

# Java HashMap

I. За структурите Arrays знаем, че съхраняват елементите в подредена редица и достъпът до елементите се осъществява чрез индекс (int type).

**HashMap структурата обаче, съхранява елементите си чрез "key/value" заедно и достъпът до елементите се осъществява чрез индекс от други типове (напр. String).**

При HashMap един обект използва key (index) към друг обект (value). Можем да използваме различни типове: String keys и Integer values, или еднакви типове : String keys и String values.

Пример: Създаваме HashMap object , наречен capitalCities , в който ще съхраняваме String keys и String values:

```
import java.util.HashMap; // import the HashMap class
```

.....

```
HashMap<String, String> capitalCities = new HashMap<String, String>(); //декларация
```

## II. Методи за работа с HashMap:

### 1. Добавяне на елементи: put() метод

```
// Import the HashMap class
```

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

```
public class MyClass {
```

```
    public static void main(String[] args) {
```

```
        // създаваме HashMap object с име capitalCities
```

```
HashMap<String, String> capitalCities = new HashMap<String, String>();
```

```
// Добавяне на keys и values (Country, City)
```

```
capitalCities.put("England", "London");
```

```
capitalCities.put("Germany", "Berlin");
```

```
capitalCities.put("Norway", "Oslo");
```

```
capitalCities.put("USA", "Washington DC");
```

```
        System.out.println(capitalCities); // отпечатва цялата структура
```

```
    }
```

```
}
```

**2. Достъп до елементи : за достъп до елемент (value) в HashMap, се използва get() метод , съответстващ на неговия ключ ( key):**

```
capitalCities.get("England");
```

**3. Премахване на елемент(value) : remove() метод, отново обвързан с неговия ключ( key):**

```
capitalCities.remove("England");
```

**4. Премахване на всички елементи : clear() метод**

```
capitalCities.clear();
```

**5. Размерност на HashMap :size() метод (връща цяло число, което някой трябва да получи!)**

```
capitalCities.size();
```

**6. Циклична обработка с for-each**

**Използва се keySet() метода, ако искаме само ключовете и values() метода, ако искаме само стойностите на елементите :**

Пример 1:

```
// Print keys
```

```
for (String i : capitalCities.keySet()) {  
    System.out.println(i);  
}
```

Пример 2:

```
// Print values
```

```
for (String i : capitalCities.values()) {  
    System.out.println(i);  
}
```

Пример 3:

```
// Print keys and values
```

```
for (String i : capitalCities.keySet()) {  
    System.out.println("key: " + i + " value: " + capitalCities.get(i));  
}
```

**Пояснение за Структурата на оператора for-each:**

```
for (type var : array)  
{  
    statements using var;  
}
```

**И тъй като при тези структури ключът е от тип String не можем да използваме оператор за цикъл for.**

### **7.Търсене по ключ в структурата**

//върща булев резултат (true/false)

```
System.out.println("containsKey=Germany?" + capitalCities.containsKey("Germany"));
```

### **8. Търсене по стойност в структурата**

//върща булев резултат (true/false)

```
System.out.println("containsValue=Oslo?" + capitalCities.containsValue("Oslo"));
```

### **III. Други типове данни за ключ и стойност в HashMap.**

Създаваме HashMap object с име people, който ще съхранява данни чрез **String keys u Integer values:**

// Import the HashMap class

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

```
public class MyClass {
```

```
public static void main(String[] args) {
```

// създаване на HashMap object с име people

```
HashMap<String, Integer> people = new HashMap<String, Integer>();
```

// Добавяне на keys и values (Name, Age)

```
people.put("John", 32);
```

```
people.put("Steve", 30);
```

```
people.put("Angie", 33);
```

//Отпечатване на данните

```
for (String i : people.keySet()) {
```

```
System.out.println("key: " + i + " value: " + people.get(i));
```

```
}
```

```
}
```

```
}
```

Напишете показаните програмни конструкции за да научите HashMap как се създава и използва!  
Приятна работа!

Преподавател: инж. В. Маринова