1. INTRODUCCIÓN A JAVA

PREGUNTA 1. ¿Qué es Java?

a. Un Sistema Operativo
b. Un lenguaje de programación
c. Un compilador
d. Un programa

PREGUNTA 2. SDK es un acrónimo de:

a. Simple Development Kit
b. Software Developing Kat
c. Some Delivery Kit
d. Software Development Kit

PREGUNTA 3. ¿Cuáles son las ediciones del JDK de Java?

a. J2EE, J2AA y J2II
b. J5EE, J5OO y J5UU
c. J2EE, J2SE y J2ME
d. J3XP, J3ME y J3VISTA

PREGUNTA 4. ¿Cuál es un componente del SDK de Java?

a. Java
b. Javac
c. javadoc
d. Todos los anteriores

PREGUNTA 5. ¿En qué sistema operativo se puede ejecutar una aplicación Java?

a. GNU/Linux
b. UNIX
c. Windows
d. En cualquiera que tenga instalado la JVM y/o JRE

PREGUNTA 6. El API (Interfaz de Programación de Aplicaciones) de Java

a. Se instala con la JVM
b. Se instala con el JDK

c. Se puede consultar online y descargase
d. Hay que comprarlo para acceder

PREGUNTA 7. ¿Cuál es la extensión de un archivo de código fuente escrito en Java?

a. .class
b. .java
c. .jav
d. .j2se

PREGUNTA 8. ¿Cuál es la extensión de un archivo compilado Java?

a. .jav
b. .byco
c. .class
d. .java

PREGUNTA 9. Las clases de Java se pueden empaquetar en archivos comprimidos, ¿Qué extensión tienen estos archivos?

a. .pack
b. .class
c. .jar
d. .tar

PREGUNTA 10. ¿Qué popular algoritmo de compresión utilizan los paquetes de java?

a. El mismo que los archivos RAR (Roshal Archive)
b. El mismo que los archivos ZIP
c. El mismo que los archivos Ark
d. El mismo que los archivos 7z

2.- CLASES Y OBJETOS

EJERCICIO 1

Crear una clase Saludo con un método saludar donde se muestre un mensaje de bienvenida.

Crear una clase Main y desde el método main crear una instancia de la clase Saludo e invocar al método saludar.

EJERCICIO 2

Crear la clase Animal con los atributos (propiedades) y operaciones (métodos) que se estimen oportunos.

Crear la clase TestAnimal desde donde se crearán 3 objetos: Perro, pez y cocodrilo. Modificar las propiedades de cada objeto y llamar a cada uno de los métodos mostrando el resultado por pantalla.

EJERCICIO 3

Crear la clase Areas con métodos para calcular el área de un circulo y el área de un rectángulo. Desde una clase principal realizar ambos cálculos y mostrarlos por pantalla.

3.- IDENTIFICADORES, PALABRAS CLAVE Y TIPOS

EJERCICIO 1

Crear una clase Monedas que me informe del numero de monedas y billetes que corresponderían al cambio si pago con 200 euros y el importe asciende a 163,27 euros. Intentar pagar con la moneda o billete de más valor posible.

EJERCICIO 2

Crear una variable distinta por cada tipo de dato, asignamos valores a cada una de ellas y después las mostramos por consola con mensajes indicando de que tipo es cada una.

EJERCICIO 3

Crear la clase Persona con un método que calcule la edad de la persona en días y también en segundos.

Desde el programa principal TestPersona calcular la edad de varias personas

4.- OPERADORES Y CONTROL DE FLUJO

EJERCICIOS CON IF-ELSE

EJERCICIO 1

Devolver si un numero es par o impar

EJERCICIO 2

Comparar dos números y decir cuál es el mayor

EJERCICIO 3

Comparar tres números y decir cuál es el mayor

EJERCICIO 4

Hallar una renta de C x r x t / 1200

Si el tiempo es <=24 meses --> 5%

Si el tiempo es <=60 meses --> 8%

Si el tiempo es >60 meses --> 10%

EJERCICIO 5

Ver si un número es par y múltiplo de 3 y que saque mensajes del tipo:

Es par y múltiplo de 3

Es impar y múltiplo de 3

No es múltiplo de 3 pero es par

No es ni par ni múltiplo de 3

EJERCICIO 6

Se va a subir el sueldo a los empleados de la siguiente manera:

Los que cobren < 15000 se les aumenta el 15%

Los que cobren >= 15000 aumento del 12%

además:

Si son mujeres tienen un 2% adicional

Si tienen hijos tienen un 1% adicional por hijo

EJERCICIOS CON SWITCH-CASE

EJERCICIO 1

Con un número del 1 al 7 decir a que día de la semana corresponde

EJERCICIO 2

Con una nota calificarlo de suspenso, aprobado, ...etc.

