REPORT

DATA STRUCTURE :- HASHMAP

HashMap<K, V> is a part of Java's collection since Java 1.2. This class is found in java.util package. It uses a technique called Hashing. It implements the map interface. It stores the data in (Key, Value) pairs, and you can access them by an index of another type. One object is used as a key (index)to another object (value).

If you try to insert the duplicate key, it will replace the element of the corresponding key. HashMap contains an array of the nodes, and the node is represented as a class. It uses an array and LinkedList data structure internally for storing Key and Value.

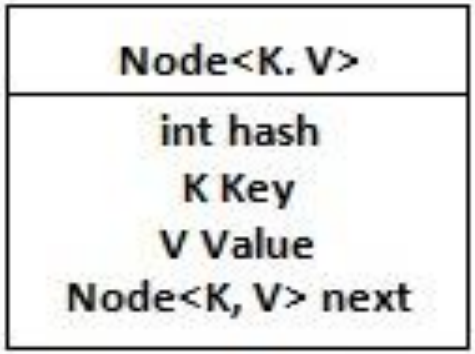
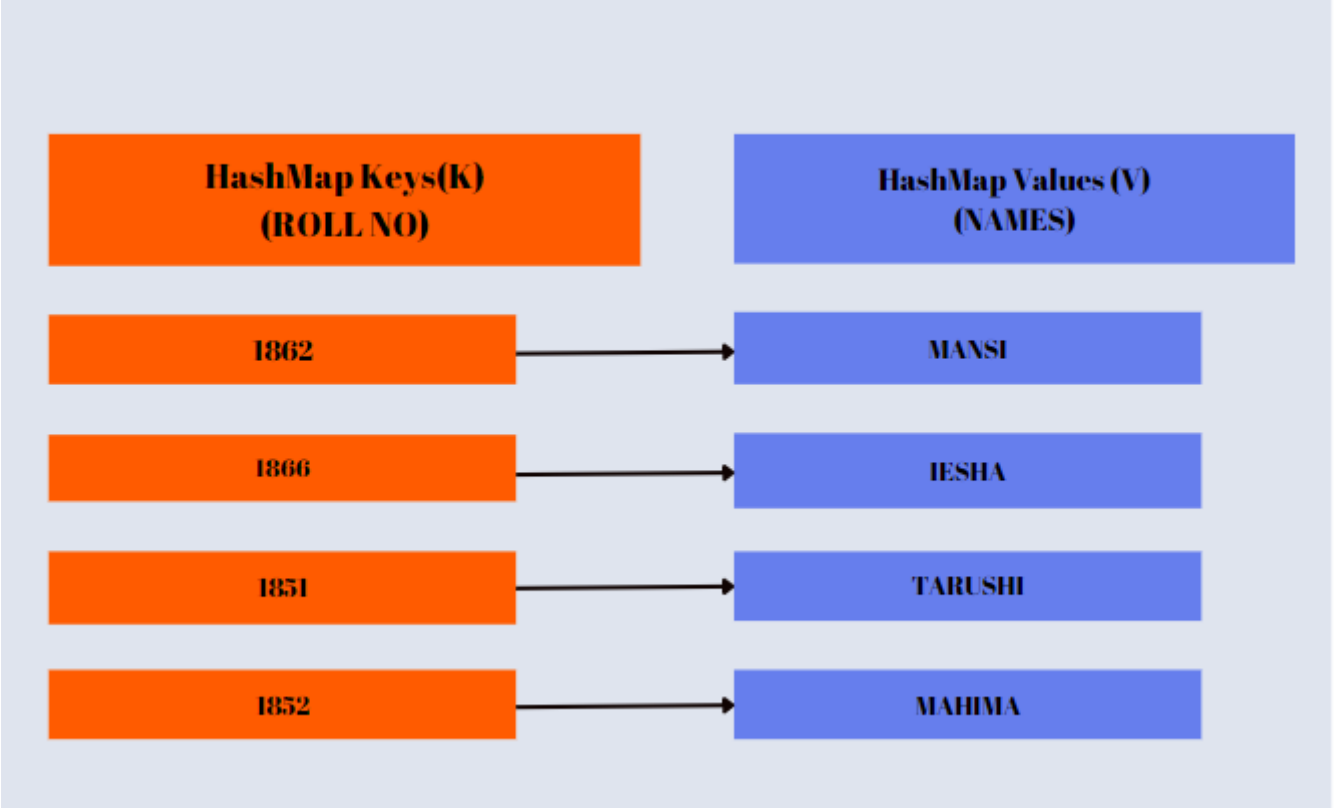


