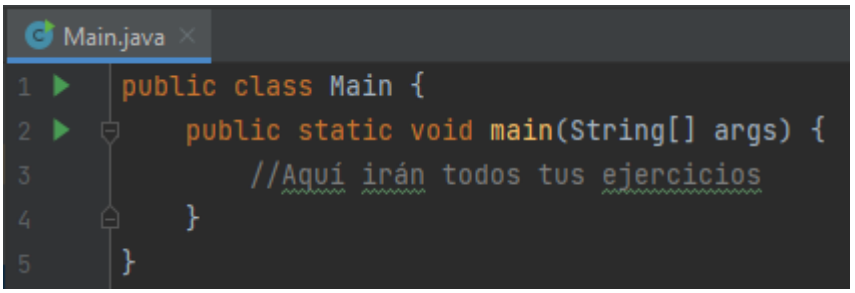
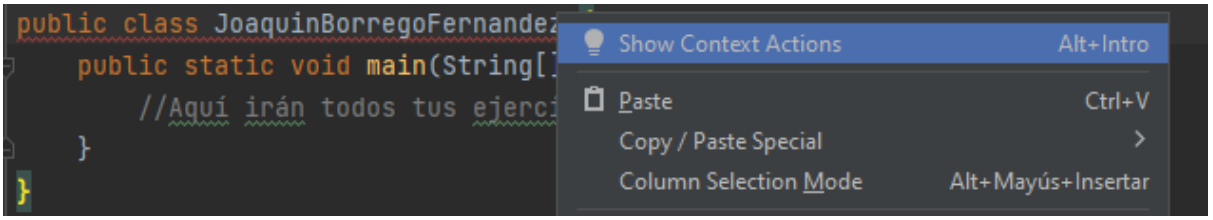
Tarea Entregable - Programación

QUÉ DEBE ENTREGAR

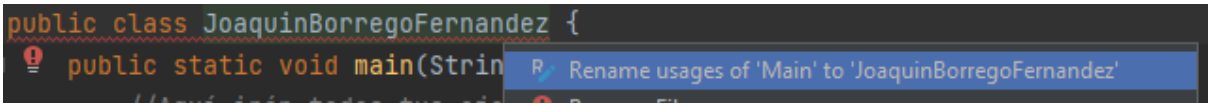
Cuando acabe su tarea debe entregar en la tarea del classroom los ficheros .java donde está su código. Debe entregar los ficheros de código fuente con el nombre exacto que pida cada enunciado.
En cuanto al main, hay que entregarlo de la siguiente manera:



Cambiadle el nombre de Main (cuidado!, Main con mayúscula la primera letra) por vuestro nombre y apellidos sin espacios ni tampoco acentos. Ésto os saldrá en rojo como error, pero con un click derecho y presionando show context actions como en la siguiente imagen:



y pulsando la primera opción, rename usages of Main to 'NombreApellidounoApellidodos' como en la imagen inferior:



Nos debería de salir todo en orden.

Se debe entregar cada ejercicio totalmente descomentado, es decir, cuando acabe cada ejercicio NO LO COMENTE con /* y */ o se interpretará como un comentario y no se corregirá. No puede pedirle al profesor ayuda. Si no entiende bien un enunciado, debe interpretarlo como crea conveniente y no pedir aclaraciones al profesor, hacer un examen también consiste en interpretar bien lo que se le pide. CRTL + ALT + L para reformatear.

Ejercicio 1:

Desarrolle una clase llamada CrucigramaNombreApellido1, siendo Nombre tu nombre y Apellido1 tu primer apellido (Ejemplo: CrucigramaJoaquinBorrego). La clase debe tener la siguiente información a la cual sólo se tendrá acceso desde esta misma clase:

1) Un atributo estático y constante Scanner ya inicializado
2) Un atributo “row” que sea un array de Strings
3) Un atributo “col” que sea un array de Strings
4) Un atributo “sol” que sea una matriz de char
5) Un atributo maxLen que será un número entero

Debe generar un constructor vacío cuyos valores por defecto serán:

1) El atributo row tendrá el array {“.”}
2) El atributo col tendrá el array {“.”}
3) El atributo sol tendrá un nuevo array de 1x1
4) El atributo maxLen valdrá 1

Debe generar además otro constructor que reciba los datos row y col y que inicialice sol como una matriz nxm siendo n la longitud de row y m la longitud de col. El atributo maxLen vendrá dado por el método maxLong() que se pedirá en apartados posteriores (de momento puedes ponerlo como 1).