EJERCICIO 3

Con varias estructuras switch ... case ver si un número es divisible por 2,3 o por 5

EJERCICIO 4

Con una letra que contiene la variable estado decir el estado civil de la persona

EJERCICIO 5

Con una letra que contiene un numero romano:

I, V, X, L, C, D, M (incluso minúsculas)

Decir a que numero corresponde

EJERCICIOS BUCLE FOR

EJERCICIO 1

Escribir los 10 primeros números (ascendente y descendente)

EJERCICIO 2

Escribir "Feliz Navidad 1990" hasta 2007

EJERCICIO 3

Escribir la tabla de multiplicar del 5

EJERCICIO 4

Escribir el factorial del numero 15

EJERCICIO 5

Hallar 2 elevado a 8

EJERCICIO 6

Hallar los números impares menores de 30 y que aparezcan en orden descendente

EJERCICIO 7

Hallar todas las tablas de multiplicar con for anidados

EJERCICIO 8

Con los 20 primeros números decir con cada uno por cual es divisible:

ejemplo: 1 --> es divisible por 1

2 --> es divisible por 1

es divisible por 2 ...

EJERCICIO 9

Hallar la lista de los números primos hasta el 100

EJERCICIOS BUCLE WHILE

EJERCICIO 1

Calcular el factorial de 25

EJERCICIO 2

Calcular la potencia de 2 elevado a 8

EJERCICIO 3

Calcular los 25 primeros números primos, NO DEL 1 AL 25

EJERCICIO 4

Encontrar el primer número primo a partir de 198

EJERCICIO 5

Un numero es perfecto cuando todos sus divisores sumados dan el mismo numero

Ejemplo: 6 = 1 + 2 + 3; luego 6 es un numero perfecto

Mostrar los números perfectos hasta el 100

EJERCICIO 6

Calcular los n primeros términos de la serie de Fibonacci

EJERCICIOS BUCLE DO-WHILE

EJERCICIO 1

Resolver la potencia de 2 elevado a 8 con do-while

EJERCICIO 2

Resolver la serie de Fibonacci con do-while

EJERCICIO 3

Calcular la suma de los números pares comprendidos entre 10 y 50

5.- ARRAYS

EJERCICIO 1

Crear un bloque de código que pinte las vocales y solo las vocales que existen en un array de caracteres. El programa debe de ir comprobando, con un bucle, para determinar si cada carácter del array es o no una vocal.

EJERCICIO 2

Crear un bloque de código que lea las componentes de un array de enteros y me pinte cuales con pares, cuales impares y cuales con múltiplos de tres. Para este ejercicio utilizar un bucle que recorra el array y utilizar el operador %.

EJERCICIO 3

Crear un bucle que pinte por consola todas las componentes de un array en orden inverso a como están guardadas en el array

EJERCICIO 4

Crear un bloque de código que recorra los siguientes arrays y me cree otro que contenga las componentes de ambos dos. Los arrays son los siguientes:

```
int[] array1 = new int[]{1,2,3,4,5};

int[] array2 = new int[]{334,23,4};
```

El array resultado es el siguiente:

```
int[] array3 = new int[]{1,2,3,4,5,334,23,4};
```

EJERCICIO 5

Dado los siguientes arrays, hacer un bloque de código que construya un array de la siguiente manera:

```
char[] cars1 = new char[]{'1','2','3','4','5','6'};
```

```
char[] cars2 = new char[]{'a','e','r','t','y','u'};
```

El array resultado que se debe de construir es el siguiente:

```
char[] result = new char[]{'1','a','2','e','3','r','4','t','5','y','6','u'};
```

EJERCICIO 6

Dado el siguiente array, crear un bloque de código que pinte, SOLO, las consonantes que existan en dicho array

```
char[] letras = new char[]{'2','f','f','u','u','g','h','i','4'};
```

EJERCICIO 7

Dado el siguiente array crear un bloque de código que sume todas las componentes del array, para ello hacer uso de un bucle que vaya obteniendo todos los valores del array

```
float[] decimales = new float[]{3.4F,5.67F,12.0F,3.141615F,0.0F};
```