Figure: Representation of a Node



SYNTAX:

```
import java.util.HashMap;
import java.util.Map;
```

HashMap<datatypeOfkey, dataytpeOfValue> <name_of_hashMap> =new HashMap<datatypeOfkey, dataytpeOfValue> ();

Eg=HashMap<Integer, String> <obj> =new HashMap<Integer,String>();

How does HashMap work in Java?

Hashmap uses hashing techniques to store and retrieve elements. For storage, it uses a linked list which is referred to as buckets. It uses two methods on the key: equals()and hashCode() for insert and retrieves operations. While insertion, hashCode determines the bucket for storing. After that, again, hashCode checks whether there is already a key with equal hashCode; if yes, the value is replaced with the new one. If not, then the new map is created into

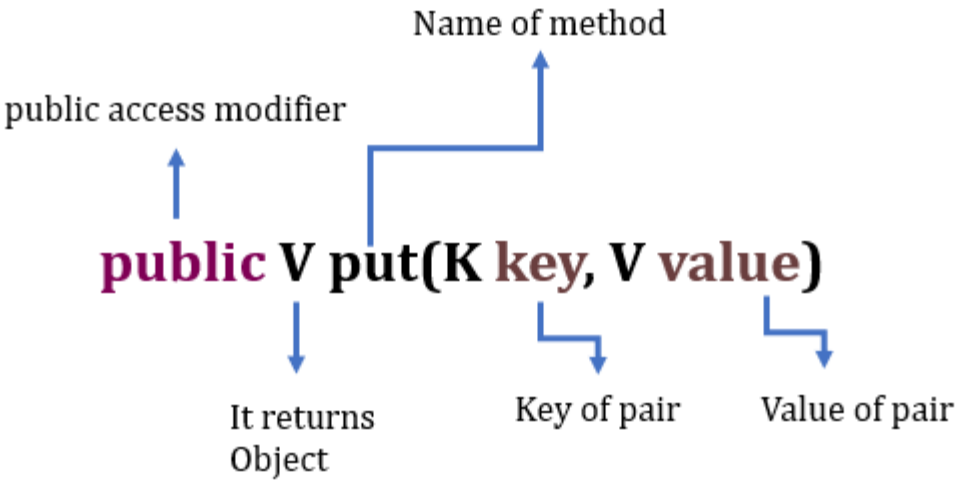
which value will be stored. While retrieval of data, hashCode determines the bucket for searching. After that, hashCode() and equals() gets the value and returns that. It returns null in case no value is present.

Top 13 Methods of HashMap in Java:-

- **public value get(Object key):**
- **public value put(K key, V value):**
- **public boolean containsKey(Object key):**
- **public boolean containsValue(Object value):**
- **public V remove(Object key):**
- **public void clear():**
- **public boolean isEmpty():**
- **Object clone():**
- **public int size():**
- **public Set<Map.Entry<K, V>> entrySet():**

- **public Set<K> keySet():**
- **public void putAll(Map <map_name>):**
- **Collection values():**

HashMap put() Method in Java:-



The java.util.HashMap.put() method of HashMap is used to insert a mapping into a map. This means we can insert a specific key and the value it is mapping to into a particular map. If an existing key is passed then the previous value gets replaced by the new value. If a new pair is passed, then the pair gets inserted as a whole.

Syntax:-

Hash_Map.put(key, value)

key: This refers to the key element that needs to be inserted into the Map for mapping.

value: This refers to the value that the above key would map into.

