תכנות מתקדם – תרגיל 3

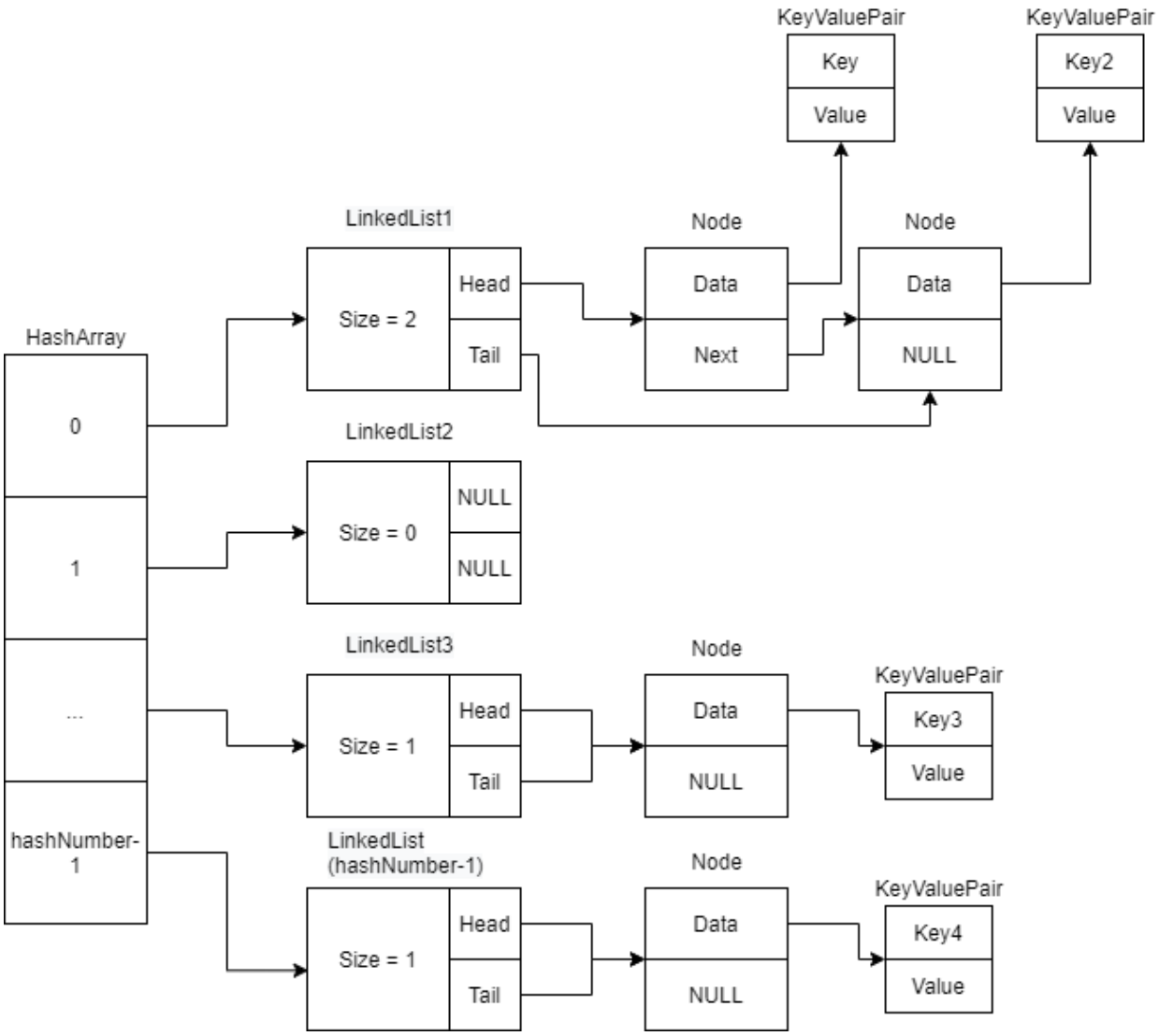
HashTable:

HashTable uses array hashArray with size of hashNumber given in createHashTable. In each cell of hashArray stored LinkedList (HashTable uses chaining for handling collisions). In each node of LinkedLists in hashArray stored KeyValuePair with generic key and value.

All data structures are **generic**, therefore key and value of pair can be any object, but for creating HashTable require copy, free, print, equal, transformIntoNumber functions for ***key*** and copy, free, print functions for ***value***.