EJERCICIO 8

Dado el siguiente array de números, crear un bloque de código que sume por un lado solo los números pares y devuelva el resultado y por otro solo los impares y también pinte el resultado

```
int[] numeros = new int[]{1,2,7,3,4,65,23,78,98,34,342,45,57};
```

EJERCICIO 9

Dado los dos siguientes arrays, crear un bloque de código que me sume las Componentes de ambos arrays de la siguiente manera:

Array1	Array2	Resultado		Resultado
1 2	3 4	(1+3) (2+4)	--->	4 6
0 5	5 8	(0+5) (5+8)		5 13

Los arrays son los siguientes:

```
int[][] matriz1 = new int[][]{ {1,2}, {0,5} };

int[][] matriz2 = new int[][]{ {3,4}, {5,8} };
```

EJERCICIO 10

Dado la siguiente matriz, crear un bloque de código que me pinte sus componentes de la siguiente manera:

{3 4 5 78}

{0 0 0 0}

{1 1 1 1}

{6 6 6 -1}

El array bidimensional es el siguiente:

```
int[][] numeros = new int[][]{ {3, 4, 5, 78},
                               {0, 0, 0,  0},
                               {1, 1, 1,  1},
                               {6, 6, 6, -1} };
```

6.- CLASES AVANZADAS

EJERCICIOS DE HERENCIA

EJERCICIO 1

Crear la clase Figura encapsulada con las siguientes propiedades: coordenadaX y coordenadaY (ambas de tipo entero) y un método posición() que devuelva la posición de la figura.

Crear las clases encapsuladas Círculo y Rectángulo heredando de la clase figura.

La clase Círculo además tendrá la propiedad radio de tipo double y la clase Rectángulo tendrá el ancho y el alto del rectángulo también de tipo double.

Sobrescribir el método posición() para cada una de las clases y utilizar el operador super para llamar al método sobrescrito.

Crear la clase TestFiguras donde se crearán objetos de las tres clases, se darán valores y mostraremos la posición de cada figura.

EJERCICIO 2

Crear la clase Vehiculo con las propiedades, métodos y constructores que creas conveniente.

Crear las clases Coche, Barco, Avión y Tren heredando de Vehiculo. Debes añadir aquellas propiedades y métodos que creas conveniente así como sobrescribir aquellos métodos que consideras necesario.

Por ultimo crea una clase TestVehiculos donde instancias todas las clases y procedas a llamar a sus métodos.

EJERCICIOS DE CLASES ABSTRACTAS

EJERCICIO 1

Crear la clase abstracta Planta con un método abstracto:

```
void regar(int cantidad, int tiempo)
```

además crearemos todas propiedades y métodos que se os ocurran

Creamos las clases PlantaTropical, Cactus donde implementaremos el método abstracto.

Desde una clase principal crear objetos utilizando polimorfismo.

EJERCICIO 2

Crear la clase abstracta Animal con los métodos abstractos comer() y moverse(). A continuación crearemos varias clases Perro, Pez, Pajaro, Gusano, …etc.

- ¿Qué ocurre si no implemento todos los métodos?
- ¿Puedo crear instancias igualmente?
- ¿Puedo modificar parte de la definición de los métodos abstractos?
- ¿Puedo utilizar polimorfismo?

Crea clases extendiendo la clase Pajaro, por ejemplo Gorrión y Buitre.

- ¿También son abstractas estas clases?

EJERCICIOS DE INTERFACES

EJERCICIO 1

Creamos la interfaz ArticuloVenta con los siguientes métodos:

getPrecio() : float

getProveedor() : String

Creamos la clase Ropa con las propiedades y métodos que preciséis.

Creamos la clase Pantalón, Camisa, Zapatos, …etc. heredando de la clase Ropa e implementando la interfaz ArticuloVenta.

Aprovechar a repasar lo visto, sobreescritura de métodos, adaptación de los métodos de la clase Object: equals() y toString().

En una clase principal crearemos distintos objetos y mostraremos toda la información que se estime oportuna utilizando polimorfismo.

EJERCICIO 2

Vamos a ampliar el ejercicio anterior creando otra interfaz ArticuloPeredecero con el método: caducar()

Podemos crear otras clases Pan, Leche, …etc.

Tiene sentido crear dichas clases implementando ambas interfaces(ArticuloVenta y ArticuloPeredecero)

Contesta a las siguientes preguntas:

- Estamos obligados a implementar todos los métodos?
- Qué ocurre si no los implementamos?
- Puedo cambiar el modificador de acceso a los métodos?
- Puedo añadir aparte los métodos que yo quiera?
- Puedo seguir trabajando con polimorfismo?

