Exercice 1

1) Quel est le type de s ? Comment le compilateur fait-il pour savoir qu'il existe une méthode length() sur s ? -s est de typejava.lang.string, "var" permet de à la variable d'être du bon type.

2) Qu'affiche le code suivant ? Expliquer. -le code suivant affiche true pour la première comparaison et false pour la seconde. -s2 est copié au niveau de la mémoire, pour s3, un nouvel objet à une nouvelle adresse mémoire est crée. La comparaison vérifie si les deux variables ont la même adresse car "==" est utilisé.

3) Quelle est la méthode à utiliser si l'on veut tester si le contenu des chaînes de caractères est le même ? -Il faut utiliser la methode .equal()

4) Qu'affiche le code suivant ? Expliquer -le code suivant affiche true, les deux variables sont initialisés avec la même chaine de charactère. les deux variables sont référées à la même adresse mémoire.

5) Expliquer pourquoi il est important que java.lang.String ne soit pas mutable. -java.lang.string ne doit pas être mutable pour des raisons de sécurité et pour éviter les modifications involontaires des variables en cas de confils avec les adresses mémoires

6) Qu'affiche le code suivant ? -Le code suivant affiche "hello", la chaine de charactère ne peut pas être modifié.

Exercice 2

1) Utiliser dans un premier temps l'opérateur + qui permet la concaténation de chaînes de caractères

```
public class Morse {
    public static void main(String[] args) {
        for(String i : args){
            System.out.print(i+" Stop. ");
        }
    }
}
```

2) A quoi sert l'objet java.lang.StringBuilder ? -java.lang.StringBuilder permet des opérations sur les chaines de charactères. Pourquoi sa méthode append(String) renvoie-t-elle un objet de type StringBuilder ? -la methode append(String) renvoie un objet de type StringBuilder pour pouvoir appliquer des méthodes de cette classe à notre objet

3) Réécrire la classe Morse en utilisant un StringBuilder. Quel est l'avantage par rapport à la solution précédente ? -l'aventage est que le code ne print qu'une seule fois.

```
import java.lang.StringBuilder;

public class Morse {
    public static void main(String[] args) {
        var builder = new StringBuilder();
```

```
        for(String i : args){
            builder.append(i);
            builder.append(" Stop. ");
        }
        System.out.println(builder.toString());
    }
}
```

4) Pourquoi peut-on utiliser ' ' à la place de " " ? -On peut utiliser ' ' à la place de " " lorsque notre variable désigne un seul caractère, un char. Pour les char on utilise ' ' et pour les strings " " Compiler le code puis utiliser la commande javap pour afficher le bytecode Java (qui n'est pas un assembleur) généré. Que pouvez-vous en déduire ? -StringBuilder() permet d'éxectuer le code en faisant moins d'action

5) Compiler le code de la question 1, puis utiliser la commande javap pour afficher le bytecode Java généré. Que pouvez-vous en déduire ? -"+" nécécite au code plus d'actions que stringBuilder() Dans quel cas doit-on utiliser StringBuilder.append() plutôt que le + ? Et pourquoi est-ce que le chargé de TD va me faire les gros yeux si j'écris un + dans un appel à la méthode append? -On doit utiliser StringBuilder.append() plutot que le + quand on doit ajouter plusieurs fois des éléments à notre chaines de charactère.

Exercice 3

1) A quoi servent la classe java.util.regex.Pattern et sa méthode compile ? -java.util.regex.Pattern permet de traiter des expressions régulières. Compile prend en entrée une chaine de charactère et retourne un objet qui peut être utilisé pour effectuer des opérations sur du texte, répérer des motifs. A quoi sert la classe java.util.regex.Matcher ? -java.util.regex.Matcher permet de chercher dans l'objet retourné par "Pattern" un motif définis

2) Reprenez la petite calculatrice Calc de l'exercice 1. On veut que si l'utilisateur saisisse autre chose qu'un nombre, le programme lui demande de saisir une autre valeur.

```
import java.util.Scanner;

public class Calc {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int valueInt=0;
        int valueInt2=0;

        boolean notANumber = true;
        while(notANumber){
            String value = scanner.nextLine();
            String value2 = scanner.nextLine();
            if(value.matches("-?\\d+")){
                if(value2.matches("-?\\d+")){
                    valueInt = Integer.parseInt(value);
                    valueInt2 = Integer.parseInt(value2);
```

```
                    notANumber=false;
                }
                else{
                    System.out.println("Veuillez entrer des nombres");
                }
            }
            else{
                System.out.println("Veuillez entrer des nombres");
            }
        }

        scanner.close();
        System.out.println(valueInt+valueInt2);
        System.out.println(valueInt-valueInt2);
        System.out.println(valueInt*valueInt2);
        System.out.println(valueInt/valueInt2);
        System.out.println(valueInt%valueInt2);
    }
}
```

3) On veut écrire un programme qui va lire un fichier HTML dont le chemin est pris sur la ligne de commande et qui va afficher la liste des URLs de tous les liens contenus dans le fichier

4) Modifiez le programme pour ne conserver que les liens contenant le mot java ou Java.

5) Si vous vous embêtez pendant le week-end, modifiez votre programme pour gérer des liens contenant plusieurs balises