Implementation of all functions explained in detail in file HashTable.c as comments.

Schema example of HashTable:



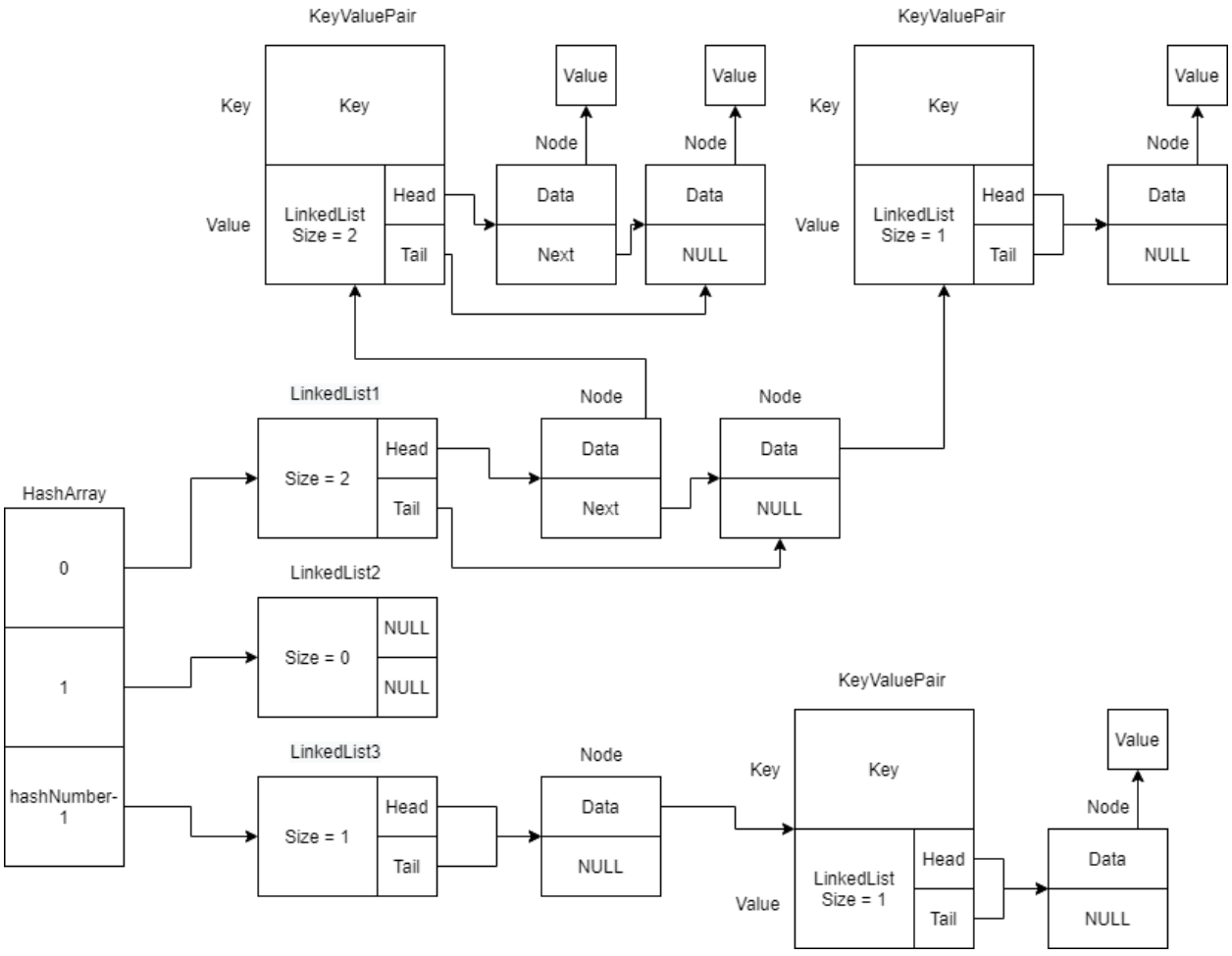
MultiValueHashTable:

MultiValueHashTable uses HashTable with hashArray in it with size of hashNumber given in createMultiValueHashTable. In each cell of hashArray in HashTable stored LinkedList. In each node of LinkedLists in hashArray stored KeyValuePair with generic key and **LinkedList** of generic values. LinkedList in each KeyValuePair used for storing several values for single key

All data structures are **generic**, therefore key and value stored in MultiValueHashTable can be any object, but for creating MultiValueHashTable require copy, free, print, equal, transformIntoNumber functions for ***key*** and copy, free, print, *equal* (equal used for searching specific value by given key) functions for ***value****.*

Implementation of all functions explained in detail in file MultiValueHashTable.c as comments.

