

Opgaver onsdag den 12. marts

Opgave 1

Lav en Bil klasse med attributterne registreringsnummer, mærke, model og farve (og constructor, toString samt get/set metoder)

Lav en App klasse med en main() metode. Opret et HashSet og indsæt nogle Bil objekter i denne. Lav flere biler med præcis de samme attributter.

Hvad sker der, når man indsætter to bil objekter der har præcis de samme attributter i det samme HashSet ?

To biler skal opfattes som værende det samme objekt, hvis de har samme registreringsnummer og mærke. Lav de ændringer der skal til, for at sådanne to biler opfattes som ens. Det betyder, at der kun skal kunne være en bil med samme registreringsnummer og mærke i det samme HashSet. Prøv det af ved at indsætte objekter i HashSet.

Opgave 2

I denne opgave skal Dictionary interfacet fra opgave 3 onsdag den 19. februar implementeres under anvendelse af hashing.

```
public interface Dictionary<K,V> {  
  
    /**  
     * Returnerer elementet hoerende til noeglen k, hvis noeglen ikke findes  
     * returneres null  
     *  
     * @return objektet med noegle key  
     * @param key  
     *         noeglen elementet skal findes til  
     */  
    public V get(K key);  
  
    /**  
     * Returnerer om dictionary er tom  
     *  
     * @return om map'en er tom  
     */  
    public boolean isEmpty();  
  
    /**  
     * Indsætter objektet value med nøgle key I dictionary. Hvis key allerede  
     * eksisterer overskrives value hoerende til key værdi og den gamle value  
     * returneres. Hvis key ikke er der returneres null. Hverken key eller  
     * value maa vaere null  
     *  
     * @return  
     * @param key  
     *         noeglen objektet skal indsaettes med  
     * @param value  
     *         objektet der skal indsaettes  
     */  
    public V put(K key, V value);  
}
```

```

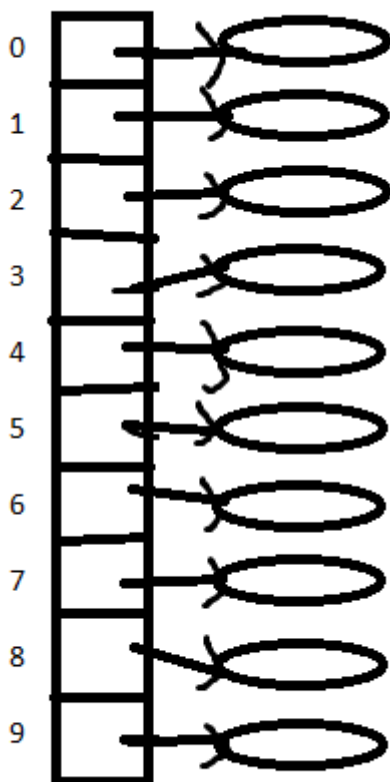
/**
 * Fjerner (noegle, vaerdi)- parret med noeglen key fra dictionary'en og
 * value returneres.
 *
 * @return objektet hoerende til noeglen key; null hvis key ikke findes
 * @param key
 *         noeglen der med tilhoerende objekt skal fjernes
 */
public V remove(K key);

/**
 * Returnerer antallet af elementer i dictionary
 *
 * @return antal elementer i dictionary
 */
public int size();
}

```

Bemærk: null er ikke tilladt som hverken nøgle eller værdi

I den udleverede kode er der lavet en delvis implementation DictionaryHashMap, der implementere interfacet som et array af java.util.HashMap. Kollision håndteres ved kædereglen. Hashfunktionen er givet ved $h(x) = x \bmod N$.



Et array af HashMap.
Hver HashMap kan så indeholde (key,value) – par
I indeks 6 er der et hashMap, indeholdende alle de
key,value par, hvor $key.hashCode() \bmod 10 = 6$

1. Læs den udleverede kode, så du er sikker på at du forstår den inden du går i gang med at programmere
2. Programmer de manglende metoder // TODO
3. Lav en applikation der afprøver implementationen.

Opgave 3

Lav endnu en implementation så Dictionary interfacet realiseres ved et array af List's (fra Javas Collection framework). Kollision håndteres ved kædereglene. Hashfunktionen er givet ved $h(x) = x \bmod N$.

Hvad er størrelsesordenen af tidskompleksiteten for de forskellige metoder.

Opgave 4

Afleveringsopgave kan findes som selvstændig opgave på Canvas.