Ejercicios con funciones

Realiza los ejercicios que siguen y prueba cada una de las funciones declaradas en un programa para saber si funcionan adecuadamente.

1º Define una función que determine si dos números *float* dados como parámetros son iguales.

Solución:

```
class Main {
  public static void main(String args[]) {
      System.out.println();
      System.out.println("0.5 y 2.0 = " + igualesFloat(0.5f, 2.0f));
      System.out.println();
      System.out.println("2.0 y 2.0 = " + igualesFloat(2.0f, 2.0f));

  } // De la función main

  /* Declaración de la función igualesFloat que toma dos parámetros float
     y devuelve un boolean a true si son iguales y a false en caso contrario. */
  public static boolean igualesFloat(float num1, float num2){
      boolean resultado = false;

      if (num1 == num2)
          resultado = true;

      return resultado;

  }// De la función igualesFloat

} // De la clase
```

2º Define una función que determine si un número entero dado como parámetro es impar.

Solución:

```
class Main {
  public static void main(String args[]) {

      if (esImpar(8))
          System.out.println("El 8 es impar");
      else
          System.out.println("El 8 es par");

      if (esImpar(13))
          System.out.println("El 13 es impar");
      else
          System.out.println("El 13 es par");
  } // De la función main

  /* Declaración de la función esImpar que toma un parámetro int
     y devuelve un boolean. Será true si el parámetro no es divisible
     entre dos y false en caso contrario. */
  public static boolean esImpar(int numero){
      boolean resultado = false;
      if (numero % 2 != 0)
          resultado = true;

      return resultado;
  }// De la función esImpar

}// De la clase
```

3º Define una función que determine si un número entero positivo dado como parámetro es primo o no. Si el número estudiado fuera negativo se ha de indicar con un mensaje y devolver *false*.

Solución:

```
class Main {
  public static void main(String args[]) {

     System.out.println();
     System.out.println("Resultados con las distintas funciones realizadas:");
     System.out.println("             esPrimo()            esPrimoMejorada()       esPrimoDelegando()");

        System.out.println("Para el 3:    " + esPrimo(3) + "           " + esPrimoMejorada(3) + "       " + esPrimoDelegando(3));
     System.out.println("Para el 4:    " + esPrimo(4) + "           " + esPrimoMejorada(4) + "       " + esPrimoDelegando(4));
     System.out.println("Para el 17:   " + esPrimo(17) + "          " + esPrimoMejorada(17) + "      " + esPrimoDelegando(17));
     System.out.println("Para el 256:  " + esPrimo(256) + "             " + esPrimoMejorada(256) + "         " + esPrimoDelegando(256));
     System.out.println("Para el -8: " + esPrimo(-8) + "           " + esPrimoMejorada(-8) + "      " + esPrimoDelegando(-8));

  } // De la función main

 /* Declaración de la función esPrimo que toma un parámetro int
    y devuelve un boolean. Será true si el parámetro no es divisible
    entre ningún número distinto de 1 o de él mismo y false en caso contrario. */
 public static boolean esPrimo(int numero){
     boolean resultado = true;
     int numDivisores = 0;

     if (numero < 0) {
         System.out.println("\nEl número ha de ser positivo");
         resultado = false;
     }
     else {
         for (int i = 2; i < numero; i++){
             if (numero % i == 0)
                 numDivisores++;
         }
         if (numDivisores > 0)
             resultado = false;
     }

     return resultado;
 }// De la función esPrimo

 /* Declaración de la función esPrimoMejorada que toma un parámetro int
    y devuelve un boolean. Será true si el parámetro no es divisible
    entre ningún número distinto de 1 o de él mismo y false en caso contrario.
    Examina menos candidatos a divisor que la anterior porque solo trata los
    números que llegan hasta la mitad más uno del parámetro*/
 public static boolean esPrimoMejorada(int numero) {
     boolean resultado = true;
     int numDivisores = 0;

     if (numero < 0) {
         System.out.println("\nEl número ha de ser positivo");
         resultado = false;
     }
     else {
         for (int i = 2; i <= (numero / 2); i++){
             if (numero % i == 0)
                 numDivisores++;
         }
         if (numDivisores > 0)
             resultado = false;
     }

     return resultado;
 } // De la función esPrimoMejorada
```

```java
/* Declaración de la función esPrimoDelegando que toma un parámetro int
   y devuelve un boolean. Será true si el parámetro no es divisible
   entre ningún número distinto de 1 o de él mismo y false en caso contrario.
   Delega en la función numDivisores para obtener cuantos divisores tiene
   el parámetro, si son más de dos el número no es primo*/
public static boolean esPrimoDelegando(int numero) {
    boolean resultado = true;

    if (numero < 0) {
        System.out.println("\nEl número ha de ser positivo");
        resultado = false;
    }
    else
        if (numeroDivisores(numero) > 2)
            resultado = false;

    return resultado;
} // De la función esPrimoDelegando

/* Declaración de la función numeroDivisores que toma un parámetro int
   y devuelve un int con su número de divisores exacto. En ese número estarán
   incluidos el 1 y él mismo. Si el número estudiado es negativo se indica
   y devuelve cero */
public static int numeroDivisores(int numero){

    int numDivisores = 0;

    if (numero < 0) {
        System.out.println("\nEl número ha de ser positivo");

    }
    else {
        for (int i = 1; i <= numero; i++){
            if (numero % i == 0)
                numDivisores++;
        }
    }

    return numDivisores;
} // De la función numDivisores

}// De la clase
```

