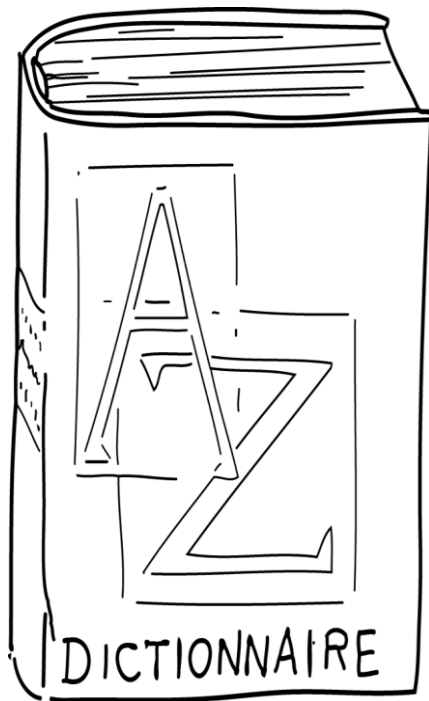


# « MAP »



# « MAP »

Dictionnaire (non trié)

clé - valeur



on associe

Clé	Valeur	type K	type V
n° étudiant	cote	Integer	Double
code-barres	article	Integer	Article
plaque de voiture	propriétaire	String	Personne
voiture	propriétaire	Voiture	Personne
matricule cours	liste de professeurs	String	ArrayList
année d'étude	ensemble d'étudiants	String	HashSet

# Interface DicoNonTrie :

`boolean estVide()`

`int taille()`

`boolean estPresent(K cle)`

`boolean ajoute(K cle, V valeur)`

`V supprime(K cle)`

`V cherche(K cle)`

Unicité de la clé



# Implémentation

Un « hashMap » est un ensemble

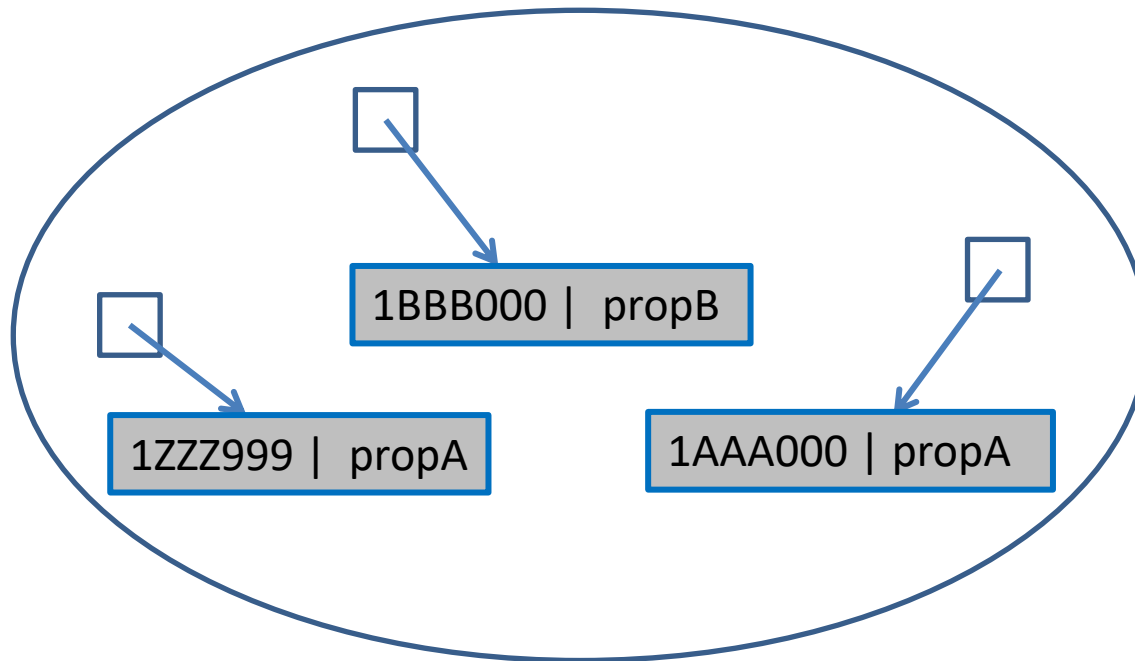
Il est implémenté via une table de hashing

C'est la **clé** qui est « hachée »!

