

Estructuras de datos y algoritmos

Proyecto de programación

Preliminares

Este proyecto pretende promover la tarea de diseño y programación durante todo el curso. El desarrollo del proyecto obligará, seguramente, a retomar decisiones de diseño y programación; eso forma parte de las actividades del curso. Todo el proceso de gestión, diseño y programación deberá quedar documentado en una *Memoria del proyecto*, que habrá que ir redactando durante todo el curso y que servirá de registro de la tarea realizada y para la evaluación de la misma en cualquier momento. La Memoria del proyecto deberá redactarse siguiendo una estructura que se indicará en otro documento.

Enunciado del proyecto

La aplicación a desarrollar deberá gestionar *sentencias* simples que tendrán la forma de *sujeto propiedad objeto* (y terminadas con un espacio en blanco y un punto); por ejemplo:

estudiante2013 matriculadoEn upv-ehu .
estudiante2013 es Estudiante .
upv-ehu es Universidad .
estudiante2013 cursa asig20 .
profesor7 encargadoDe asig20 .
asig20 esDeNivel segundo .
asig20 es Asignatura .

En la primera sentencia enunciada el sujeto es *estudiante2013*, la propiedad es *matriculadoEn* y el objeto es *upv-ehu*. Los sujetos, los objetos y las propiedades son entidades. Las propiedades, en cada sentencia, relacionan una entidad con otra. Cada entidad tiene un identificador único. En el ejemplo, las sentencias primera, segunda y cuarta tienen el mismo sujeto, el sujeto de la tercera sentencia es el mismo que el objeto de la primera sentencia y aparecen cinco propiedades distintas en esas siete sentencias. Una propiedad puede ser también sujeto u objeto de una sentencia.

Deberá ser posible crear y almacenar sentencias, retirar sentencias de un almacén, recolectar todas las sentencias que tengan un sujeto determinado y, así, toda una colección de operaciones que se consideren interesantes para la aplicación.

Al menos, deberán implementarse las siguientes operaciones:

1. Sobre un almacén: Colección de sentencias del almacén que tienen un sujeto determinado, dado como parámetro.
2. Sobre un almacén: Colección de sentencias distintas del almacén que tienen un sujeto determinado, dado como parámetro.
3. Sobre un almacén: Colección de propiedades distintas que aparecen en las sentencias del almacén.

4. Sobre un almacén: Colección de entidades distintas que son sujeto de alguna sentencia y también son objeto de alguna sentencia de ese almacén.

5. Dada una colección de almacenes: Colección de entidades distintas que son sujetos en todos y cada uno de esos almacenes.

Observación: Aquí será importante la consideración de distintos algoritmos porque es muy relevante para la eficiencia. Por ejemplo, ¿en qué orden conviene calcular las intersecciones?

6. Sobre un almacén: Colección ordenada de todas las sentencias que aparecen en el almacén.

La colección debe quedar ordenada según la relación de orden siguiente:

Dadas dos sentencias $(s1, p1, o1)$ y $(s2, p2, o2)$,
 $(s1, p1, o1) < (s2, p2, o2)$ si $s1 < s2$, o bien
 $s1 = s2$ y $p1 < p2$, o bien
 $s1 = s2$ y $p1 = p2$ y $o1 < o2$

La relación de orden en la clase que represente entidades y en la clase que represente propiedades se implementará según el orden lexicográfico (el orden del diccionario) de los string que se utilicen para representar sus atributos.

7. Hay varias propiedades especiales. Una de ellas es la propiedad *es*, que indica la clase de entidad que es el sujeto de la sentencia; por ejemplo, *upv-ehu es Universidad*. Una entidad puede ser de varias clases distintas; por ejemplo, *estudiante2013 es Estudiante*, *estudiante2013 es Deportista* y *estudiante2013 es Coleccionista*.

Otra propiedad especial es *subClaseDe*, que representa que todos los individuos de la clase del sujeto son también individuos de la clase del objeto; por ejemplo, *Estudiante subClaseDe Persona*.

La propiedad *subClaseDe* es transitiva. Es decir:

Si $C \text{ subClaseDe } D$ y $D \text{ subClaseDe } E$ entonces $C \text{ subClaseDe } E$

Por lo tanto, si además aparece la sentencia *Persona subClaseDe SerVivo*, entonces toda entidad de la clase *Estudiante* es *Persona* y es *SerVivo*.

La noción de *superClaseDe* es la inversa de *subClaseDe*. Es decir: si $C \text{ subClaseDe } D$ entonces $D \text{ superClaseDe } C$.

Queremos disponer de los métodos siguientes:

- a) Sobre un almacén: Colección de las clases de un sujeto, dado como parámetro, en ese almacén.
 - b) Sobre un almacén: Colección de las clases que son superclase de una clase, dada como parámetro, en ese almacén.
8. Sobre un almacén: Colección de entidades que son de una determinada clase, dada como parámetro, en ese almacén.
9. Hay que incluir métodos que permitan:
 - a) Cargar sentencias en un almacén, tomadas de archivos de texto de nuestro directorio.

```
public AlmacenDeSentencias cargar (String nombreDeArchivo);
```

b) Descargar las sentencias de un almacén en un archivo de texto de nuestro directorio.

```
public void descargar (String nombreDeArchivo);
```

10. Sobre un almacén: Colección de estudiantes distintos que cursan alguna asignatura de la que es encargado un determinado profesor, dado como parámetro.
11. Sobre un almacén: Colección de profesores distintos que trabajan para algún departamento de una universidad, dada como parámetro.

Observaciones

Los ficheros de datos que vamos a manejar para las pruebas están generados de manera artificial, pero sus datos simulan datos reales. En particular, la correspondencia entre los identificadores (strings) de las propiedades citadas en el enunciado y las cadenas de caracteres (strings) que aparecen en las sentencias de los ficheros de prueba es la siguiente:

Propiedad	Identificador
<i>es</i>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<i>subClaseDe</i>	<http://www.w3.org/2000/01/rdf-schema#subClassOf>
<i>cursa</i>	<http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#takesCourse>
<i>encargadoDe</i>	<http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#teacherOf>
<i>departamentoDe</i>	<http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#subOrganizationOf>
<i>trabajaPara</i>	<http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#worksFor>