4º Define una función que determine si un número entero dado como parámetro es múltiplo del entero que se proporciona como segundo parámetro.

Solución:

```
class Main {
  public static void main(String args[]) {

      System.out.println();
      System.out.println("100 es múltiplo de -10 = " + esMultiplo(100,-10));
      System.out.println();
      System.out.println("15 es múltiplo de 3 = " + esMultiplo(15, 3));
      System.out.println();
      System.out.println("11 es múltiplo de 7 = " + esMultiplo(11,7));
      System.out.println();

  } // De la función main

  /* Declaración de la función esMultiplo que toma dos parámetros int
     y devuelve un boolean si el primero es multiplo del segundo. */
  public static boolean esMultiplo(int num1, int num2){
      boolean resultado = false;

      if (num1 % num2 == 0)
          resultado = true;

      return resultado;

  } // De la función esMultiplo

} // De la clase
```

5º Define una función que determine cuantos divisores tiene un número entero positivo dado como parámetro. Si el número estudiado fuera negativo se ha de indicar con un mensaje y devolver cero.

Solución:

```
class Main {
  public static void main(String args[]) {

      System.out.println();

      System.out.println("Número de divisores de -8 = " + numeroDivisores(-8));
      System.out.println("Número de divisores de 4 = " + numeroDivisores(4));
      System.out.println("Número de divisores de 13 = " + numeroDivisores(13));
  }  // De la función main

  /* Declaración de la función numeroDivisores que toma un parámetro int
     y devuelve un int con su número de divisores exacto. En ese número estarán
     incluidos el 1 y él mismo. Si el número estudiado es negativo se indica
     y devuelve cero */
  public static int numeroDivisores(int numero){

      int numDivisores = 0;

      if (numero < 0) {
          System.out.println("\nEl número ha de ser positivo");

      }
      else {
          for (int i = 1; i <= numero; i++){
              if (numero % i == 0)
                  numDivisores++;
          }

      }

      return numDivisores;

  } // De la función numDivisores

} // De la clase
```

6º Define una función que acepte un carácter como parámetro e imprima un saludo de “Buenas tardes “ más el carácter suministrado.

Solución:

```
class Main {
  public static void main(String args[]) {

      System.out.println();
      saludo('D');
      saludo('A');
      saludo('M');
      System.out.println();

  }  // De la función main

  /* Declaración de la función saludo que toma un char como parámetro
     e imprime un saludo más el char pasado. No devuelve nada. */
  public static void saludo(char inicial){
      System.out.println("Buenas tardes " + inicial);

  } // De la función saludo

} // De la clase
```

7º Define una función que tome dos *int* y realice la potencia del primer *int* elevado al segundo parámetro. Debe devolver ese resultado. Recuerda que tanto la base como el exponente pueden ser positivos o negativos y eso afecta a los resultados obtenidos.

Solución:

```
class Main {
  public static void main(String args[]) {

      System.out.println();
      System.out.println("5 al-2 = " + potencia(5,-2));
      System.out.println();
      System.out.println("2 al 0 = " + potencia(2, 0));
      System.out.println();
      System.out.println("-2 al 7 = " + potencia(-2, 7));
      System.out.println();
      System.out.println("2 al 8 = " + potencia(2, 8));
      System.out.println();
  } // De la función main

  /* Declaración de la función potencia que toma dos parámetros int
     y devuelve un double con la base elevada al exponente. */
  public static double potencia(int base, int exponente){
      double resultado = 0.0;
      long aux = 1;

      if (exponente == 0)
          resultado = 1.0;
      else
        if (exponente < 0) {
            while (exponente < 0) {
                aux = aux * base;
                exponente++;
            }
            resultado = (double) 1 / aux;
        }
        else {
            while (exponente > 0) {
                aux = aux * base;
                exponente--;
            }
            resultado = (double)aux;
        }

      return resultado;

  }// De la función potencia

}// De la clase
```