Inhalt - Überblick

1. **Java Grundlagen: Entwicklungszyklus, Entwicklungsumgebung** Ausführungsmodelle
2. Datentypen, Kodierung, Binärzahlen, Variablen, Arrays Binärzahlen
3. Ausdrücke, Operatoren, Schleifen und Verzweigungen Reihenfolgen Operatoren n++, ++n ....
4. Blöcke, Sichtbarkeit und Methoden (Teil 1) Stacktrace
5. Grundkonzepte der Objektorientierung Getränkeautomat nicht wichtig
6. Objektorientierung: Sichtbarkeit, Vererbung, Methoden (Teil 2), Konstruktor transitive Vererbung => folgt einen Baum (statische Variable nochmal anschauen)
7. Packages, lokale Klassen, abstrakte Klassen und Methoden, Interfaces, enum packages, Sichtbarkeit, was sind abstract class pro/contra
8. Arbeiten mit Objekten: Identität, Listen, Komparatoren, Kopien, Wrapper, Iterator early out condition, equals, listen, Sortierung, clone Unterschiede, Iterator-Interface
9. Fehlerbehandlung: Exceptions und Logging Exceptions, log lvl
10. Utilities: Math, Date, Calendar, System, Random Date!, Zeitzonen, Unix-Zeit, Kalender-klasse, error lvl => Random anschauen (halten wir nicht UL)
11. ~~Rekursion, Sortieralgorithmen und Collections~~
12. Nebenläufigkeit: Arbeiten mit Threads Thread-konkurrenz, Zeitscheiben multiplexing
13. ~~Benutzeroberflächen mit Swing~~
14. ~~Streams: Auf Dateien und auf das Netzwerk zugreifen~~

Klausur 4 Aufgaben (generell, je länger der Absatz, je mehr wird gefragt)

1. Wissensfrage z.B. Binär, Error lvl, Code lesen

2. Anwendung (größe) z.B. Kassenverwaltungsprogramm
   - Klassenverzeichnis (Pfeil => Vererbungspfeil) (Klassennamen groß)
     - Sequenzdiagramm, kein UML
   - 2x Methoden programmieren (getter, setter nicht nötig)

   (O package - protected; Sichtbarkeiten mit dazu)

3. a) Was macht der Code
   => was passiert, nicht Code einzeln kommentieren

   b) Fehlersuche bei Code
   - finden und verbessern

4. Multithreading
   - Muss man überhaupt syncen? Gibt es einen Konflikt zwischen Threads?