Technique d'implémentation



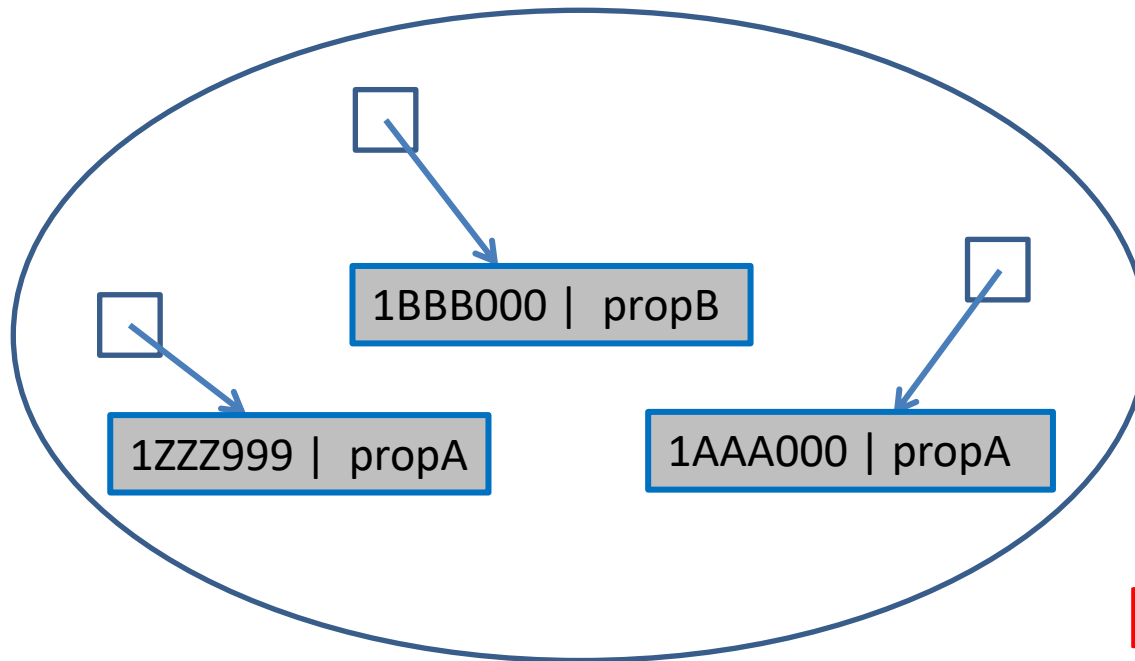
# Ensemble des voitures autorisées



HashSet

