void property="liste1">
  <void method="add">
   <object class="entity.Punkt" id="Punkt0">
    <void property="x">
     <int>20</int>
    </void>
    <void property="y">
     <int>30</int>
    </void>
   </object>
  </void>
  <void method="add">
   <object class="entity.Punkt">
    <void property="x">
     <int>50</int>
    </void>
    <void property="y">
     <int>30</int>
    </void>
   </object>
  </void>
  <void method="add">
   <object idref="Punkt0"/>
  </void>
 </void>
```

Frage: Wir haben es mit Streams beim Clonen versucht, da ging aber gar nichts.

Achtung, es gibt zumindest zwei Varianten, in denen in Java der Begriff Stream benutzt wird. Die klassische Variante ist auch in den Beispiellösungen zum Clonen enthalten und nutzt Klassen wie

FileInputStream, FileOutputStream oder die 16-Bit-Varianten FileReader und FileWriter. Dies sind alles typische Klassen die eine Dateibearbeitung ermöglichen.

Der Stream-Begriff ist dann in anderer Form bei der Einführung von Lambda-Ausdrücken und der funktionalen Programmierung in Java erneut relevant, z.B.

```
int[] arr= {9,7,3,1};
Arrays.stream(arr)
      .forEach(i -> System.out.println(i));
```

Es können u. a. Collections in eine Stream-Verarbeitung gesteckt werden, mit der jedes Objekt des Streams schrittweise in der danach beschriebenen Form verarbeitet wird. Diese Streams können dabei beliebig lang sein und sehr unterschiedliche Quellen haben. Das Thema „Einführung in die reaktive Programmierung in Java anhand eines Beispiels“ ist in diesem Zusammenhang auch für Hausarbeiten wählbar.

Diese kurze Übersicht soll zur Vermutung führen, dass die beiden Stream-Begriffe in der Nutzung direkt auf wesentlich andere Anwendungsszenarien zielen, was ein sinnvoller Basisgedanke ist. In einer abstrakteren Sichtweise stellt sich die Frage, warum dann der gleiche Begriff gewählt wurde. Die Vermutung, dass jede Art von Byte-Stream auch als funktionaler Stream betrachtet werden kann, ist dabei sinnvoll. Es gibt also auch funktionale Möglichkeiten Dateien zu lesen und zu schreiben, was allerdings beim Clone meiner Erfahrung nach nicht hilft (?).

```
public static void main(String[] args) {
  try(Stream<Path> projekt = Files.list(Path.of("../"))) {
    projekt.filter(Files::isDirectory)
           .forEach(System.out::println);
   } catch (IOException e) {
      System.out.println("Execption: " + e);
   }
}
```