Generics, wildcard

Exercice 1 - Carte sauvage

- 1 Pourquoi le code ci-dessous ne compile-t-il pas ?

```
private static void print(List<Object> list) {
    for(Object o:list)
        System.out.println(o);
}
public static void main(String[] args) {
    List<String> list=Arrays.asList(args);
    print(list)
}
```

Que doit-on changer pour qu'il compile ?

- 2 On veut écrire la méthode `printLength` prenant en paramètre une liste d'objet implémentant l'interface `CharSequence` et affichant la longueur des chaînes de caractères de la liste.

Exercice 2 - Générification

- 1 Générifier le code ci-dessous :

```
public static List listLength(List list) {
    ArrayList length=new ArrayList();
    for(int i=0;i<list.size();i++) {
        CharSequence seq=(CharSequence)list.get(i);
        length.add(seq.length());
    }
    return length;
}
public static void main(String[] args) {
    List l=Arrays.asList(args);
    System.out.println(listLength(l));
}
```

- 1 En utilisant une variable de type `T`
- 2 En utilisant la notation wildcard.
- 2 À quoi sert la constante `Collections.EMPTY_LIST` ?
Comment peut-on l'utiliser dans l'implantation de la méthode `listLength()` ?
- 3 Changer l'implantation de la méthode `listLength()` pour utiliser la méthode `emptyList` de la classe `java.util.Collections`.

Exercice 3 - C'est loin la merge

On souhaite écrire une méthode permettant de fusionner deux listes `List` pour obtenir une liste contenant alternativement un élément de chaque liste.

La méthode devra s'assurer que les deux listes ont la même taille.

Quel est le profil de la méthode `merge` (le plus générique possible) sachant que le code suivant est valide.

```
List<String> list1=...
List<StringBuilder> list2=...
```

```
List<? extends CharSequence> result1=merge(list1,list2);  
List<?> result2=merge(list1,list2);
```

Indiquer sans implanter de nouveau code quelle est la complexité de la solution choisie si une des liste est une `LinkedList` ?

Comment doit-on faire pour avoir une meilleur complexité ?

Ecrire la solution proposée.

Expliquer ce qu'est la capture.