COLEXIO VIVAS S.L.	RAMA:	Informática	CICLO:	DAM		
	MÓDULO	Programación			CURSO:	1º
	PROTOCOLO:	Boletín de ejercicios	AVAL:		DATA:	
	AUTOR	Francisco Bellas Aláez (Curro)				

Boletín de prácticas de Programación III: Métodos

***Notas**:*

- *En ciertos ejercicios se pide simplemente una función. Evidentemente hay que hacer un pequeño programa (como quiera el alumno) para probar que dichas subrutinas funcionan, por lo que conviene hacer llamadas a la función con distintos datos o preparar una entrada de datos por parte del usuario.*
- *Se recomienda realizar todos los ejercicios del boletín antes de introducirse en las partes "opcionales" de los distintos ejercicios salvo que el alumno tenga las cosas muy clara y se vea suelto.*
- *Aunque el ejercicio no lo indique (sobre todo en los últimos) **es necesario** aplicar modularidad en la medida de lo posible evitando repetir código y minimizando el tamaño del programa principal.*
- ***Es obligatorio comentar** las funciones tal y como se vio en clase. No se valida la función que no esté comentada para javadoc.*
- *La usabilidad es obligatoria a partir de este boletín si ya se vio en Entornos de desarrollo.*
- *Evita buscar soluciones en Internet, no es una buena idea. Trata de pensar tú la forma de realizar los ejercicios.*

1. a) Codifica un método que deje en pantalla *N* líneas en blanco (*N* es un parámetro).
   b) Escribe una función denominada *par* con un único parámetro entero. Devuelve true si el parámetro es número par y false si no lo es.
   c) Escribir una función que tenga un argumento (otra forma de llamar a un parámetro) de tipo entero y que devuelva la letra *P* (devuelve char) si el número es positivo y la letra *N* si es cero o negativo. Intenta hacerlo con el operador ternario (si no te sale hazlo con if).

2. Función año bisiesto. Realizar una función denominada *bisiesto* a la cual se le pasa un año como parámetro y devuelve *true* si dicho año es bisiesto y *false* en caso contrario.
   Un año es bisiesto cuando:
   - Es múltiplo de 4 (P. ej 1984)
   - Pero los múltiplos de 100 no lo son (Por ejemplo 1800)
   - Salvo si a su vez son múltiplos de 400 que caso sí lo son (p. ej. 2000)

   En el programa principal se hará un bucle que pida continuamente años al usuario hasta que introduzca el año 0 momento en el cual el programa termina.

3. a) Realizar una función que halle y devuelva la potencia de un número (**No se permite usar funciones de Math**). La base puede ser real y el exponente entero puede ser negativo (recuerda que $a^{-b} = (1/a)^b$). Parámetros mínimos del procedimiento: la base y el exponente. Valor devuelto: la potencia
   b) Realiza un nuevo método que muestre en pantalla las N primeras potencias de un número (N será un parámetro del método). Para hacerlo deberás llamar al creado en (a) para calcular las potencias, no las puedes calcular de nuevo en este método.

4. Escribir un método que sume la progresión geométrica (*x* puede ser real) y devuelva el resultado.
   $1+x+x^2+x^3+x^4+...\ x^n$
   Se debe utilizar la función potencia (sin modificarla, sólo llamándola) programada anteriormente.
   Parámetros del método: x y n. Valor devuelto*: resultado de la progresión.*

5. Escribir una función que se le pase un número y que devuelva *true* o *false* dependiendo de si dicho número es o no es primo. Para probarla haz un programa que muestre los números primos menores que cierto valor que se le pide al usuario.
   Nota: Un número es primo si y sólo si es divisible únicamente por 1 y por él mismo. Por tanto para saber si un número es primo se debe dividir por todos los números menores que él y mayores que 1, y si

alguna de esas divisiones es cero entonces el número NO es primo. Existen métodos de optimizar lo anterior, piensa a ver si se te ocurre alguno.

6. a) Realizar el programa "Adivina un número entre 1 y 100" según las especificaciones del boletín previo utilizando subprogramas. Al menos deben existir los siguientes subprogramas:
   - Un método al que se le pasen dos números y devuelva 0 si el primero es menor, 1 si son iguales y 2 si el primero es mayor. No debe tener ningún interfaz con el usuario. Luego, en el programa principal, dependiendo del resultado de esta función y de una estructura *switch-case*, se mostrarán unos mensajes u otros por pantalla.
   - Un método de introducción de datos. Se pide una variable donde se debe almacenar un número entre 1 y 100, mostrando mensaje de error y volviendo a pedir el número sucesivamente si no está en ese rango. Devolverá dicha variable. A este método se le llamará tanto para la introducción de datos del jugador 1 como para los intentos del jugador 2.

   Si se desea, se pueden realizar más subprogramas.
   b) Modificar el programa anterior para añadirle una opción para un solo jugador (plantea un menú de 1 o 2 jugadores al principio del programa). De forma que el número inicial debe ser aleatorio generado por el ordenador mediante la función *Math.random()*. Al principio el programa pedirá si se desea jugar con uno o dos jugadores.

7. a) Realizar un método que permita hallar el área de un rectángulo o de un triángulo rectángulo a partir de la base, la altura y un parámetro booleano (denominado bandera, flag o interruptor) para decidir si se trata de un rectángulo o un triángulo. Devuelve el área.
   Parámetros del método: base, altura y la bandera.

   b) Codificar un menú con las siguientes opciones:
   - Area de un triángulo (base*altura/2)
   - Area de un rectángulo (base*altura)
   - Area de un cuadrado (lado*lado)
   - Area de un círculo (Pi*radio^2)
   - Salir

   Las tres primeras opciones deben realizarse **utilizando el método creado en el apartado anterior** (por supuesto, **sin modificarlo**). Para la cuarta debe crearse un nuevo método con parámetro radio.

8. a) Realizar una función que devuelva el factorial de un número (recuerda que 0!=1).
   b) Realizar un método que realice y devuelva el siguiente cálculo (es el denominado binomio de Newton).

$$\frac{m!}{n!\cdot(m-n)!}$$

   Parámetros: *m, n*. Se debe usar la función de (a)
   c) Realizar un programa que pida al usuario los datos *m* y *n* del binomio y muestre el coeficiente. Es obligatorio que *m* y *n* sean mayores que 0 y además *m>=n.*
   Opcional: Realiza la función factorial aplicando recursividad (ver apéndice de apuntes, evita buscar en internet).

9. a) Codificar un programa que genere una quiniela aleatoria. Es decir, se deben dar 14 resultados aleatorios como 1, X ó 2 indicando delante el número de partido y los resultados alineados en una columna (no uses \t, si no ajuste con printf). Se debe realizar al menos una función que devuelva un 1 una X o un 2 (char o String).
   b) Realizar la quiniela ponderada, es decir, que la probabilidad de sacar 1 sea del 60%, la de sacar X sea 25% y la de sacar un 2 sea un 15%.
   Pista: Sacar un numero aleatorio entre 1 y 100. Si el resultado es menor o igual que 60, se asigna un 1 al resultado, si es entre 61 y 85 (60+25) se le asigna una X y si es entre 86 y 100 se le asigna un 2.

10. Semi-Lotería primitiva: Realizar un programa que pida 3 números distintos al usuario entre 1 y 20 (usar 3 variables) y luego que el ordenador saque 2 números aleatorios distintos y diga los aciertos que ha tenido el usuario. Evita repetir código haciendo funciones.

    Opcional: Crear un interfaz más agradable. Se le mostrará un tablero con los 20 números y aparecerán en color verde los aciertos, en rojo sacados por el ordenador y no acertados y en azul los seleccionados no acertados.

11. Juego craps: Se tiran dos dados de seis caras. Se calcula la suma de ambos. Si la suma es 7 u 11 en la primera tirada el jugador gana. Si la suma es 2, 3 o 12 en la primera tirada (se denomina craps) el jugador pierde (gana la CPU). Si la suma es un nº entre 4 y 10 salvo el 7, dicha suma son los puntos del jugador. Luego tira la CPU con las mismas reglas. Si al final ambos sacan puntuación gana la de mayor valor. Deben existir al menos las siguientes funciones:
    tirada: Tira dos dados, muestra sus valores en pantalla y devuelve la suma.
    comprobacion: Se el pasa un valor y devuelve -1 si pierde, 0 si gana o la puntuación en otro caso.

