Examen Final: Laboratorio 2

Paradigmas de la Programación - 25/02/2026

Generales

- Para codear el parcial solo podrán utilizar algún editor de texto plano o IDE sin asistencia de IA. No utilizar teléfonos durante el examen. No acceder a sitios que permitan comunicarse con otras personas como whatsapp, discord, zulip, etc. **No respetar esta regla implica desaprobación inmediata del final y será calificada con 0 en el acta del examen.**
- Pueden utilizar sus propias laptops o alguna máquina del laboratorio que tenga instalado todo lo necesario para el lab2 (openjdk v21, etc).
- Pueden tener el código fuente del laboratorio 2 para consulta pero **NO** deben entregarlo. Solo entregar el código con la solución del examen. Resta puntos entregar el código de los laboratorios.
- **Verificar que se haya recibido el correo de confirmación** luego de la entrega**.** Si no llega, deben avisar antes de retirarse del aula. **No se aceptarán reclamos posteriores por entregas incorrectas o incompletas.**

El examen consiste en ir agregando con cada ejercicio, una nueva funcionalidad a nuestro FeedReader. Pero al ser un examen de Programación Orientada a Objetos (POO), no solo es importante lograr la funcionalidad que se requiere (condición necesaria pero no suficiente) sino cómo modelan su solución a través de los conceptos del paradigma (clases, herencia, interfaces, modificadores de scope, etc). Es decir, es importante la calidad de su solución.

Esqueleto

Descargar del siguiente link:

https://drive.google.com/file/d/1lfmvpicxv-gOmicIqXplgc_8tsjTVrrH/view?usp=sharing

Si al descomprimir se encuentran con archivos que comienzan con “._” deben borrarlos a todos, incluso en sub-directorio. Piensen y busquen un script que lo haga por ustedes

Otras librerías que pueden ser útiles

- json-20231013.jar
- jsoup-1.21.1.jar

Instrucciones para Compilar y Correr

Para compilar y ejecutar el código del examen, solo es necesario utilizar el Makefile provisto desde la carpeta del esqueleto:

```
None
$ make run ARGS="-h quick"
```

El resultado es similar a:

```
************* FeedReader version 2.0 *************

================================================
Feed: http://rss.cnn.com/rss/cnn_topstories.rss
================================================
Top 5 Named Entities by Frequency:
US (OTHER): 10
Tuesday (OTHER): 8
Heres (OTHER): 5
Dominion (OTHER): 5
GOP (OTHER): 4

================================================
Feed: https://feeds.a.dj.com/rss/RSSWorldNews.xml
================================================
Top 5 Named Entities by Frequency:
Gaza (OTHER): 6
Israeli (OTHER): 4
Hamas (OTHER): 4
Hostages (OTHER): 4
Oct (OTHER): 2

================================================
Feed: https://feeds.bloomberg.com/markets/news.rss
================================================
Top 5 Named Entities by Frequency:
US (OTHER): 12
Donald (OTHER): 11
President (OTHER): 8
Trumps (OTHER): 7
Supreme (OTHER): 6
```

Notar que el esqueleto del examen difiere del código del laboratorio en varios puntos. En particular:

- Las entidades nombradas se cuentan por feed, no globalmente.
- Para asignar a cada palabra una categoría de entidad nombrada se utiliza el método Heuristic.getCategory. Si la palabra no es una entidad nombrada, entonces devuelve null.
- Para obtener las entidades nombradas de un artículo, se utiliza el método Heuristic.getNamedEntities

News Topic Classifier

El objetivo del examen es convertir el FeedReader en un clasificador temático de noticias. El sistema debe ser capaz de identificar si una suscripción pertenece a

secciones como Economía, Política, Tecnología o Mundo, basándose en un diccionario desarrollado por expertos.

Para ello, simplemente va a clasificar cada palabra dentro de un tema (por ejemplo la palabra “pelota” se clasifica como “Deporte”), y luego va a asignar a la suscripción la clase que más palabras tenga.

Ejercicio 1. Definición de las clases

En nuestro sistema tendremos un número limitado de clases que serán compartidas entre todas las Heurísticas. En este primer ejercicio debemos

1.a) Hardcodear la siguiente lista de clases (puede elegir otra estructura de datos):

ECONOMY **POLITICS** **TECH** **WORLD** **OTHER**

1.b) Agregar chequeos que aseguren que ninguna palabra se asigna a una clase incorrecta. Por ejemplo, si el método Heuristic.getCategory devuelve **HEALTH,** el sistema debería ignorar la entidad nombrada. En este punto ustedes deben elegir cómo implementan esos chequeos.

1.c) Como segundo punto, debe justificar EN EL FORMULARIO DE ENTREGA por qué resolvió el ejercicio de la manera que lo hizo, utilizando conceptos de POO.

Ejercicio 2. Clasificación Temática

Agregar un método para clasificar cada palabra en una categoría, utilizando el diccionario de valores definido al final de la consigna.

Sólo las palabras que no están en `Heuristic.stopWords` deben ser clasificadas.

Este método debe utilizarse en lugar de la heurística Quick si se selecciona con la opción -h topic al ejecutar el programa.

