

PRG (ETS d'Enginyeria Informàtica) - Curs 2019-2020
Pràctica 2. Resolució d'alguns problemes amb recursió

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València



Índex

1	Context i treball previ	1
2	Problema A. <i>Prefixe</i>	2
3	Problema B. <i>Subcadena</i>	3
4	Avaluació	4

1 Context i treball previ

En aquesta pràctica es proposa la resolució de forma recursiva de dos problemes amb **Strings**. Per a això, es dissenyaran els mètodes corresponents i les classes de prova per assegurar que les solucions dels problemes siguin correctes.

És convenient haver estudiat la secció 10.6 *Recursividad con objetos de tipo String* de la 3^a edició del llibre de l'assignatura¹ i haver comprès alguns exemples com el problema de comptar el nombre de caràcters 'a' en certa **String** s.

Activitat 1: Creació del paquet BlueJ pract2

Obre el projecte *BlueJ* de treball de l'assignatura (**prg**) i crea un nou paquet **pract2**. Agrega al paquet el fitxer **PRGString.java** que hauràs descarregat prèviament de la carpeta **Recursos/Laboratorio/Práctica 2** de la PoliformaT de PRG. La classe **PRGString** és una classe d'utilitats que inclou els mètodes que resolen el problema de comptar el nombre de 'a's en una **String** s (secció 10.6 del llibre) i els mètodes (a completar) que resolen els problemes que se't plantegen a continuació.

¹Si tens la 2^a edició: secció 11.6.

2 Problema A. *Prefixe*

Donades dues `Strings` `a` i `b`, potencialment buides, es diu que `a` és *prefixe* de `b` quan tots els caràcters de `a` estan consecutius, en el mateix ordre original, al començament de `b`.

Conseqüència de la definició anterior és que la *cadena buida* és prefixe de qualsevol altra, fins i tot si aquesta altra també estigués buida. Nota, d'altra banda, que una cadena no pot ser prefixe d'una altra si la primera és de longitud més gran que la segona.

Activitat 2: mètode `isPrefix(String, String)`

Defineix recursivament un mètode `isPrefix(String, String)` per comprovar si una cadena és prefixe d'una altra. Per a això:

- Estableix els casos base i general de la recursió definint, a més, la solució del problema en cadascun d'aquests casos. La capçalera del mètode (en la que no hi ha paràmetres posicionals) haurà de ser necessàriament la que segueix:

```
public static boolean isPrefix(String a, String b)
```

- Documenta adequadament el mètode, explicitant quins són els seus paràmetres, el tipus del seu resultat i, cas d'haver-ne, la seua precondició.
- Comprova que el codi del mètode segueix les normes d'estil usant el *Checkstyle* de *BlueJ* i corregeix-lo si no és el cas.

Activitat 3: validació del mètode `isPrefix(String, String)`

Escriu una classe programa `TestIsPrefix` que permeti executar el mètode amb diferents dades per tal de comprovar que no hi ha errors d'execució i que el resultat que torna en cada cas és el correcte.

Les dades a provar han de reflectir les diferents situacions que es poden donar en l'execució del mètode, com ara, per exemple: que ambdues cadenes estiguen buides, que ho estiga només una, que la primera cadena siga més llarga que la segona, que la primera cadena siga prefixe o no de la segona, etc. A la taula següent es detallen els diferents casos, amb instàncies concretes i el resultat esperat per a cada cas.

Cas	a	b	Resultat
a i b buides	""	""	true
Només a buida	""	"recursion"	true
Només b buida	"recursion"	""	false
a de major longitud que b	"recursion"	"rec"	false
a i b de la mateixa longitud i a és prefixe de b	"recursion"	"recursion"	true
a i b de la mateixa longitud i a no és prefixe de b	"123456789"	"recursion"	false
a de menor longitud que b i a és prefixe de b	"rec"	"recursion"	true
a de menor longitud que b i a no és prefixe de b:			
- pel primer caràcter	"pecur"	"recursion"	false
- pel darrer caràcter	"recurso"	"recursion"	false
- per un caràcter intermedi	"remursi"	"recursion"	false

La classe `TestIsPrefix` ha d'incloure un mètode amb el següent perfil:

```
private static void testIsPrefix(String a, String b)
```

que mostre per pantalla les `Strings` de prova, el resultat del teu mètode `isPrefix(String, String)` i el resultat esperat. Per a això últim, pots utilitzar el mètode `startsWith(String)` de la classe `String`.

El `main` ha d'invocar al mètode `testIsPrefix(String, String)` per a cada cas de prova. Pots definir un array de `String` per emmagatzemar les diferents instàncies dels casos a provar.

3 Problema B. *Subcadena*

Donades dues `Strings` `a` i `b`, potencialment buides, es diu que `a` és *subcadena* de `b` quan tots els caràcters de `a` estan consecutius, en el mateix ordre original, en algun lloc de `b`. O, el que és el mateix, quan `a` és prefixe de `b` o d'alguna de les possibles subcadenaes de `b`.

Naturalment, com passava en el cas de `isPrefix(String, String)`, es pot veure que la *cadena buida* és subcadena de qualsevol altra, encara que aquella altra també estigüés buida. A més, una cadena no pot ser subcadena d'una altra si la primera és de longitud més gran que la segona.

Activitat 4: mètode `isSubstring(String, String)`

Defineix recursivament, en termes de `isPrefix(String, String)`, el mètode `isSubstring(String)`, per tal de poder comprovar si una cadena és subcadena d'una altra. Per a això:

- Enuncia els casos base i general de la recursió, definint la solució del problema en cada cas. La capçalera del mètode ha de ser necessàriament:

```
public static boolean isSubstring(String a, String b)
```

Nota que, igual que per a l'operació `isPrefix(String, String)`, no hi ha paràmetres posicionals en la capçalera anterior.

- Documenta adequadament el mètode, explicitant quins són els seus paràmetres, el tipus del seu resultat i, cas d'haver-ne, la seua precondition.
- Comprova que el codi del mètode segueix les normes d'estil usant el *Checkstyle* de *BlueJ* i corregeix-lo si no és el cas.

Activitat 5: validació del mètode `isSubstring(String, String)`

Escriu una classe programa `TestIsSubstring` que permeti executar el mètode amb diferents dades per tal de comprovar que no hi ha errors d'execució i que el resultat que torna en cada cas és el correcte. Com abans, has d'identificar les diferents situacions que es poden donar, provant que, en totes elles, el mètode funciona adequadament. En aquest cas, pots comparar el resultat amb el del mètode `contains(String)` de la classe `String`.

4 Avaluació

Aquesta pràctica forma part del primer bloc de pràctiques de l'assignatura que serà avaluat en el primer parcial d'aquesta. El valor d'eixe bloc és d'un 50% respecte al total de les pràctiques. El valor percentual de les pràctiques en l'assignatura és d'un 25% de la nota final.