Generar los siguientes métodos:

1) Método setter sólo para el campo sol.
2) Métodos validadores para los campos row, col y sol y usarlos donde estime conveniente. Estos métodos solo se podrán utilizar en esta misma clase.
3) Método maxLong() que no recibe nada y devuelve un entero. Este método recorrerá tanto el atributo row como

el atributo col y devolverá la longitud del String más largo. Se usará en el constructor con datos.

4) Método ampliar que recibe un String str y que devuelve el String resultante de concatenar lo siguiente:
   - La repetición del espacio en blanco “ ” un total de (maxLen - longitud de str) veces
   - La propia cadena str
   - La cadena “ |”
5) Método introducirSol() que no recibe ni devuelve nada y que pedirá al usuario por pantalla que meta el carácter que considere que va en cada una de las posiciones.
6) Método comprobarSol() que no recibe nada y devuelve un booleano resultante de comprobar si la solución es correcta. Es decir, que la palabra que forma los caracteres ubicados en la fila i cumpla la regex ubicada en row[i] y la palabra que forma los caracteres ubicados en la fila j cumpla la regex ubicada en col[j].
7) Método borrarSol() que inicializa de nuevo el atributo sol.
8) Método toString() que devuelva un String que muestre la clase como un crucigrama (Pista: ampliar cada uno de los datos con el método ampliar()).

Por último, realizar un bucle en el main que recorra un array de Crucigramas y se vaya mostrando el crucigrama a realizar, pidiendo al usuario que introduzca la solución y mientras que esa solución no sea la correcta se le vuelva a preguntar por ese mismo.

El array de Crucigramas puede ser este:

```
String[] col = {"[JUNDT]*","APA|OPI|OLK","(NA|FE|HE)[CV]" };

String[] row = {"[DEF][MNO]*","[^DJNU]P[ABC]","[ICAN]*"};

//char[][] sol = {{'D','O','N'},{'T','P','A'},{'N','I','C'}};
```

```
CrucigramaJoaquinBorrego cr = new CrucigramaJoaquinBorrego(col, row);


String[] col2 = {"UB|IE|AW","[TUBE]*","[BORF]." };

String[] row2 = {"[NOTAD]*","WEL|BAL|EAR"};

//char[][] sol2 = {{'A','T','O'},{'W','E','L'}};

CrucigramaJoaquinBorrego cr2 = new CrucigramaJoaquinBorrego(col2, row2);


String[] col3 = {"[BQW](PR|LE)","[RANK]+"};

String[] row3 = {"[AWE]+","[ALP]+K","(PR|ER|EP)"};

//char[][] sol3 = {{'W','A'},{'L','K'},{'E','R'}};

CrucigramaJoaquinBorrego cr3 = new CrucigramaJoaquinBorrego(col3, row3);


CrucigramaJoaquinBorrego[] crs = {cr, cr2, cr3};
```

```
                |       [JUNDT]* |    APA|OPI|OLK | (NA|FE|HE)[CV] |
    [DEF][MNO]* |              Ø |              Ø |              Ø |
  [^DJNU]P[ABC] |              Ø |              Ø |              Ø |
        [ICAN]* |              Ø |              Ø |              Ø |

Introduce caracter en la posicion (0, 0):
D
Introduce caracter en la posicion (0, 1):
O
Introduce caracter en la posicion (0, 2):
N
Introduce caracter en la posicion (1, 0):
T
Introduce caracter en la posicion (1, 1):
P
Introduce caracter en la posicion (1, 2):
A
Introduce caracter en la posicion (2, 0):
N
Introduce caracter en la posicion (2, 1):
I
Introduce caracter en la posicion (2, 2):
C
                |       [JUNDT]* |    APA|OPI|OLK | (NA|FE|HE)[CV] |
    [DEF][MNO]* |              D |              O |              N |
  [^DJNU]P[ABC] |              T |              P |              A |
        [ICAN]* |              N |              I |              C |

Correcto! Siguiente
             |    UB|IE|AW |     [TUBE]* |     [BORF]. |
    [NOTAD]* |           Ø |           Ø |           Ø |
 WEL|BAL|EAR |           Ø |           Ø |           Ø |

Introduce caracter en la posicion (0, 0):
```