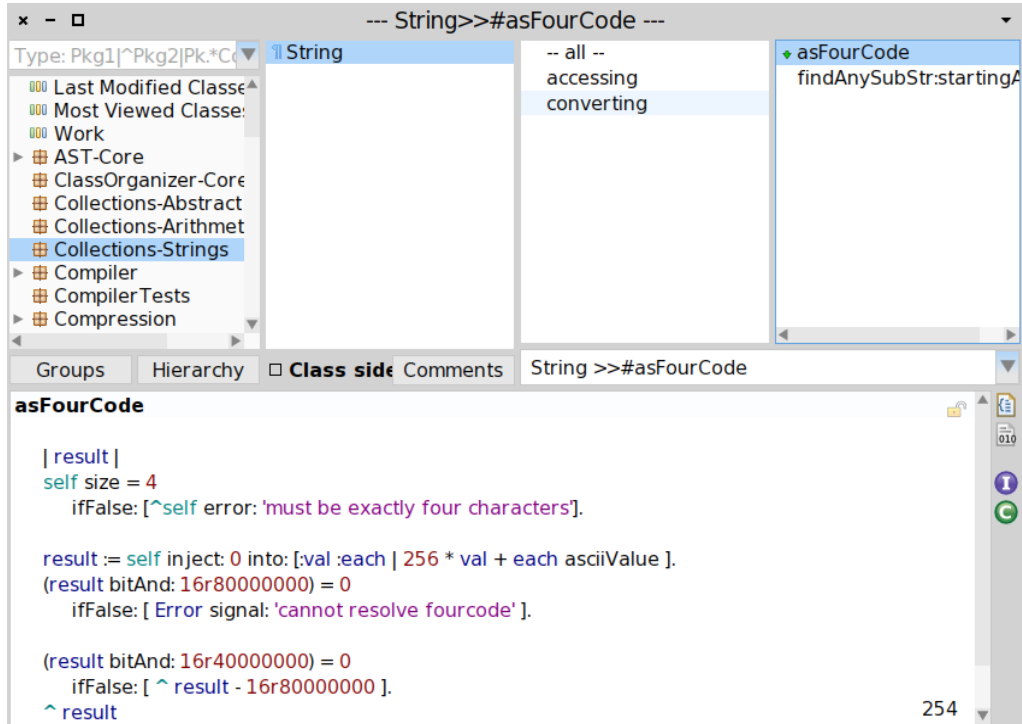


## Exercicis de Reflexió amb Smalltalk

1.- Fes un programa (per ser executat al *workspace*) que, donat un símbol que representarà el selector d'un mètode, generi un *browser* amb tots els mètodes que l'utilitzen dins el seu codi font. Per exemple, si el símbol fos `#inject:into:` el resultat hauria de ser:



És a dir, un *browser* només amb els mètodes (i classes i paquets corresponents) que fan servir `#inject:into:`.

2.- Per quina raó s'avalua a **true** la següent expressió?

**Class class class class == Class class class class class class**

3.- Els mètodes `#new` i `#new:` són mètodes d'instància de la classe **Behavior**, tot i que és habitual que se'ls redefeixi en altres classes. Malgrat són mètodes d'instància de **Behavior**, usualment es redefeixen com a mètodes de classe. Per exemple, utilitzem **Array new: 5** per crear instàncies de la classe **Array**, i el missatge `#new:` l'estem enviant a la classe **Array**. Les redefinicions, doncs, s'acostumen a fer en el *Class side*. Això aparentment viola la "regla" que diu que en l'herència els mètodes d'instància s'hereten en l'*instance side* i els mètodes de classe s'hereten en el *Class side*. Expliqueu per quina raó no hi ha res d'incorrecte en el fet de redefinir `#new` i `#new:` en el *Class side*.

4.- Expliqueu com un bloc pot cridar-se a ell mateix (aconseguint així blocs sense nom, anònims, però recursius). *No podeu suposar que el bloc ha estat assignat a una variable.* Feu-ho servir per fer un bloc que calculi el factorial.

5.- Escriviu un mètode **find: aString** que, *enviat a una classe*, retorni una col·lecció de selectors tal que els mètodes corresponents contenen **aString** dins del seu codi font.

*Exemple:* El resultat d'executar al *workspace* **Object find: 'reflect'** hauria de ser una col·lecció amb els selectors de tots aquells mètodes d'**Object** tals que la string 'reflect' apareix en el seu codi font. En aquest cas obtindriem:

```
#(#perform:withArguments:inSuperclass:    #perform:with:
#perform:withArguments:    #perform:with:with:    #perform:
#perform:with:with:with:)
```

A quina classe cal posar aquest mètode per a que *qualsevol* classe o metaclasses del sistema sigui capaç d'executar-lo? I on el posem, a l'*instance side* o al *Class side*?

6.- Escriviu un fragment de codi (per ser executat en el *workspace*) que ens digui quants mètodes tenen la **String 'this'** en el seu codi font.

7.- A classe vam veure un mètode anomenat **#haltIf: aSymbol** que permetia fer un *halt* només quan a la cadena de crides hi havia un mètode amb **aSymbol** com a nom. Feu ara un mètode **#haltIfTrue: aBlock** on **aBlock** és un bloc sense paràmetres. Aleshores, **self haltIfTrue: [...]** s'aturarà només si el resultat d'avaluar [...] és **true**. A quina classe ha d'anar aquest mètode?

8.- a) Què fa aquest codi?

```
#aa bindTo: 1 in: [
    Transcript show: #aa binding asString.
    #aa bindTo: 2 in: [
        Transcript show: #aa binding asString. ].
    Transcript show: #aa binding asString , ' again'. ].
```

b) I aquest codi?

```
#aa bindTo: '1' in: [
    #bb bindTo: 'a' in: [
        Transcript show: #aa binding , '-' , #bb binding.
    #aa bindTo: '2' in: [
        #bb bindTo: 'b' in: [
            Transcript show: #aa binding , '-' , #bb binding.].
        Transcript show: #aa binding , '-' , #bb binding. ].
    #bb bindTo: 'b' in: [
        Transcript show: #aa binding , '-' , #bb binding ]]]
```