Salida esperada:

```
└─$ make run ARGS="-h topic"
mkdir -p bin
javac -d bin -cp "lib/jsoup-1.21.1.jar:lib/json-202
tion/SubscriptionParser.java src/subscription/Subsc
 src/feedParser/FeedParser.java src/namedEntity/heu
/heuristic/TopicHeuristic.java src/namedEntity/heur
java -cp "bin:lib/jsoup-1.21.1.jar:lib/json-2023101
************ FeedReader version 2.0 ************

================================================
Feed: http://rss.cnn.com/rss/cnn_topstories.rss
================================================
Top 5 Named Entities by Frequency:
election (POLITICS): 2
business (ECONOMY): 2
AI (TECHNOLOGY): 2
technology (TECHNOLOGY): 2
economy (ECONOMY): 2

================================================
Feed: https://feeds.a.dj.com/rss/RSSWorldNews.xml
================================================
Top 2 Named Entities by Frequency:
president (POLITICS): 2
minister (POLITICS): 1

================================================
Feed: https://feeds.bloomberg.com/markets/news.rss
================================================
Top 5 Named Entities by Frequency:
President (POLITICS): 8
market (ECONOMY): 4
Market (ECONOMY): 3
Trade (ECONOMY): 3
trade (ECONOMY): 3
```

Ejercicio 3. Clasificación de feeds

En este ejercicio debemos clasificar cada uno de los feed entre las categorías válidas que definimos en el ejercicio 1. Pueden implementarlo como quieran, pero una manera es recorrer la lista de categorias y devolver la que tenga más entidades.

Ejercicio 4: Leer stopwords a partir de archivo .json (Solo Alumnos Libres)

En lugar de hardcodear el mapa, leer el archivo `config/stopwords.json`. Se debe utilizar la librería `org.json` (proporcionada en `lib/`) para parsear el archivo.

Formato del archivo

```JSON
["i", "me", "my", "myself", "we", "our", "ours", "ourselves",
"you", "yours", "yourself", "yourselves" ...]
```

Entrega

Comprimir el directorio raíz de su examen en un archivo “tar.xz” con el siguiente formato “Apellido_Nombre.tar.xz”. Para que quede con el formato adecuado, usar los siguientes comandos:

```Shell
mv skeleton Juarez_MatiasAlfonso

tar -czvf Juarez_MatiasAlfonso.tar.xz Juarez_MatiasAlfonso
```

Adjuntar el comprimido en el formulario de entrega, llenar todos los campos del mismo y submitear con su correo institucional. Por último, chequear en su bandeja de entrada si recibieron el mail de confirmación de entrega realizada, gracias.

Diccionario de clases por palabra

Java

```java
  private static final Map<String, String> dictionary = new HashMap<>();


   static {
       // TECHNOLOGY
       dictionary.put("software", "TECHNOLOGY");
       dictionary.put("hardware", "TECHNOLOGY");
       dictionary.put("computer", "TECHNOLOGY");
       dictionary.put("internet", "TECHNOLOGY");
       dictionary.put("algorithm", "TECHNOLOGY");
       dictionary.put("database", "TECHNOLOGY");
       dictionary.put("server", "TECHNOLOGY");
       dictionary.put("cloud", "TECHNOLOGY");
       dictionary.put("ai", "TECHNOLOGY");
       dictionary.put("robot", "TECHNOLOGY");
       dictionary.put("cybersecurity", "TECHNOLOGY");
       dictionary.put("blockchain", "TECHNOLOGY");
       dictionary.put("startup", "TECHNOLOGY");


       // POLITICS
       dictionary.put("government", "POLITICS");
       dictionary.put("election", "POLITICS");
       dictionary.put("president", "POLITICS");
       dictionary.put("senate", "POLITICS");
       dictionary.put("congress", "POLITICS");
       dictionary.put("minister", "POLITICS");
       dictionary.put("policy", "POLITICS");
       dictionary.put("democracy", "POLITICS");
       dictionary.put("law", "POLITICS");
       dictionary.put("campaign", "POLITICS");
       dictionary.put("vote", "POLITICS");
       dictionary.put("parliament", "POLITICS");


       // SPORTS
       dictionary.put("football", "SPORTS");
       dictionary.put("soccer", "SPORTS");
       dictionary.put("basketball", "SPORTS");
       dictionary.put("tennis", "SPORTS");
       dictionary.put("goal", "SPORTS");
       dictionary.put("match", "SPORTS");
       dictionary.put("tournament", "SPORTS");
       dictionary.put("league", "SPORTS");
```

```
dictionary.put("coach", "SPORTS");
dictionary.put("player", "SPORTS");
dictionary.put("championship", "SPORTS");
dictionary.put("stadium", "SPORTS");


// ECONOMY
dictionary.put("economy", "ECONOMY");
dictionary.put("inflation", "ECONOMY");
dictionary.put("market", "ECONOMY");
dictionary.put("stock", "ECONOMY");
dictionary.put("investment", "ECONOMY");
dictionary.put("bank", "ECONOMY");
dictionary.put("finance", "ECONOMY");
dictionary.put("currency", "ECONOMY");
dictionary.put("trade", "ECONOMY");
dictionary.put("recession", "ECONOMY");
dictionary.put("tax", "ECONOMY");
dictionary.put("budget", "ECONOMY");


// ENVIRONMENT
dictionary.put("climate", "ENVIRONMENT");
dictionary.put("pollution", "ENVIRONMENT");
dictionary.put("sustainability", "ENVIRONMENT");
dictionary.put("recycling", "ENVIRONMENT");
dictionary.put("wildlife", "ENVIRONMENT");
dictionary.put("forest", "ENVIRONMENT");
dictionary.put("carbon", "ENVIRONMENT");
dictionary.put("emissions", "ENVIRONMENT");
dictionary.put("biodiversity", "ENVIRONMENT");
dictionary.put("renewable", "ENVIRONMENT");
dictionary.put("ecosystem", "ENVIRONMENT");
dictionary.put("drought", "ENVIRONMENT");
```