Schema example of MultiValueHashTable:



JerryBoree:

JerryBoree uses 5 data structures:

- 3 LinkedLists:

1. allJerries for storing all Jerries created in program
2. allPlanets for storing all planets created in program
3. allOrigins for storing all origins created in program

- HashTable:

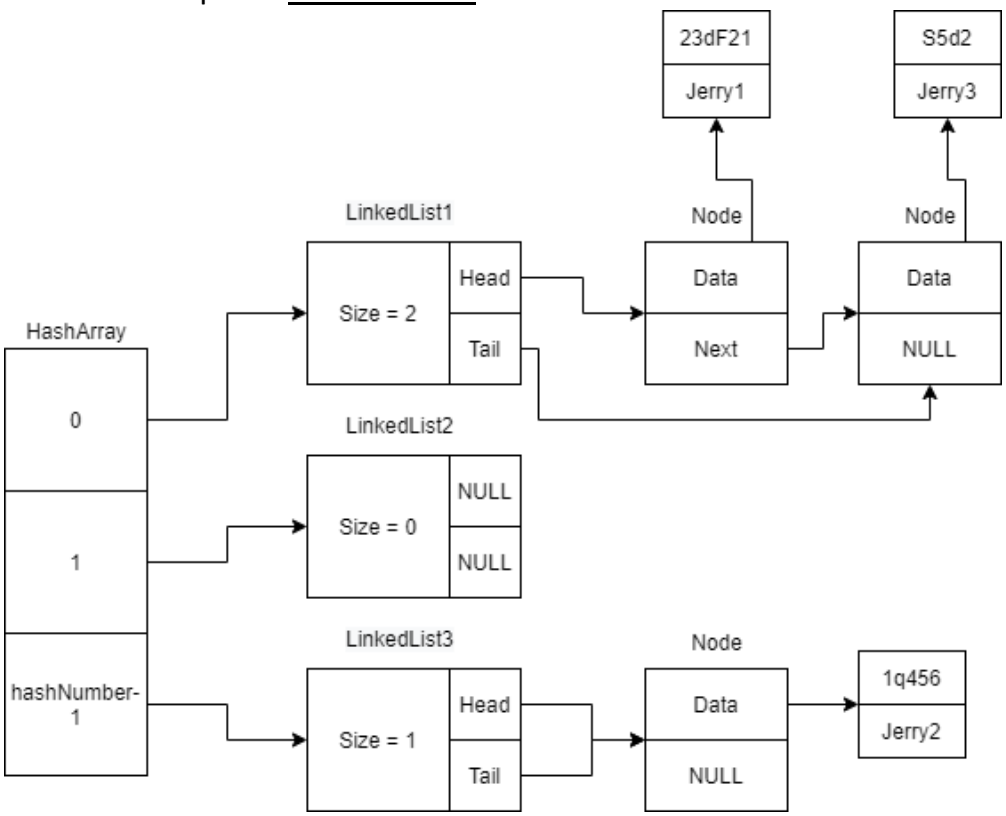
4. hashJerriesID:
   Key: jerryID, type: string (copied in hash)
   Value: jerry, type: jerry (not copied in hash, only pointer stored)
   Size: $number\ of\ jerries\ in\ program\ *\ 3$ (multiplication by 3 for proper $\theta(1)$ time complexity because we can add additional Jerries)
   Used for finding Jerry by his ID in $\theta(1)$

- MultiValueHashTable:

5. multiHashPCs:
   Key: name of physicalCharacteristic, type: string (copied in hash)
   Value: jerry, type: jerry (not copied in hash, only pointer stored)
   Used for finding LinkedList of all Jerries having specific physicalCharacteristic in $\theta(1)$.
   Size: $number\ of\ physical\ characteristics\ in\ program\ (with\ repetitions)$, (without multiplication because we take number with repetitions)

Also, all these data structures, variables and functions explained in detail in JerryBoreeMain.c as comments.

Schema example of hashJerriesID:



Schema example of multiHashPCs: