examen-1.pdf

ml_gj

Tecnologias y Paradigmas de la Programacion

2º Grado en Ingeniería Informática del Software

Escuela de Ingeniería Informática
Universidad de Oviedo

Tecnologías y Paradigmas de la Programación. Examen práctico L6. 29/04/2020 - 2019/2020

NOTA: Se incluye un fichero txt con un método para generar arrays de palabras con letras aleatorias y longitud variable en un intervalo pasado como parámetro. En los siguientes ejercicios, puede considerarse que si un carácter no es una vocal, es una consonante. No es necesario considerar en el código el uso de tildes y pueden tratarse como consonantes.

Ejercicio 1. Impleméntese un método Vocales() extensor de la clase String que devuelva la distribución de vocales en una lista (empleando la de .NET) **(1,00 puntos)**. Por tanto, "Mensajero".Vocales() devolverá una lista con 5 entradas (una por vocal) indicando cuántas veces se repite cada vocal.
Sin borrar el código anterior, implemente una funcionalidad similar empleando cláusulas:

```
public static Func<int> ClausulaVocales(string cadena).
```

Cada vez que se invoque a la función devuelta, irá devolviendo un entero con la distribución de la vocal actual (se irán devolviendo en orden: a, e, i, o, u; y vuelta a empezar). Por tanto, si probamos con ClasulaVocales("Mensajero") e invocamos 5 veces a la función devuelta: obtendremos los valores: 1, 2, 0, 1, 0 **(1,50 puntos)**.

Ejercicio 2. Impleméntese una función de orden superior genérica que reciba (**1,75 puntos**):

- Un IEnumerable de Predicados
- Un IEnumerable de elementos

Devolverá un IDictionary con la siguiente estructura:

- Como claves utilizará el número de predicados incumplidos.
- Para cada entrada tendrá una IList con los elementos que han incumplido ese total de predicados.

A través de un proyecto de test, pruébese la funcionalidad del método empleando un array de 1000 palabras de longitud 2-4 caracteres y usando de predicados: uno para saber si la longitud de la palabra es impar y otro para saber si la longitud de la palabra es igual que 4 **(0,75 puntos)**.

Ejercicio 3. Empleando la paralelización de tareas en TPL y a partir de una lista de 10000 palabras con longitudes entre 3 y 8 caracteres, obtener en dos variables de tipo long **(2,00 puntos)**:

- Número de palabras de longitud impar que empiezan por consonante.
- Número de palabras de longitud par que empiezan por vocal.

Imprímase por pantalla el valor de las variables anteriores. Imprímase por pantalla el número de hilos utilizados por TPL para resolver el cálculo **(0,50 puntos)**.

Ejercicio 4. Empleando el esquema Master-Worker, impleméntese la funcionalidad para almacenar, en la cola implementada en clase, palabras que tengan más consonantes que vocales. El Master pasará la misma cola a todos los workers en su constructor **(1,25 puntos)**. Mídase, grafíquese y explíquese el context switching del ejercicio de 1 a 20 hilos **(1,25 puntos).**

En la evaluación de los ejercicios se valorará el uso adecuado de los conocimientos que debe haber adquirido el alumno durante los laboratorios prácticos y sus tareas. Debe entregarse todo el código necesario para que funcionen correctamente los ejercicios planteados. Aquellos ejercicios que no funcionen, no realicen exactamente lo que se pide o no se prueben tal y como se pide, tendrán una puntuación igual a cero. Si el examen no se entrega o se entrega en blanco, la puntuación total del examen será de cero.