12. Programar un juego "mini-hundir-la-flota" contra el ordenador. Se debe generar una coordenada (de un tablero de 10x10 posiciones) aleatoria (sin mostrar nada en pantalla) y guardarla (barco del ordenador). Luego el usuario debe introducir una coordenada que el ordenador intentará averiguar (barco del usuario). A partir de ese momento y por turnos el usuario meterá coordenadas con la idea de localizar el barco del ordenador. Tras su turno será el ordenador el que diga una coordenada aleatoria y comprobará si coincide con la del barco del usuario. No es necesario comprobar si el ordenador repite coordenadas. Lo que sí es necesario es que el usuario pueda introducir coordenadas del estilo de (A,4) con letras mayúsculas o minúsculas (pide primero la letra y luego el número de la coordenada). Se debe hacer programación modular.

    Opcional: Realizar el juego pero con un barco de 3 casillas. Se debe sacar una coordenada aleatoria, y una dirección aleatoria (horizontal o vertical). A partir de ahí se crean las 3 coordenadas del barco. Si el ordenador acierta una casilla no sigue tirando aleatoriamente si no que lo intenta en las coordenadas de alrededor en horizontal o vertical (no diagonal)

13. a) Realizar una función que halle y devuelva la tangente de un ángulo. (*tan(x)=sen(x)/cos(x)*)
    b) Realizar un método al que se le pase un ángulo dado en grados y lo devuelva en radianes (*rad=grados*Pi/180*).
    c) Realizar un programa que pida si el usuario desea trabajar en grados o en radianes. Si elige grados debes usar la función de conversión a radianes para usar Math.cos y Math.sin. Posteriormente se debe

plantear un menú con las opciones:

- Coseno de un ángulo (Usa Math.cos): Pide un ángulo y muestra su coseno
- Seno de un ángulo (Usa Math.sin): Pide un ángulo y muestra su seno
- Tangente de un ángulo (Usa la función creada en (a): Pide un ángulo y muestra su tangente
- Salir: mientras no se seleccione esto no se sale del menú. No vuelve a pedir la unidad del ángulo, solo el menú.

Ejercicios Avanzados (Opcionales)

14. Calendario Perpetuo:

    Escribir un programa mediante subrutinas que permita ver el mes de cualquier año de la forma

```
 L     M     X     J     V     S     D
             1     2     3     4     5
 6     7     8     9    10    11    12
…
27    28    29    30
```

    El usuario indicará únicamente mes y año que desea visualizar. Para saber en qué día de la semana correspondiente a una fecha dada se debe buscar en la web ya que hay varios válidos. Una vez encontrado el algoritmo se usará para obtener el primer día del mes y el resto se colocan a partir de este.

15. Mejorar alguno de los opcionales del boletín anterior reestructurándolo en funciones.

16. Realiza un programa que represente algunas funciones matemáticas como seno, coseno, tangente, una parábola, etc... permitiendo modificar alguno de los parámetros. Evidentemente se dibujaran en una resolución muy baja usando | para el eje vertical, - para el horizontal y * para la función.

17. Realizar el juego de los cañones (Versión antigua del Angry Birds o Worms). Para dos jugadores o un jugador contra la CPU. Aparecen un cañon a cada lado de la pantalla apuntando al otro. Ambos están en un terreno con distinta elevación (que será aleatoria en cada partida) y dispararán de forma alterna dando indicaciones de angulo y fuerza. Puede haber también fuerza del viento u otros factores que afecten a los disparos.
    Nota: este último opcional es complicado pues incluye un manejo de los códigos ANSI más fuerte así como la creación de la física del juego (un disparo parabólico).

18. Realiza un juego conversacional sencillo en consola. Puedes tomar como referencia Zork (dejo abajo varios enlaces). Haz, eso sí, un mapa más pequeño y una historia más breve. Modularízalo correctamente.
    Juego en español: http://iplayif.com/?story=http%3A//media.textadventures.co.uk/games/GQLrNEIS8k26ddJ2nco6ag/Zork.gblorb
    Juego original: http://textadventures.co.uk/games/play/5zyoqrsugeopel3ffhz_vq
    Mapa: http://www.caad.es/sites/default/files/descargas/Juegos/Glulx/Map%20(Eng).jpg
    Video demostrativo: https://www.youtube.com/watch?v=xzUagi41Wo0