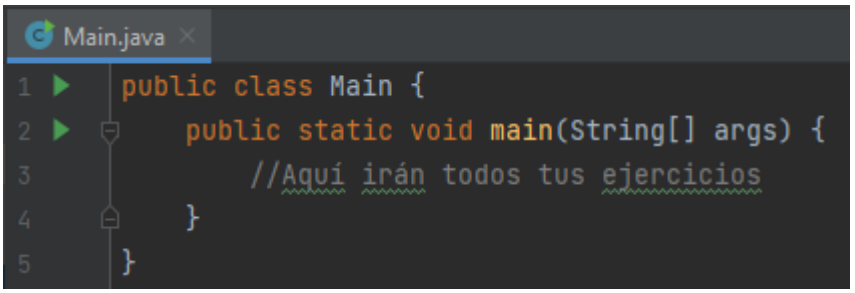
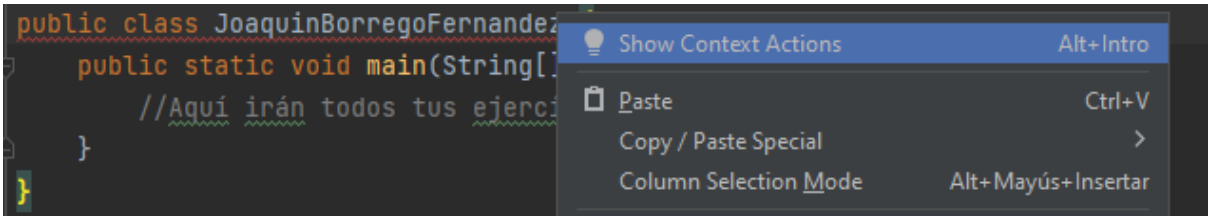
Tarea Entregable - Programación

QUÉ DEBE ENTREGAR

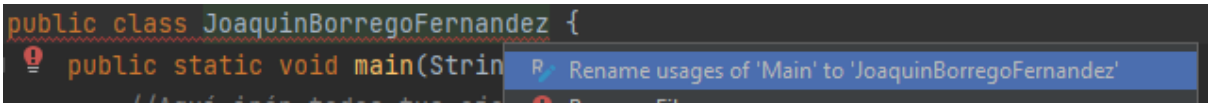
Cuando acabe su tarea debe entregar en la tarea del classroom los ficheros .java donde está su código. Debe entregar los ficheros de código fuente con el nombre exacto que pida cada enunciado.
En cuanto al main, hay que entregarlo de la siguiente manera:



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 ArmaNombreApellido1, siendo Nombre tu nombre y Apellido1 tu primer apellido (Ejemplo: ArmaJoaquinBorrego). La clase debe tener la siguiente información:

1) Una variable constante rarezas, que contendrá el siguiente array: {"\033[0;37m", "\033[0;32m", "\033[0;36m", "\033[0;35m", "\033[0;33m"}
2) Una variable constante reset que tendrá el siguiente valor: "\033[0m"
3) modelo de arma
4) la cantidad de munición máxima que puede guardar el cargador
5) la cantidad de munición actual que tiene el cargador
6) la rareza del arma, representada por un entero entre 0 y 4

Debe generar un constructor vacío, cuyos valores por defecto serán:

- modelo: Sig-Sauer P-226
- munición maxima cargador: 20
- munición actual cargador: 20
- rareza: 0

Debe generar, además, un constructor que reciba todos los datos.

También se precisa generar todos los getters y setters.

Debe crear código de validación: si se intenta insertar un objeto null o un valor numérico incorrecto en alguno de los atributos lance una excepción del tipo adecuado.

Debe crear el método toString, de forma que cuando se imprima un libro por pantalla tenga exactamente el siguiente formato:

RAREZAS[rareza] + " +--^----------,--------,-----,--------^-,\n" +

" | ||||||||| `--------' | O\n" +

" `+------------------------^----------|\n" +

"   `_,---------,---------,--------------'\n" +

"   / XXXXXX /'| /'\n" +

"   / XXXXXX / `\\   /'\n" +

"   / XXXXXX /`-------'\n" +

" / XXXXXX /\n" +

" / XXXXXX /\n" +

"(_____________(\n" +

" `------'   \n" + RESET +

modelo + "\t\t\t" + municionActualCargador + " / " + municionMaxCargador;

Debe crear un método disparo que no reciba ningún argumento ni devuelva nada y que simplemente imprima por pantalla el string “Pam”.

Ejercicio 2:

Desarrolle una clase llamada FusilDeAsaltoNombreApellido1, siendo Nombre tu nombre y Apellido1 tu primer apellido (Ejemplo: FusilDeAsaltoJoaquinBorrego) teniendo en cuenta que un fusil de asalto es un tipo de arma. La clase, además de la información típica de un arma, debe tener la siguiente información:

1) Un atributo silenciador cuyo valor nos indicará si tiene silenciador puesto o no

Debe generar un constructor vacío, cuyos valores por defecto serán los mismos que arma a excepción de:

- modelo: Carabina M4

Debe generar, además, un constructor que reciba todos los datos.

También se precisa generar los getters y setters.

