## B.1.8 The HashMap class

The abstract [HashMap](#HashMap) class represents a hashmap data structure for storing key-value pairs of objects. This collection can be indexed with the keyElement part of the pair, to receive the valueElement of the pair.
Please note that each key has to be unique according to the given **equalsFunction**.

A new Instance can be created via the external function **createHashMap**.
The hash value of the keyElement object can be calculated using the provided **hashFunctionType** function and the equality of two given keyElements can be determined using the provided **equalsFunctionType** function.

External function and class methods:

* createHashMap
Factory function for creating a new HashMap instance, that will use the provided hashFunction for calculating the hash values of the key element objects and an equalsFunction for determinining the equality of keys. The two functions need to fulfil the property that for all pairs of objects o1, o2, if equalsFunction(o1,o2) is true then also hashFunction(o1)==hashFunction(o2) is true.
* hashFunctionType
A behaviour type allowing the user of the collection to provide their implementation for calculating the hash value of their key element objects.
Functions of this type will be called with a key element object as their only parameter and shall return an integer value that represents the hash value of the object.
Functions of this type can also raise an exception, for example if the object received as their actual parameter is not of the expected class.
* equalsFunctionType
A behaviour type allowing the user of the collection to provide their implementation with an equality relation between key objects insofar that different object instances of the same content can be seen as equal which allows to ensure the uniqueness property for the keys as there can be no two different key instances k1, k2 where equalsFunction(k1.k2) is true.
* put
Adds a new keyElement – valueElement pair to the HashMap.
If the HashMap already contains a pair with the same keyELement, the old pair is removed before inserting the new pair.
Raises an exception in case of error, for example: running out of memory.
* get
Returns the valueElement part of a keyElement – valueElement pair in the HashMap, if such a pair with the provided keyElement object exists in the HashMap.
Raises an exception if the HashMap has no keyElement – valueElement pair with the provided keyElement.
* containsKey
Returns **true** if the HashMap contains a keyElement – valueElement pair with the provided keyElement, **false** otherwise.
Raises an exception in case of error, for example the hashFunction raised an exception.
* remove
Removes a keyElement – valueElement and returns the valueElement part of a keyElement – valueElement pair in the HashMap, if such a pair with the provided keyElement object exists in the HashMap.
Raises an exception in case of error, for example the hashFunction raised an exception.
* keyset
Returns a Set object containing a set of the keyElements of all the keyElement – valueElement pairs in the HashMap.
* values
Returns a List object containing the valueElement objects of all the keyElement – valueElement pairs in the HashMap.
* size
Returns the number of pairs stored in the HashMap.