HashMap get() Method in Java:-

The java.util.HashMap.get() method of HashMap class is used to retrieve or fetch the value mapped by a particular key mentioned in the parameter. It returns NULL when the map contains no such mapping for the key.

java-collection-framework-fundamentals-fundamentals-self-paced

Syntax:-

Hash_Map.get(Object key_element)

CODE:-

```
1
2  import java.util.HashMap;
3
4  public class Main {
5     public static void main(String args[]) {
6                 HashMap<Integer, String> map = new HashMap<>();
7
8         map.put(1862,"Mansi");
9         map.put(1866,"Iesha");
10        map.put(1851,"Tarushi");
11        map.put(1852,"Mahima");
12        System.out.println("HashMap="+map);
13
14        map.put(1866,"Naina");
15        System.out.println("Hashmap after changing value:"+map);
16
17        if(map.containsKey(1862)) {
18            System.out.println("The searched key is present in the map");
19        } else {
20            System.out.println("The searched key is not present in the map");
21        }
22        System.out.println("Accessing the value using key:"+map.get(1866)); //key exists
23        System.out.println("Checking wether the key is present in the map:"+map.get(1761)); //key
24        map.remove(1851);
25        System.out.println("HashMap after remove method:"+map);
26    }
27 }
```

OUTPUT:-

```
HashMap={1862=Mansi, 1866=Iesha, 1851=Tarushi, 1852=Mahima}
Hashmap after changing value:{1862=Mansi, 1866=Naina, 1851=Tarushi, 1852=Mahima}
The searched key is present in the map
Accessing the value using key:Naina
Checking wether the key is present in the map:null
HashMap after remove method:{1862=Mansi, 1866=Naina, 1852=Mahima}
```

HashMap Application:-Snake And Ladders:-

```
import java.util.Hashap;
    import java.util.Map;
    import java.util.Random;
    import java.util.Scanner;
```

```
public class SnakeLadderTest {
    public static void main(String[] args) {
        SnakeNLadder snakeLadder = new SnakeNLadder();
        snakeLadder.startGame();
    }
}
class SnakeNLadder {
    final static int WINPOINT = 100;
    static Map<Integer, Integer> snake = new HashMap<>();
    static Map<Integer, Integer> ladder = new HashMap<>();
    {
        snake.put(99, 54);
```

```
        snake.put(70, 55);
        snake.put(52, 42);
        snake.put(25, 2);
        snake.put(95, 72);


        ladder.put(6, 27);
        ladder.put(11, 40);
        ladder.put(60, 85);
        ladder.put(46, 90);
        ladder.put(17, 69);
}
```

```
public int rollDice() {
        int n = 0;
        Random r = new Random();
        n = r.nextInt(7);
        return (n == 0 ? 1 : n);
}

public int calculatePlayerValue(int playerPosition, int diceValue) {
        int playerNewPosition=playerPosition+diceValue;
```

```
if (playerNewPosition > WINPOINT)
        return playerPosition;


if (null !=snake.get(playerNewPosition)) {
        System.out.println("Oops..swallowed by the snake..");
        playerNewPosition=snake.get(playerNewPosition);
}

if (null !=ladder.get(playerNewPosition)) {
        System.out.println("YAY! climbing the ladder..");
```