Debe crear el método toString, de forma que cuando se imprima un libro por pantalla tenga exactamente el siguiente formato:

```
RAREZAS[rareza]    +    "                                |\\_________________
(_____\\\\_________________\n" +

"HH======#H################H##########################\n" +
            "       ' ~\"\"\"\"\"\"\"\"\"\"\"\"\"`##(_))#H\\\\\"\"\"\"Y########\n" +
            "                        ))    \\#H\\       `\"Y###\n" +
            "                        \"      }#H)" + "\n" + RESET +
              modelo + (silenciador ? "(Con" : "(Sin") + " Silenciador)\t\t\t" +
municionActualCargador + " / " + municionMaxCargador;
```

Sobreescribir el método disparo para que, en caso de no tener silenciador, imprima por pantalla el String “Ratatatata ratatatata”. En caso de tenerlo, imprimir “Rititititi rititititi”

Ejercicio 3:

Desarrolle una clase llamada EscopetaNombreApellido1, siendo Nombre tu nombre y Apellido1 tu primer apellido (Ejemplo: EscopetaJoaquinBorrego) teniendo en cuenta que una escopeta es un tipo de arma. La clase, además de la información típica de un arma, debe tener la siguiente información:

2) Un atributo culata cuyo valor nos indicará si tiene culata puesta o no

Debe generar un constructor vacío, cuyos valores por defecto serán los mismos que arma a excepción de:

- modelo: Remington 870

Debe generar, además, un constructor que reciba todos los datos.

También se precisa generar los getters y setters.

Debe crear el método toString, de forma que cuando se imprima un libro por pantalla tenga exactamente el siguiente formato:

```
RAREZAS[rareza] + " ,__________________________________________ \n" +
"|______________________,----------._ [____] \"\"-,__ __....-----=====\n" +
" (_(||||||||||||)________________/ \"\" |\n" +
" `----------' [ ))\"-, |\n" +
" \"\" `, _,--....___ |\n" +
" `/ \"\"\"\" \n" + RESET +
modelo + (culata ? "(Con" : "(Sin") + " Culata)\t\t\t" + municionActualCargador + " / " + municionMaxCargador;
```

Sobreescribir el método disparo para que imprima por pantalla el string “Pump”.

Ejercicio 4:

Testear dichas clases en el main de la siguiente forma:

- Declarar una variable ArmaNombreApellido1, pero no instanciarla.
- Intentar asignarle una instancia de dicha clase con datos inválidos y, cuando capturemos la excepción, asignarle a la variable una instancia dicha variable con el constructor vacío.
- Setear la rareza de dicha variable con valor 4.

- Declarar una variable FusilDeAsaltoNombreApellido1, pero no instanciarla.
- Intentar asignarle una instancia de dicha clase con datos inválidos y, cuando capturemos la excepción, asignarle a la variable una instancia dicha variable con el constructor vacío.
- Setear la rareza de dicha variable con valor 3 y el silenciador con valor true.
- Declarar una variable EscopetaNombreApellido1, pero no instanciarla.
- Intentar asignarle una instancia de dicha clase con datos inválidos y, cuando capturemos la excepción, asignarle a la variable una instancia dicha variable con el constructor vacío.
- Setear la rareza de dicha variable con valor 2.

Ejercicio 5:

Desarrolle una clase InventarioNombreApellido1 (Ejemplo: InventarioJoaquinBorrego), destinada a almacenar una serie de armas. Implemente los atributos necesarios, getters, setters y un constructor que reciba la serie de armas que va a almacenar en el inventario. Como máximo se podrán almacenar 5 armas. Desarrollar un validador en caso de que se intente almacenar más armas, seleccionando en este caso las 5 primeras y, en caso de que se introduzca un valor nulo, lanzar la excepción debida. El método toString debe devolver la concatenación de todos los toString de los elementos de la colección.

Ejercicio 6:

Desarrolle una clase MetodosEstaticosDeNombreApellido1. En ella, deberá desarrollar los siguientes métodos estáticos:

- Un método imprimirStrings que reciba un array de String y no devuelva nada. El método imprimirá cada String en una línea distinta.
- Un método ampliar que reciba un string y un entero y devuelva la concatenación del string recibido con tantos espacios " " como la diferencia entre el entero recibido y la longitud del string recibido. (Por ejemplo, si ejecuto ampliar("Hola", 7) devolverá "Hola "). Pista: usar método repeat.
- Un método desplazar que reciba un array de string y un numero entero y devuelva un array de string. Éste array de string que devuelve debe ser un array con el tamaño del numero entero recibido y debe almacenar el array de string recibido en sus últimas posiciones, rellenando las primeras con cadenas vacías. (Por ejemplo, si ejecuto desplazar({"Hola","Mundo"}, 5) devolverá {"","","","Hola","Mundo"}).
- **(MUY DIFÍCIL)** Un método imprimirInventario que reciba un InventarioNombreApellido1, no devuelva nada e imprima las armas en la misma línea. En caso de conseguirlo, adaptar el ejercicio para usar dicho String como retorno del método toString del inventario. Éste método sería más correcto desarrollarlo en la clase InventarioNombreApellido1. Pista: usar métodos anteriores.


Sig-Sauer P-226 20 / 20
Sig-Sauer P-226(Sin Culata) 20 / 20