Chaque objet voiture  
est unique

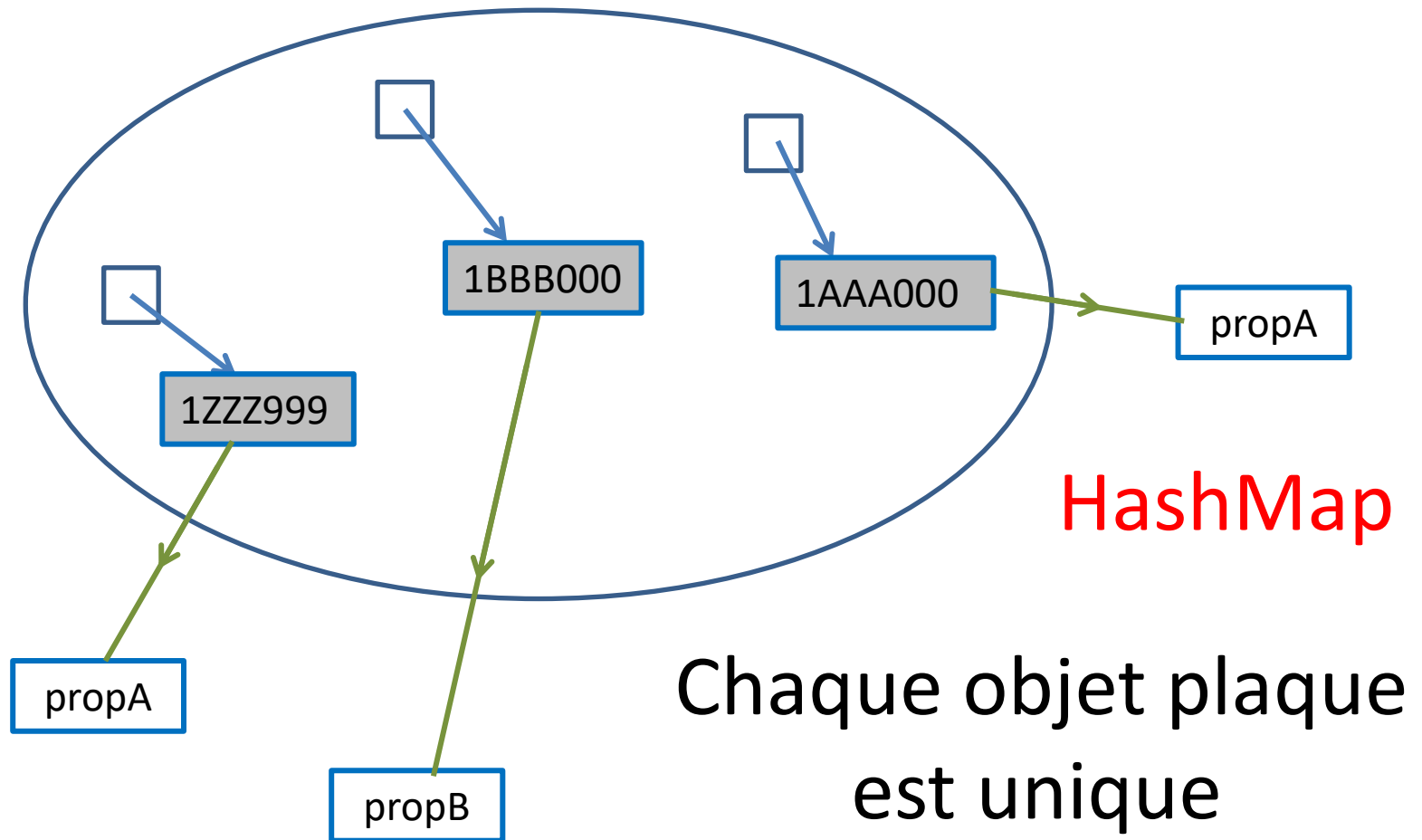
# Ensemble des voitures autorisées



HashSet

