Annexe 7B - les expressions régulières ( exercices de base + Pattern, Matcher )

## 1. Vrai ou Faux ?

**A) String un = ”allo”;**

**un.matches(”\\d{4}”);**

**faux**

**B) String deux = " dede";**

**deux.matches("\\s{1,}[a-e]{2,5}");**

**vrai**

**C) String trois = "343---";**

**trois.matches("...\\w\\w\\w");**

**faux**

**D) String quatre = "2222";**

**quatre.matches("22");**

**faux**

**E) String cinq = "sage";**

**cinq.matches("\\s.{3}");**

**false**

**F) String six = "éric";**

**six.matches("\\w{4}");**

**false a cause du e accent**

**G) String six = "éric";**

**System.out.println ( six.matches("[\\wé]{4}"));**

**vrai**

H) Une expression régulière permettant de représenter tout matricule du collège ?

\\d{7}

I) Une expression régulière pouvant représenter n'importe quel nombre entier positif?

[1-9][0-9]{0,} ou [1-9][0-9]\*

positif, négatif ou nul ? [-]? Ou -? Ou -?[1-9]\\d\*|0

J) Une expression régulière pouvant contenir une lettre majuscule suivie de deux astérisques ? [A-Z]\\\*{2}

## 2. Classes Pattern, Matcher

La méthode matches de la classe String imite en tous points celle de la classe Pattern ( package java.util.regex ). En effet, on peut également créer des expressions régulières en créant un objet Pattern. Cela permet d’accéder à de nouvelles méthodes…

**Les trois principales classes du package javax.util.regex :**

* Pattern :
* La classe Pattern représente une version compilée d’une expression régulière
* Elle n’a pas de constructeur; on crée un objet Pattern à l’aide de la méthode statique compile :

Pattern p = Pattern.compile (« \\d{5} »);

* Matcher :
* La classe Matcher permet d’obtenir un objet qui interprétera le Pattern ( L’expression régulière ) et d’y appliquer des méthodes
* Comme Pattern, on ne peut pas créer un objet Matcher avec un constructeur, on doit utiliser la méthode matcher de la classe Pattern :

Matcher m = p.matcher(«123456666633 »);

* méthodes à utiliser sur le Matcher : matches, find, reset

m.matches() -> faux car 123456666633 ne correspond pas a 5 chirffres

m.find()-> vrai car tu peux trouver 5 chiffres dans 123456666633

m.find()->vrai car tu peux trouver 5 chriffres a la suite de l’autre 5 chriffres

m.find()-> false car il ne reste qu’un chiffre

m.reset - > retourne au début

* PatternSyntaxException
* Lancée lorsque la syntaxe de l’expression régulière n’est pas correcte

## 3. autres méthodes pour travailler avec une chaîne de caractères :

* méthode split de la classe String :

String[] tab= "asdf4pa4osi".split("\\d");

for ( int i = 0; i < tab.length; i++ )

System.out.println(tab[i]);

Sa va separer la chaine de carachtere dans un tableau de string selon les chiffres et fait de plus petit string

asdf~~4~~pa~~4~~osi, alors tab[0] = asdf

tab[1] = pa

tab[2] = osi

* classe Scanner ( package java.util )

* permet de « scanner » tout un fichier plutôt que seulement une String
* permet de retourner tout type prédéfini plutôt que seulement des Strings
* délimiteur par défaut : un caractère blanc ( espace, \r, \n, d'autres ) mais on peut utiliser une expression régulière à la place