

## Problema 1: la classe TextHistogram

En l'àmbit de l'estadística, un histograma és una representació gràfica del nombre d'ocurrències que s'han observat en diferents categories d'un experiment.

Per exemple, la figura següent mostra un histograma de les notes finals que es van obtenir a l'assignatura de Programació 2 en un curs determinat:

```

***** (1)
***** (2)
(3)
***** (4)
*** (5)

```

En aquest histograma hi ha cinc categories representades: la categoria 1 indica "Suspens"; la categoria 2, "Aprovat"; la 3, "Notable"; la 4, "Excel·lent"; i, finalment, la 5, "Matrícula d'honor". El nombre d'asteriscs indica el nombre d'observacions fetes en cada categoria. Tot plegat, ho podríem llegir: a l'assignatura hi ha hagut 7 suspesos, 10 aprovats, cap notable, 6 excel·lents i 3 matrícules d'honor.

Com volem fer estadístiques de varies coses, hem decidit crear una classe que anomenarem `TextHistogram` que permeti:

- configurar un histograma
- recollir les dades per a construir-lo
- convertir l'histograma en un `String` per a poder-lo escriure

Per poder fer això la classe tindrà els següents constructors i mètodes públics:

- **`new TextHistogram(char symbol, int numCategories)`**
  - o el constructor rep el caràcter a usar al dibuixar l'histograma i el nombre de categories que es considera (és a dir, es podran entrar els nombres des de 1 fins a `numCategories`).
- **`void observed(int category)`**
  - o indica que s'ha d'afegir una nova observació de la categoria indicada per `category`.
  - o si el nombre donat no està al rang 1 fins a `numCategories` aquesta operació no fa res.
- **`String toString()`**
  - o converteix l'histograma en un `String` amb una sèrie de línies, una per cada categoria, començant des de la 1 fins a `numCategories`.
  - o per cada categoria afegeix l'`String` una línia amb tants caràcters `symbol` com observacions s'han produït de la categoria; després escriu un espai i, entre parèntesis, el número corresponent a la categoria.
    - el salt de línia està representat pel caràcter `'\n'` o la cadena `"\n"`
  - o si una categoria no té observacions s'afegeix només l'espai i entre parèntesis el número corresponent a la categoria.
  - o **no** cal que controleu la longitud de la línia.
  - o aquest dibuix us ha de quedar de manera similar a la figura inicial.

**Feu servir els mètodes privats que considereu convenient per a que la vostra solució quedi**

el més clara possible.

## Problema 2: histograma de longituds de paraules

Fer un programa que demani un text a l'usuari i dibuixi l'histograma corresponent a les llargàries de cadascuna de les paraules que hi conté.

Per a fer això utilitzarem la classe del problema anterior (**la podeu usar encara que no l'hagueu implementat, ja que només necessiteu saber el seu constructor i els mètodes públics**). Per exemple, considerant només 3 longituds, si n'hi ha 2 de longitud 1, cap de longitud 2 i 8 de longitud 3, l'histograma serà:

```
** (1)
(2)
***** (3)
```

Suposicions:

- podeu considerar com caràcters de separació els espais i signes de puntuació
- si una paraula conté algun caràcter que **no** és una lletra, s'ha d'ignorar completament
- les paraules de més de 8 caràcters també s'han d'ignorar
  - o d'aquesta manera, tots els histogrames que dibuixarà el programa tindran exactament 8 categories: les corresponents a paraules des d'1 fins a 8 lletres
- el símbol a usar a l'escriure l'histograma és \* (asterisc).

El que es demana és que implementeu aquest programa, que usarà la classe del problema anterior a més de d'altres definides tant a la biblioteca de l'ACM com al Java estàndard.

**Feu servir els mètodes privats que considereu convenients per a que la vostra solució quedi el més clara possible.**

## Alguns mètodes útils

- `class Character`
  - o `static boolean isDigit(char ch)`
  - o `static boolean isLetter(char ch)`
  - o `static boolean isWhitespace(char ch)`
- `class ConsoleProgram`
  - o `String readLine(String message)`
  - o `void println(String str)`
- `class String:`
  - o `char charAt(int index)`
  - o `int length()`
  - o `String concat(String s)`
    - També podeu fer servir l'operador + per concatenar Strings
  - o `static String valueOf(char ch)`
  - o `static String valueOf(int n)`
- `class StringTokenizer:`
  - o `new StringTokenizer(String str)`
  - o `new StringTokenizer(String str, String delims)`
  - o `new StringTokenizer(String str, String delims, boolean returnDelims)`

- o `boolean hasMoreTokens()`
- o `String nextToken()`