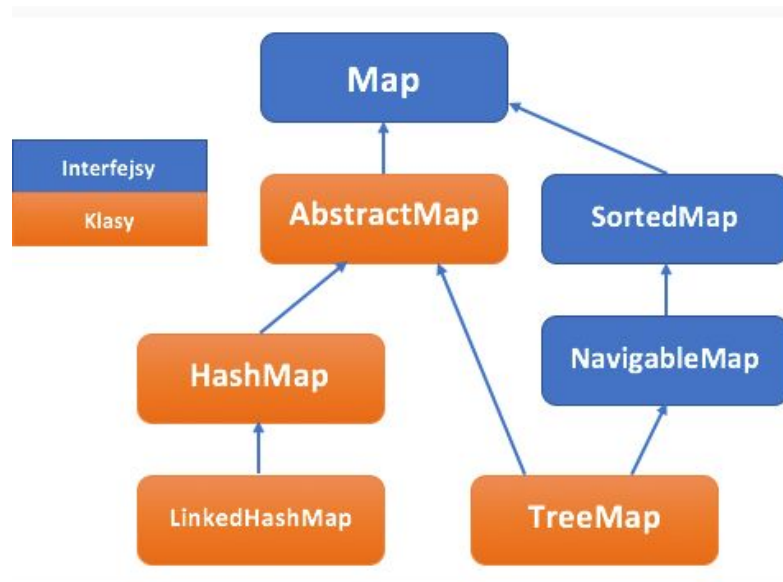
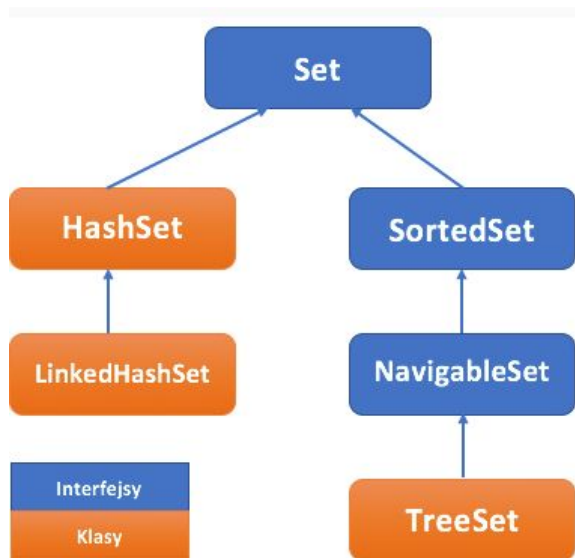





# Map i Set

Oskar Cywiński 2DT5

Mapy i Sety - są to struktury danych, które zawierają właściwości obiektów oraz tablic.





Set - kolekcja, która składa się z unikalnych elementów, gdzie każda z nich może być typu prostego lub złożonego, a kolejność ich przechowywania zależy od implementacji. W przeciwieństwie do Map, jest to zbiór pojedynczych wartości.

Set możemy stworzyć na 2 sposoby:

```
const set = new Set("pies");  
//{"p","i","e","s"}
```

```
const set = new Set();  
set.add(1);  
set.add("text");  
set.add({name: "pies"});  
console.log(set);  
//{1, "text", {name: "pies"}}
```



## Metody, których możemy użyć podczas tworzenia obiektów Set:


- clear() - czyści cały set,
- delete(key) - usuwa dany klucz oraz zwraca wartość zależną od tego czy operacja się udała,
- add(value) - dodaje nową wartość,
- entries() - zwraca iterator zawierający tablicę "klucz - wartość",
- has(key) - sprawdza czy set posiada podany przez nas klucz,
- keys() - zwraca iterator, który zawiera listę kluczy danego setu,
- forEach - robi pętlę po elementach setu,
- size - zwraca długość setu
- values - zwraca iterator, który zawiera listę wartości danego setu.



## Set i tablice

Z powodu tego, że Set posiada niepowtarzające się wartości, można to wykorzystać do odrzucania duplikatów np. w tablicy:

```
const set = new Set("pies");  
console.log(set) //Set {"p", "i", "e", "s"}  
set.add("p");  
set.add("s");  
set.add("e");  
set.add("s");  
console.log(set); //Set {"p", "i", "e", "s"}
```



Map - wykorzystywany do tworzenia zbiorów z parami klucz - wartość. Od normalnych obiektów różni się tym, że kluczami może być dowolny typ danych.

Map można stworzyć na 2 sposoby:

```
const map = new Map([  
  ["kolor1", "green"],  
  ["kolor2", "white"],  
]);
```

```
const map = new Map();  
map.set("kolor1", "green");  
map.set("kolor2", "white");
```



## Metody, których możemy użyć podczas tworzenia obiektów Map:

- set(key,value) - ustawia nowy klucz z daną wartością ,
- get(key) - zwraca zawartość danego klucza,
- has(key) - sprawdza czy mapa ma dany klucz,
- delete(key) - usuwa dany klucz i zwraca wartość zależnie od tego czy operacja się udała,
- clear() - usuwa wszystkie elementy z mapy,
- entries() - zwraca iterator zawierający tablicę par “klucz-wartość”,
- keys() - zwraca iterator zawierający listę kluczy z danej mapy,
- values() - zwraca iterator zawierający listę wartości z danej mapy,
- forEach – robi pętlę po elementach mapy,
- size - zwraca długość mapy.



## Klucze w Mapie

Mapy w przeciwieństwie do obiektów mogą mieć klucze dowolnego typu, gdzie w przypadku obiektów są one konwertowane na tekst.

```
const map = new Map();
map.set("1", "green");
map.set(1, "white");

console.log(map);
//{"1" => "green", 1 => "white"}
```

```
const ob = {}
ob["1"] = "green";
ob[1] = "white";
console.log(ob);
//{"1" : "green"}
```