1. Was können Datencontainer des Collection Framework enthalten?

* können verschiedenste Objekte enthalten
* können beliebig viele Objekte aufnehmen
* können auf bestimmte Objekte typisiert werden

1. Welche drei Arten von Containern kennen Sie?

* Listen (List), Mengen (Set), Schlüssel-Wert-Paare (Map)

1. Worin unterscheiden sich die drei Containerarten?

* List
  + Zugriff sequentiell oder wahlfrei
  + Duplikate erlaubt
  + Reihenfolge des Einfügens bleibt erhalten
* Set
  + Zugriff erfolgt über Iteratoren
  + keine Duplikate
  + Reihenfolge des Einfügens bleibt nicht erhalten
* Map
  + zusammengehörige Objektpaare
  + Schlüssel sind immer eindeutig
  + Zugriff über Schlüssel

1. Was ist der Unterschied zwischen Arrays und Containern des Collection Framework?

* Größe der Liste wird dynamisch beim Einfügen oder Löschen von Elementen angepasst

1. Sie möchten ein Schachfeld oder auch den Spielplan von „Schiffe versenken“ in Java implementieren. Dazu müssen Sie einen Repräsentanten des Spielfeldes im Speicher erzeugen und verwalten. Welche Datenstrukturen würden Sie dabei verwenden? Wie begründen Sie Ihre Entscheidung?

* Eine Map man könnte zB A5 als Schlüssel nehmen

1. Wozu werden Iteratoren benötigt?

* einheitlicher Standard zum Durchlaufen von Datencontainern
* es können keine Elemente übersprungen werden

1. Was müssen Sie bei Containern beachten, in denen die Objekte sortiert abgelegt werden?

* bei Änderung des Containerinhalts muss der Iterator neu erzeugt werden

1. Wozu brauchen Sie die Interfaces Comparable und Comparator?

* Comparable

Sortiert nach der natürlichen Ordnung Elemente beim Einfügen in Sets oder Maps

muss für alle Klassen implementiert werden, deren Instanzen in Sets oder Maps gespeichert werden

* Comparator

Sortierung erfolgt nach einer beliebigen Sortierreihenolge und übersteuert die natürliche Ordnung

Comparator sollten in eigener Klasse implementiert werden

1. Können Sie Listen-Container sortieren? Wenn ja, wie gehen Sie bei der Realisierung vor?

* static void sort(List liste) sortiert die Liste nach der natürlichen Ordnung
* static void sort(List liste, Comparator c) übersteuert die natürliche Ordnung und sortiert die Objekte der Liste über den entsprechenden Comparator c

1. Welche Möglichkeiten haben Sie, um Objekte miteinander zu vergleichen? Worin liegt der Unterschied der Vergleichsmöglichkeiten?

* ==-Operator Vergleicht Speicherreferenz
* equals()-Methode Vergleicht Inhalt
* Hash-Code Vergleicht Hash-Wert

1. Was müssen Sie beim Überschreiben der equals()-Methode für direkte und indirekte Sub-Klassen der Klasse Object beachten?

* sobald die equals()-Methode überschrieben wird, muss auch die hashCode()-Methode überschrieben werden, so dass o.g. Bedingung erfüllt wird