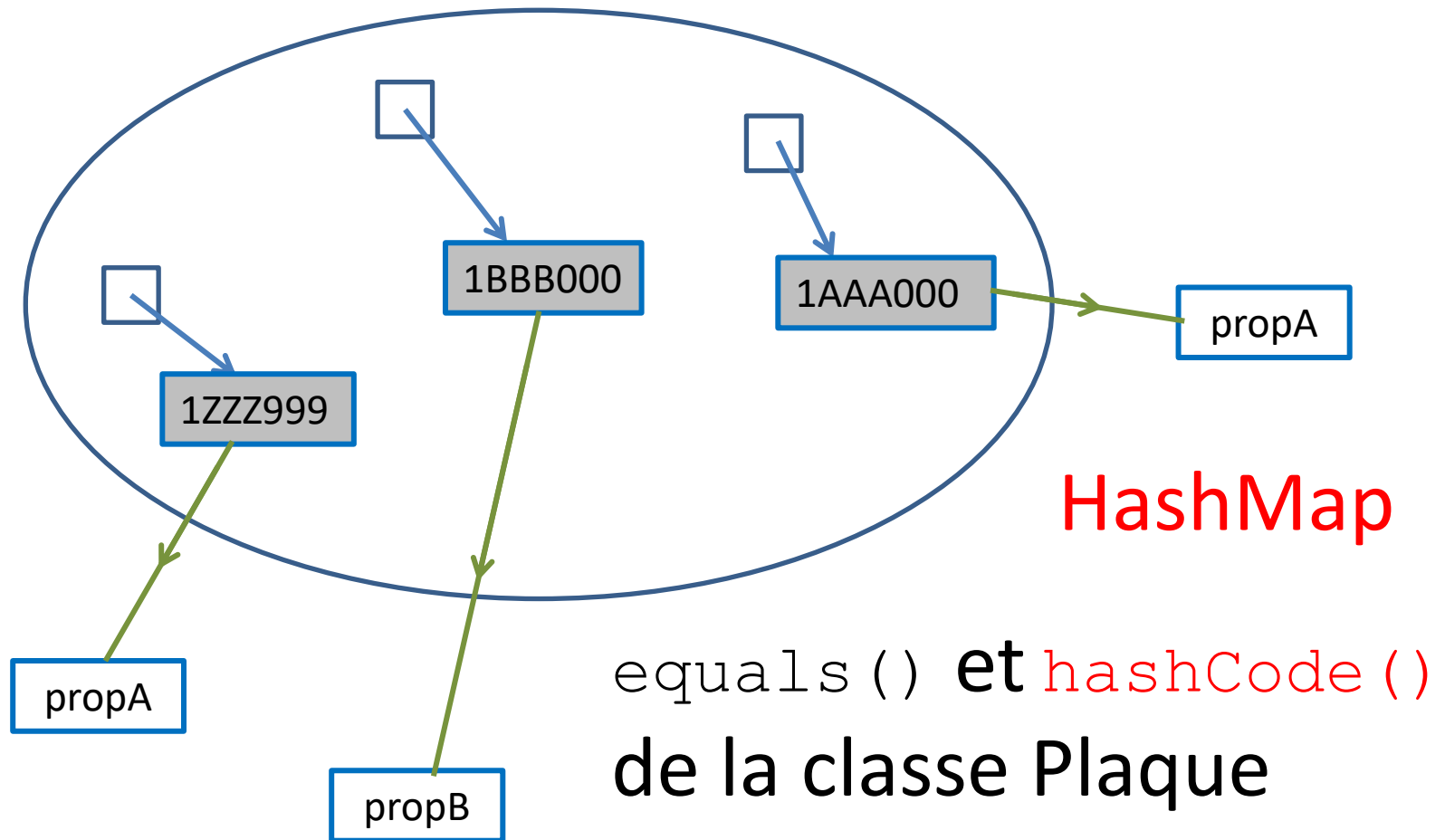
`equals()` **et** `hashCode()`  
de la classe Voiture

