EXAMEN TIPO B

Sigue las intrucciones que marca cada ejercicio, no solo para la realización del mismo sino también para la entrega de los mismos.

No se permite ningun documento de ayuda, excepto los aportados por el profesor

No se corregirá ningun ejercicio que tenga errores de compilación

Crea una carpeta denominada examen, creala tambien como repositorio local de *git* y crea un archivo *REAME.md* que contenga como etiquetas tu nombre y apellidos. Crea un archivo *.gitignore* para que *git* ignore los archivos binarios de java y el directorio de documentación denominado *doc*

Crea un repositorio en tu *GitHub* denominado *examenPrimeraEvaluacion* y sincronízalo con el repositorio local

1. (5 ptos) Realiza el siguiente programa de cálculo numérico -denominado *Numero.java*- en el cual el programa reciba mediante la clase *Scanner* dos números enteros. Esto se hará en el método *main*.

   - Se debe comprobar que ambos números son mayores que cero y menores de 1000. En caso contrario el programa terminará indicando en pantalla que los numeros introducidos no son válidos.
   - Mediante un método nos devolverá cuál de los dos números es mayor. En caso que sean iguales nos devolverá cualquiera de los números.
   - Mediante un método mostraremos en pantalla los primeros diez primeros múltiplos de tres de uno de los números que se han leído en el *Scanner*, da igual que número uses, solamente tienes que hacer la llamada en el método main con el número que tu quieras.
   - Un método que nos diga si es capicúa alguno de los números leídos, una posibilidad es convertir el número en String y darle la vuelta, posteriormente convertirlo a entero con la clase Integer y comprobar que los número son iguales. Puedes usar cualquier otro algoritmo que se te ocurra.

   Desglose de la puntuación:

   (a) (0.5 ptos) Si lees los número con el *Scanner*. En el caso que no sabes hacerlo, introdúcelos directamente en el programa y esta parte queda sin valorar.

   (b) (1 pto) Por la comprobación y terminación del programa en el caso que los números solicitado no corresponda con las especificaciones.

   (c) (0.5 ptos) Por el métodos que nos da el número mas grande.

   (d) (0.5 ptos) Por el métodos que nos da los múltiplos

   (e) (1 pto) Por el método que nos dice si es capicúa o no

   (f) (0.5 ptos) Si usas *printf* para mostrar las salidas en consola.

(g) (1 pto) Realiza la documentación de la clase que incluya las etiquetas *author* y *version*, mas la documentación de cada uno de los métodos, todo ello en una carpeta denomina *doc*. No olvides hacer la documentación de la clase después de la sentencia *import* de la clase *Scanner*

2. (4 ptos) Realiza un programa -denominado *Cadena.java*-que realiza lo siguiente: (apoyate en los metodos de la clase *String*).

   - Nos debe mostrar la cadena en mayúscula y minúscula, ejemplo (Cadena : MAYUSCULAminuscula).
   - Recorriendo los caracteres de la cadena, nos diga cuantas vocales acentuadas tiene.
   - Nos diga si acaba en consonante.
   - Nos diga si la cadena empieza o acaba por vocal.
   - Nos muestre la cadena, donde reemplacemos la vocal *o* por el número 1 y la la vocal *a* por el número 2

   Desglose de la puntuación:

   (a) (0.5 ptos) Si mostramos la concatenación de la cadena en mayúscula y minúscula.

   (b) (1 pto) Por contar el número de vocales acentuadas.

   (c) (0.5 ptos) Si nos dice que acaba en consonante.

   (d) (0.5 ptos) Si nos dice que empieza o acaba en vocal

   (e) (0.5 ptos) Si hacemos correctamente los reemplazos

   (f) (1 pto) Si usas *printf* para mostrar en pantalla los resultados, siempre debe aparecer la cadena original, ejemplo *La cadena hormiga ¿acaba en consonante? false*

3. (1 ptos) GitHub Crea los siguientes commmits:

   - Un *commit* denominado *inicio del repositorio* que incluya el inicio del repositorio con los archivos *README.md* y *.gitignore*
   - Un *commit* denominado *ejercicio 1*, que incluya al fichero del código fuente del primer ejercicio.
   - Un *commit* denominado *documentación*, que la realizas cuando hayas realizado la documentación del primer ejercicio.
   - Un *commit* denominado *ejercicio 2*, que incluya al fichero del código fuente del segundo ejercicio.

   Sincroniza el directorio local con el remoto.

Debes entregar los siguientes documentos:

- Numero.java
- Cadena.java
- Fichero de texto cuyo contenido sea la *URL* de tu repositorio del *GitHub* del examen

Entrégalos en un documento comprimido con el formato: *apellidosNombre.tar.gz* o *apellidosNombre.zip*. Tanto a la plataforma *moodle* como al profesor.