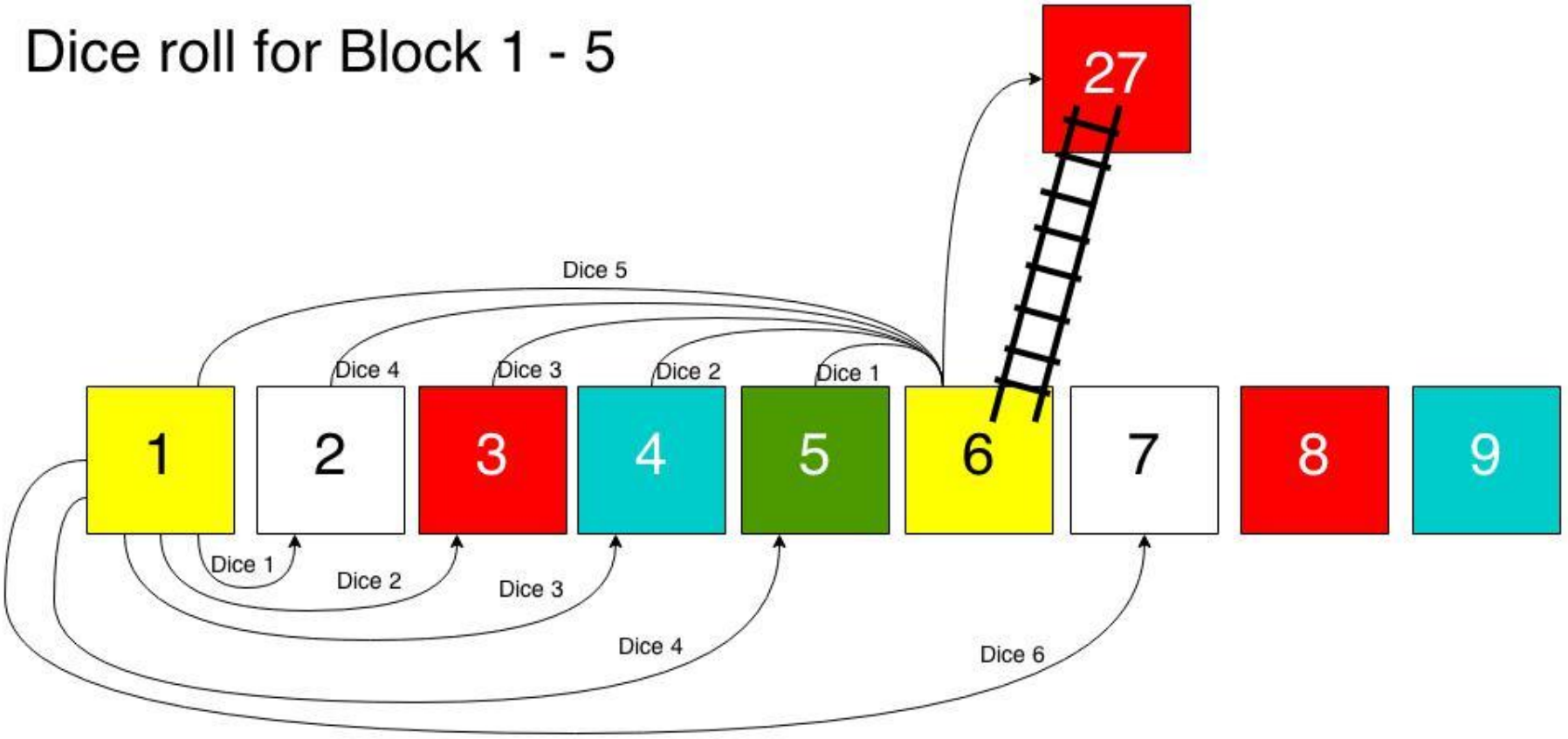
```
            playerNewPosition=ladder.get(playerNewPosition);
        }

        return playerNewPosition;
    }

    public boolean isWin(int playerPosition) {
        return WINPOINT==playerPosition;
    }

    public void startGame() {
        int player1Position=0, player2Position=0;
```

```
int currentPlayer=-1;
Scanner scan= new Scanner(System.in);
String rPressed;
int diceValue = 0;
do {
    System.out.println(currentPlayer == -1
    ? "\n\nFirst player's turn" : "\n\nSecond player's turn");
    System.out.println("Press 'r' to roll Dice");
    rPressed=scan.next();
    diceValue=rollDice();
```