# Ensemble des voitures autorisées

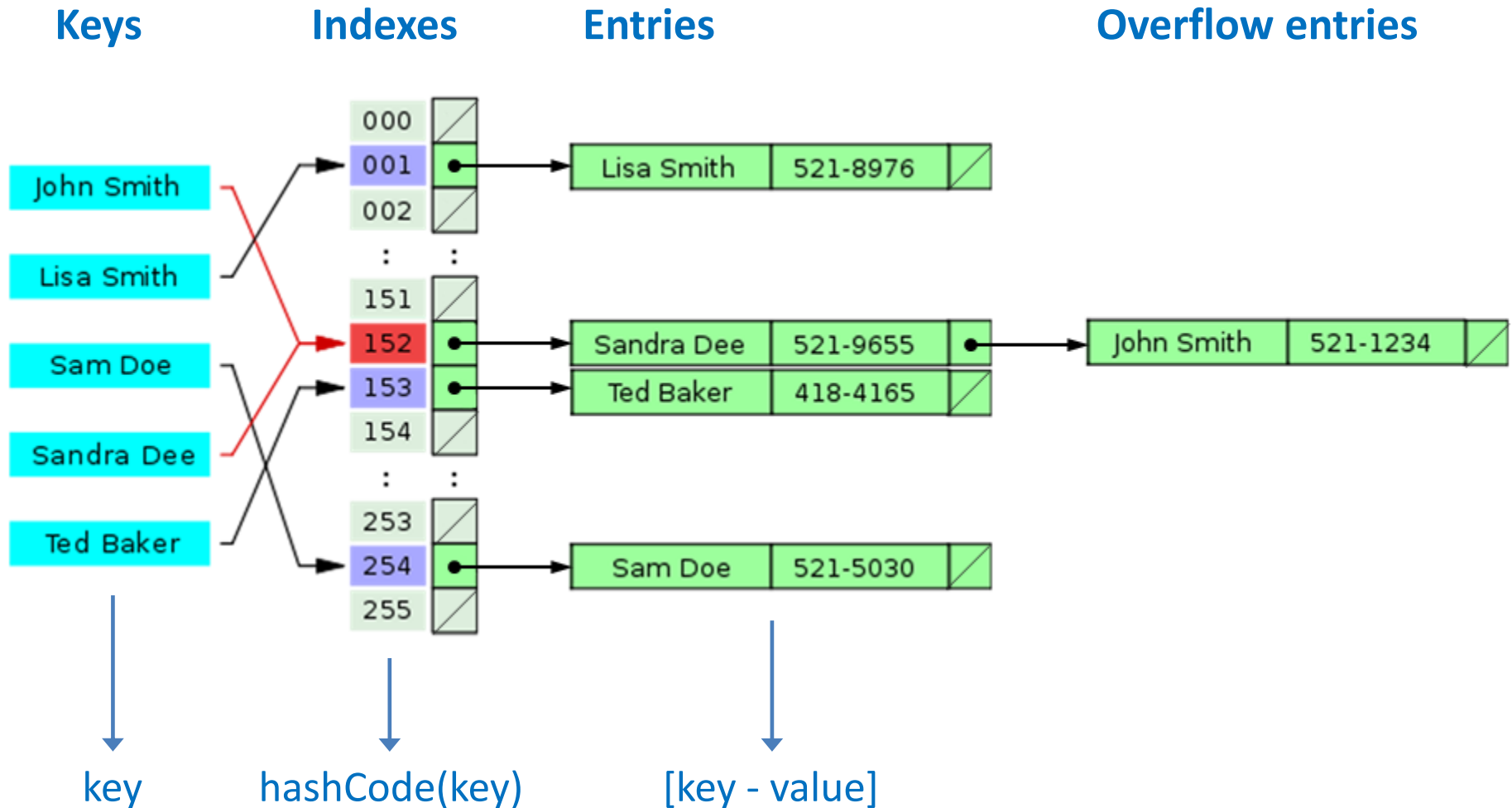




# Ensemble des voitures autorisées



# Java : Implémentation de HashMap avec une table de hachage



# ET JAVA?

DicoNonTrieImpl	HashMap
<code>DicoNontrieImpl(int capacite)</code>	<code>HashMap(int initialCapacity)</code>
<code>int taille()</code>	<code>int size()</code>
<code>boolean estVide()</code>	<code>boolean isEmpty()</code>
<b><code>boolean estPresent(K cle)</code></b>	<b><code>boolean containsKey(Object o)</code></b>

# ET JAVA?

DicoNonTriImpl	HashMap
<code>boolean ajoute(K cle,                   V valeur)</code>	
<code>V remplace(K cle, V valeur)</code>	<code>V put(K key, V value)</code>
<code>V supprime(K cle)</code>	<code>V remove(Object key)</code>
<code><b>V cherche(K cle)</b></code>	<code><b>V get(Object key)</b></code>