Tarea evaluable. Unidad 4. Cadenas de caracteres y arrays

Indicaciones de entrega

Una vez realizada la tarea, el envío se realizará a través de la plataforma Moodle, en la tarea "Tarea 3. Cadenas de caracteres y arrays".

Debes comprimir la carpeta que contiene el proyecto en un archivo. Este archivo comprimido se nombrará de la siguiente forma:
Apellido1_Apellido2_Nombre_Tarea3.zip

Evaluación

Cómo se valora y puntúa la tarea

Ejercicio 1. Creación del proyecto y entrega → 1.25 puntos
Ejercicio 2. Librería StringUtils → 1.25 puntos
Ejercicio 3. Ruleta de la fortuna → 2.5 puntos
Ejercicio 4. Clase StringBuilder → 1.25 puntos
Ejercicio 5. Clase Arrays → 1.25 puntos
Ejercicio 6. Máquina expendedora → 2.5 puntos

Factores que influyen en la puntuación de cada ejercicio

- Principal: El código debe compilar y ejecutarse correctamente
- Errores de base sobre conceptos de temas anteriores: nombres de identificadores incorrectos, elección de tipo inadecuado, etc.
- Elección de la estructura de control de flujo más apropiada para cada solución: if-else, switch-case, for, while...
- Calidad, legibilidad y limpieza del código.
- Control y manejo de excepciones que proporcionen una buena experiencia de usuario.
- La implementación de métodos que modularicen los programas.

Resultados de aprendizaje que se trabajan en la tarea

RA6: Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

Ejercicios

1. Creación del proyecto. Crea un único proyecto en Eclipse llamado tarea3CadenasArrays. Crea también un paquete que se llame "com.*nombreAlumno*.tarea3" donde ubicarás todo el código necesario.

2. Estudia la librería StringUtils de apache. Escoge 5 métodos que te parezcan útiles para el trabajo con cadenas. Explica qué hace y pon un ejemplo de uso que ayude a comprenderlo en el archivo de texto UtilesDeCadena.txt dentro del paquete del ejercicio 1.
   Ejemplo:

```
metodoA
El método A compara dos cadenas y devuelve la de mayor
longitud
String a=”aa”;
String b=”bbb”;
StringUtils.metodoA(a, b);
```

3. Realiza una aplicación dentro del paquete del ejercicio 1. (RuletaSuerte.java). que simule el juego de la ruleta de la suerte, para ello crea un array de 10 Strings que inicializaremos con 10 frases de platos de comida.
   1. Risotto con setas y parmesano
   2. Pollo al curry con arroz basmati
   3. Pescado en salsa de limón y alcaparras
   4. Ensalada de quinoa con aguacate y aderezo cítrico
   5. Ternera estofada con patatas y zanahorias
   6. Sushi de salmón y aguacate fresco
   7. Pizza con mozzarella y albahaca
   8. Lasaña de carne con ricotta y espinacas
   9. Tarta de chocolate con ganache y frambuesas
   10. Tacos de carnitas con salsa de mango

   La aplicación nos mostrará un menú como el que sigue:

   1.- Jugar
   2.- Finalizar el juego

   - Si elegimos la opción "2.- Finalizar el juego" se mostrará un mensaje de despedida y cerrará la aplicación.
   - Si seleccionamos la opción “1.- Jugar” la aplicación elegirá aleatoriamente una de esas 10 frases y nos mostrará en pantalla tantos guiones bajos como letras tenga la frase (los espacios deben quedar como tal), y debajo el submenú

     C.- Comprar letra
     R.- Resolver frase
     S.- Salir

- Si elegimos la opción "C.- Comprar letra" nos pedirá que introduzcamos una letra. Si la letra que introduzcamos existe en la frase nos mostrará de nuevo la frase pero con las apariciones de esa letra en su ubicación correcta.
- Si elegimos la opción "R.- Resolver frase" nos pedirá introducir una frase y nos mostrará un mensaje de enhorabuena si coincide con la frase o un mensaje de fracaso si nos hemos equivocado.
- Si elegimos la opción "S.- Salir" volverá al menú principal.

4. Estudia la clase StringBuilder y comprende cómo funciona.
Echa un vistazo a los métodos:

1. **append(String str)**
2. **insert(int offset, String str)**
3. **delete(int start, int end)**
4. **reverse()**
5. **toString()**
6. **length()**
7. **charAt(int index)**
8. **replace(int start, int end, String str)**
9. **substring(int start)**
10. **substring(int start, int end)**

Luego crea una pequeña aplicación (UsoDeStringBuilder.java) que a partir de una variable de tipo String, muestre por un lado las vocales y por otro las consonantes (en dos objetos StringBuilder).

5. Estudia la librería Arrays de java.util. Explica al menos los siguientes métodos.

1. **sort(T[] a)**
2. **binarySearch(T[] a, T key)**
3. **equals(T[] a, T[] a2)**
4. **fill(T[] a, T val)**
5. **copyOf(T[] original, int newLength)**
6. **toString(T[] a)**

Explica qué hacen y pon un ejemplo de uso que ayude a comprenderlo en el archivo de texto UtilesDeArrays.txt dentro del paquete del ejercicio 1.

6. Implementa el software para una máquina expendedora de bebidas (MaquinaExpendedora.java).
Cada bebida tiene un nombre y una cantidad de unidades disponibles:

```
String[][] productos = {
        {"Coca-Cola", "Sprite", "Fanta naranja"},
        {"Fanta limón", "Red Bull", "Monster"},
        {"Pepsi", "Schweppes tónica", "Agua mineral"},
        {"Cruzcampo", "Heineken", "Zumo de piña"}
};
```

La cantidad inicial en principio será de 5.

```
Integer[][] cantidad =
        {5, 5, 5},
        {5, 5, 5},
        {5, 5, 5}
        {5, 5, 5}
};
```

Tendremos un menú con las siguientes opciones:

1. **Pedir bebida:** pedirá la posición de la bebida que quiera. Esta máquina tiene bebidas en cada posición, identificados por su fila y columna, que será lo que introduzca el usuario al pedir una bebida, por ejemplo si el usuario teclea 21 significa que está pidiendo la bebida que está en la fila 2 y la primera columna. Cuando no haya más bebidas en dicha posición se le indicará al usuario. Recuerda de disminuir la cantidad al pedir.
2. **Mostrar bebidas:** mostrara todas las bebidas con al menos una unidad disponible. Mostrará el código que debe introducir el usuario y el nombre. La cantidad disponible no se mostrará.
3. **Rellenar bebidas:** esta es una función exclusiva de un técnico por lo que nos pedirá una contraseña, si el usuario escribe "MaquinaExpendedora2024" le pedirá la posición de la bebida y la cantidad que ha recargado en la máquina.
4. **Apagar maquina:** sale del programa, antes de salir mostrara las ventas totales durante la ejecución del programa.

El programa debe ser modularizado, es decir, todas las funciones que veas que sean necesarias debes crearlas, así como todas aquellas acciones que veas que se repitan.
Las funciones deben ser lo más genéricas posibles.
Utiliza arrays auxiliares si lo ves necesario.
Usa los métodos de la clase Arrays que te puedan ser útiles.