

Map

- Kopplar nycklar till värden
- Nycklarna är som indexvärden för en List
- En Map kan inte innehålla dubbletter av nycklar
- Map interfacet definierar metoder för förfrågningar, uppdateringar tillgång till nycklar och värden

Map fortsättning

- Ett nyckelvärdepar kallas för ett **Entry**
- Map tillhandahåller metoder för att få tre olika Collectionvyer: *nyckel*, *värde* och *Entry*.
- Map har ett subinterface **SortedMap** som sorterar nycklarna i en Map
- Map tillhandahåller inte stöd för Iterators
- Iterators får du genom dina collection vyer
- Tre konkreta implementationer av Map är: **HashMap**, **LinkedHashMap** och **TreeMap**

LinkedHashMap

- Konkret klass som implementerar Map
- Ordnad efter insättningsordning
- Access order

HashMap

- Konkret klass som implementerar Map
- Är inte ordnad
- Bra prestanda vid operationer på att hitta, sätta in och ta bort en *entry*

TreeMap

- Konkret klass som implementerar Map och SortedMap
- Bra för att stega igenom en collections nyckelvy
- Sorterarad efter nycklarna genom *Comparable* samt *Comparator*

Collections

- Static Factory-klass
- Hjälpklass för Collections-ramverket
- Mest stöd för List
- Har metoder för att manipulera Collection-klasser – **reverse, shuffle, copy**
- Kan skapa Singleton collections – (Singleton - en instans per JVM)
- Kan skapa read-only collections
- Kan skapa synchronized collections

Arrays

- Hjälpklass för vanliga arrayer
- Kan t.ex. konvertera en array till en List
- Kan sortera och kopiera arrayer

Metoden hashCode()

- Finns definerad i java.lang.Object klassen – alla ärver denna metod
- Om du definerar equals-metoden i din klass ska du alltid definiera din egen hashCode-metod
- Tre regler vid skapande av en bra hashCode-metod
 1. Ditt objekt måste ge samma hash vid upprepad exekvering (om du inte ändrar på ditt objekt)
 2. Om två objekt är lika enligt dess equals-metod måste hashCode() ge lika hash-resultat
 3. Om två objekt som inte är lika enligt equals-metoden behöver inte hashCode() ge olika resultat
(även om detta är att föredra)

Metoden hashCode() fortsättning

- Exempel på hashCode():

```
// Datamedlemmar hos en klass
private int userId = 1001;
private int age = 20;
private String username = "anca01";
...
public int hashCode(){
    int result = 15;           // ta ett tal, inte noll
    result = 37*result + userId ; // multiplicera med ett primtal plus din variabel
    result = 37*result + age; // gör likadant med dina andra varibler
    result = 37*result + username.hashCode();
    return result;
}
```

- Om du har objekt som attribut i ditt objekt anropa hashCode() metoden på dem och lägg detta till resultatet

Övning

- Sätt nya attribut på dina Animal-klasser t.ex. *weight*, *age*, *nickname* etc.
 1. Definiera metoden **hashCode()** och **equals()** i dina Animal-klasser
 2. Ditt Zoo har fått ett nytt krav. De får inte ha djur som har samma namn.
Använd passande Map-implementation