

Práctica

Realiza un programa en Java que compruebe la fortaleza de una contraseña introducida por el usuario. Para que una contraseña sea adecuada, según nuestro programa, debe ser elegida de acuerdo a unas reglas:

1. Debe contener al menos una letra mayúscula.
2. Debe contener al menos una letra minúscula.
3. Debe contener al menos un dígito (0-9).
4. Debe contener al menos un carácter entre los siguientes: * + - _ # \$ % &
5. No se considerarán caracteres válidos las vocales con tilde o diéresis, ya sean mayúsculas o minúsculas, así como cualquier otro no incluido expresamente en las reglas anteriores.
6. No puede contener espacios en blanco.
7. La longitud de la contraseña no puede ser inferior a 8 caracteres.

El programa invitará al usuario a introducir una clave candidata y debe verificar que contiene todos los elementos enumerados en las reglas anteriores. Si le faltan uno o más elementos de los citados en las reglas se verá un mensaje anunciando la carencia detectada por cada una de las reglas no respetada.

Si la clave candidata se ajusta al formato propuesto se invita a teclearla de nuevo para asegurar que no ha habido errores tipográficos y si coincide con la anterior se convierte en la contraseña con notificación en pantalla con un mensaje.

El programa permanecerá invitando a introducir claves candidatas y comprobándolas hasta que alguna sea designada como contraseña, en ese momento acabará tras mostrar *“Programa terminado”* en pantalla.

Sugerencia: Se puede usar el método *codePointAt(posición)* de la clase *String* que nos devuelve el código ASCII del carácter que se le haya pasado como parámetro.

Ejemplo de uso: si tenemos un *String* puedo saber cuál es el código ASCII de los caracteres que lo componen:

```
String unaCadena = "DAM";  
int unCodigoASCII = unaCadena.codePointAt(1);
```

Si se ejecutan esas líneas en la variable *unCodigoASCII* tendremos el valor 65 que es el código ASCII de la 'A', el carácter que está en la posición 1 de *unaCadena*.