

TP 1 : Tableaux avec TypeScript

Exercice

Étant donné le tableau suivant qui représente les données personnelles des employés d'une entreprise ainsi que leurs villes d'intervention :

```
const personnes = [
  { type: 'user', nom: 'Max Mustermann', age: 25, villes: ['Marseille', 'Lyon', 'Paris'] },
  { type: 'admin', nom: 'John Wick', age: 45, villes: ['Paris'] },
  { type: 'user', nom: 'Kate Muller', age: 23, villes: ['Nantes', 'Lyon', 'Lille', 'Nice'] },
  { type: 'admin', nom: 'Bruce Willis', age: 64, villes: ['Paris', 'Nantes'] },
  { type: 'user', nom: 'Jack Wilson', age: 35, villes: ['Marseille', 'Lyon', 'Montpellier'] },
  { type: 'admin', nom: 'Carol Smith', age: 23, villes: ['Marseille', 'Nice', 'Montpellier'] }
];
```

Répondez aux questions suivantes en utilisant `forEach`, `map`, `filter`, `reduce`, `every` et `some`.

- Écrivez une fonction fléchée `getByType(type: string): Array<Object>` qui retourne la liste des personnes selon le type passé en paramètre.
- Écrivez une fonction fléchée `getByVille(ville: string): Array<Object>` qui retourne la liste des personnes ayant dans `villes` la ville passée en paramètre.
- Écrivez une fonction fléchée `getOtherVilleThan(ville: string): Array<Object>` qui retourne la liste des personnes n'ayant pas dans `villes` la ville passée en paramètre.
- Écrivez une fonction fléchée `countVilles(): Array<Object>` qui retourne un tableau d'objets : chaque objet contient le nom d'une personne ainsi que son nombre de villes (voir ci-dessous).

```
[{ nom: 'Max Mustermann', nbrVilles: 3 },
 { nom: 'John Wick', nbrVilles: 1 },
 { nom: 'Kate Muller', nbrVilles: 4 },
 { nom: 'Bruce Willis', nbrVilles: 2 },
 { nom: 'Jack Wilson', nbrVilles: 3 },
 { nom: 'Carol Smith', nbrVilles: 3 }]
```

- Écrivez une fonction fléchée `getByVillesNumber(nbr: number): Array<Object>` qui retourne les personnes dont le nombre de villes d'intervention correspond au paramètre `nbr`.
- Écrivez une fonction fléchée `countCharacterInVilles(): Array<Object>` qui retourne un tableau d'objets : chaque objet contient le nom d'une personne ainsi que le nombre total de caractères de ses villes (voir ci-dessous).

```
[{ nom: 'Max Mustermann', total: 18 },
 { nom: 'John Wick', total: 5 },
 { nom: 'Kate Muller', total: 19 },
 { nom: 'Bruce Willis', total: 11 },
 { nom: 'Jack Wilson', total: 24 },
 { nom: 'Carol Smith', total: 24 }]
```

- Écrivez une fonction fléchée `countByVille(ville: string): number` qui retourne le nombre de personnes qui interviennent dans la ville passée en paramètre (ici c'est 3 pour `Marseille` par exemple).

8. Écrivez une fonction fléchée `findHavingMaxVille(): number` qui permet de retourner le nombre max de villes d'intervention de toutes les personnes (ici c'est 4).
9. Écrivez une fonction fléchée `getAllVilles(): Array<String>` qui permet de retourner un tableau de toutes les villes (sans doublons).
10. Écrivez une fonction fléchée `getAvgAge(ville: string): number` qui retourne la moyenne d'âge (d'employés) de la ville passée en paramètre.
11. Écrivez une fonction fléchée `getAvgAgeByVille(): Array<Object>` qui retourne la moyenne d'âge de chaque ville d'intervention sous forme d'un tableau d'objet.
12. Écrivez une fonction fléchée `getYoungerByVille(): Array<Object>` qui retourne le nom de la personne la plus jeune par ville (voir ci-dessous).

```
[
  {
    ville: 'Marseille',
    plusJeune: { type: 'admin', nom: 'Carol Smith', age: 23, villes: [Array] }
  },
  {
    ville: 'Lyon',
    plusJeune: { type: 'user', nom: 'Kate Muller', age: 23, villes: [Array] }
  },
  {
    ville: 'Paris',
    plusJeune: { type: 'user', nom: 'Max Mustermann', age: 25, villes: [Array] }
  },
  {
    ville: 'Nantes',
    plusJeune: { type: 'user', nom: 'Kate Muller', age: 23, villes: [Array] }
  },
  {
    ville: 'Lille',
    plusJeune: { type: 'user', nom: 'Kate Muller', age: 23, villes: [Array] }
  },
  {
    ville: 'Nice',
    plusJeune: { type: 'user', nom: 'Kate Muller', age: 23, villes: [Array] }
  },
  {
    ville: 'Montpellier',
    plusJeune: { type: 'admin', nom: 'Carol Smith', age: 23, villes: [Array] }
  }
]
```