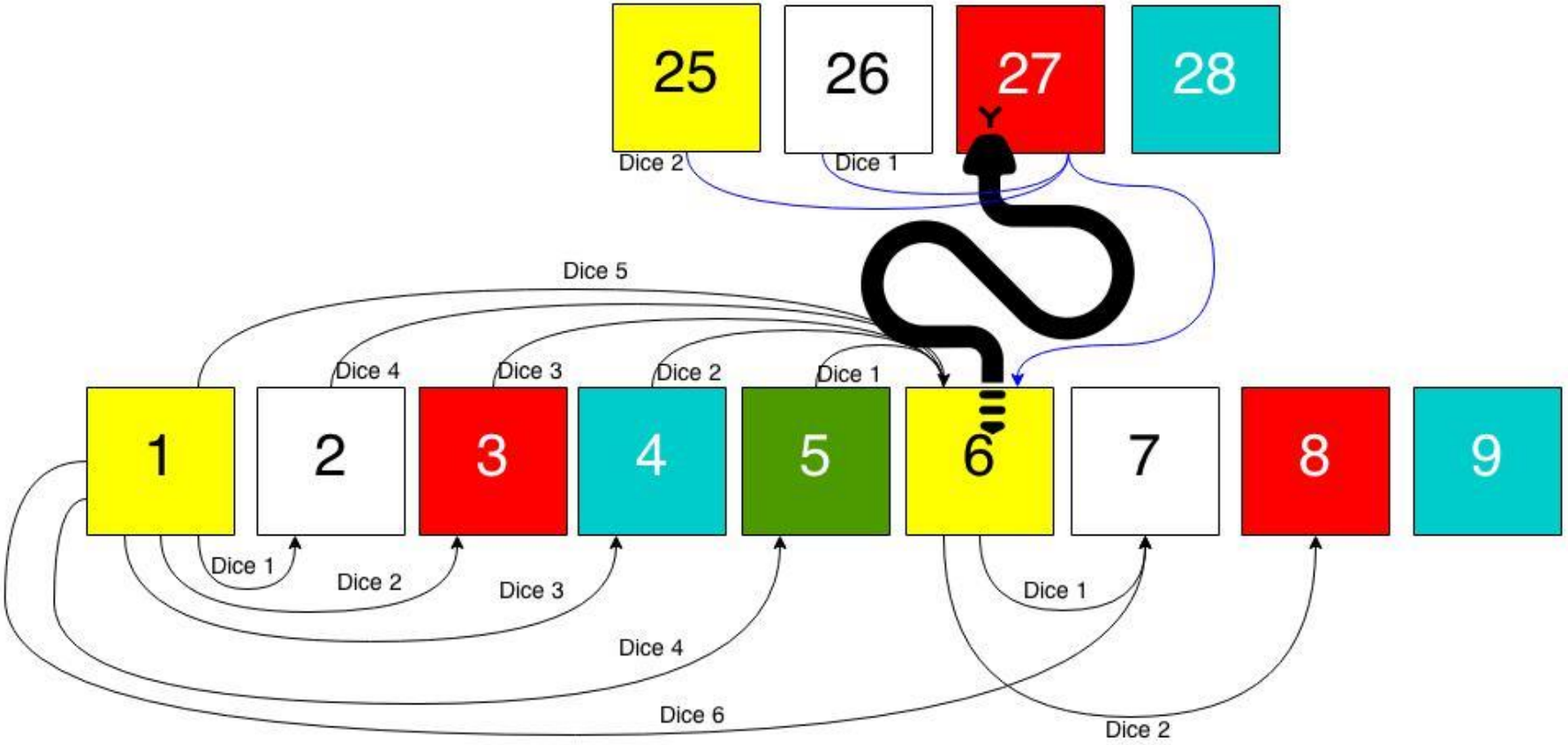
```
                if (currentPlayer==-1) {
                  player1Position=calculatePlayerValue(player1Position,
diceValue);
                  System.out.println("First Player Position:"+player1Position);
                  System.out.println("Second Player Position:"+player2Position);
                  System.out.println("-----------------------");
                  if (isWin(player1Position)) {
                    System.out.println("Congratulations! First player won");
 return;
                  }
                  } else {
                    player2Position = calculatePlayerValue(player2Position,
diceValue);
```

```
                System.out.println("First Player Position:"+player1Position);
                System.out.println("Second Player
Position:"+player2Position);
                System.out.println("-----------------------");
                if (isWin(player2Position)) {
                  System.out.println("Congratulations! Second player won");
                  return;
                }
              }
              currentPlayer = -currentPlayer;
            } while ("r".equals(rPressed));}}
```

Dice roll for Block 1 - 5



Dice roll for Block 25



Why HashMap is used instead of Array:-

If you use an array you would either need to resize it or initialise it to the Maximum possible snake length to start with which can be wasteful on memory.

Reference sites:- https://www.javatpoint.com/working-of-hashmap-in-java#:~:text=It%20uses%20an%20array%20and,are%20four%20fields%20in%20HashMap.

https://www.geeksforgeeks.org/java-util-hashmap-in-java-with-examples/

https://youtu.be/WeF3_nk-UqY

Contribution: -

Mahima=Algo,poster,pseudo code

Tarushi: -Content,poster,pseudo code

Iesha=Poster & Report

Mansi: -Poster & Report