Dibuja en tu cuaderno como sería el diagrama UML resultante.

EJERCICIOS ESTATICOS

EJERCICIO 1

Crear la clase globo con una variable estática entera para asignar la referencia del globo.

En una clase principal PruebaGlobo se crearán distintos objetos y desde cada uno de ellos se mostrará la referencia del globo.

EJERCICIO 2

Crear una clase Calculadora con métodos estáticos: suma, resta, multiplica y divide.
Desde una clase principal PruebasCalculadora invocar a los cuatro métodos.

EJERCICIOS TIPOS ENUMERADOS

EJERCICIO 1

Crear un tipo de dato enumerado para definir el estado civil de una persona. Los valores que podrá tomar serán: SOLTERO, CASADO, VIUDO y DIVORCIADO.

Lo vamos a crear aparte (fuera de cualquier clase) y de momento será muy simple.

Crearemos la clase Persona con sus propiedades y métodos. Utilizaremos el tipo enumerado creado anteriormente para definir el estado civil de la persona. Además sobrescribiremos el método toString() y el método equals(Object o).

En una tercera clase TestPersona, crearemos diferentes objetos de Persona y mostraremos la información relativa a cada una.

EJERCICIO 2

Vamos a crear una segunda versión del ejercicio anterior.

El tipo enumerado lo vamos a completar de la siguiente manera.

Por cada valor definiremos un alias para luego mostrarlo por pantalla según el ejemplo

Crearemos una propiedad para almacenar ese alias.

Crearemos un constructor que asigne el alias recibido como parámetro a la propiedad

Crearemos un método que devuelva el alias.

Ejemplo: DIVORCIADO (“Persona Divorciada”)

EJERCICIOS CLASES ENVOLVENTES

EJERCICIO 1

Hacer un método que reciba una cadena y me devuelva un array de caracteres que contiene todos los caracteres de la cadena que recibe.

EJERCICIO 2

Hacer un método que reciba un número y me devuelva un String que es la representación en binario del número que recibe.

EJERCICIO 3

Hacer un método que reciba un cadena y me devuelva otra que sea la misma cadena en minúsculas

EJERCICIO 4

Hacer un método que reciba un cadena y me devuelva otra que sea la misma cadena en mayúsculas

EJERCICIO 5

Hacer un método que reciba un carácter y me devuelva un String que contiene ese carácter, es decir, recibe 'a' y devuelve "a"

EJERCICIO 6

Hacer un método que reciba un float y me devuelva un cadena que es la representación de ese float. Por ejemplo: recibe 12.78F y devuelve "12.78"

EJERCICIO CONSULTA API

EJERCICIO 1

Crear la cadena de texto: “Bienvenidos al curso de Java”

- Mostrar la cadena toda en mayúsculas
- Mostrar la cadena en minúsculas
- Mostrar solo la palabra Bienvenidos
- Mostrar solo la palabra Java
- Mostrar la palabra curso
- Mostrar la posición de la letra c
- Devolver la longitud de la cadena
- Modificar las e minúsculas por E mayúsculas
- Concatenar la frase anterior con “con fecha 14 de Septiembre”

7.- EXCEPCIONES Y ASERCIONES

EJERCICIO 1

Crearemos la clase Producto con las siguientes propiedades: ID, nombre, precio, proveedor y descripción.

Nos vamos a crear múltiples constructores.

Dicha clase tendrá un método comprobar(). Si el producto no tiene asignado un ID lanzaremos la excepción ProductoException que evidentemente nos crearemos nosotros.

También comprobaremos mediante aserciones que el precio introducido es mayor que 0.

EJERCICIO 2

Manejar la excepción generada en el ejercicio anterior de dos formas:

- Hacer una versión que la maneje con un bloque try – match
- Que sea el método quien lance la excepción.

EJERCICIO 3

Crear una calculadora segura que me permita operar (sumar, restar, multiplicar y dividir) con datos tanto enteros o decimales. Capturar la posible excepción que se puedan dar al dividir por 0.

8.- COLECCIONES Y GENERICOS

EJERCICIO 1

Cree una clase que tenga un objeto de la clase HashSet, que añada varias cadenas al objeto e imprima todas las cadenas. Utilizar el tipo genérico String.

EJERCICIO 2

Cree un objeto de la clase ArrayList, añada una lista de cadenas y utilice un iterador para imprimir las cadenas.

EJERCICIO 3

Cree un programa almacene números del 0 al 9 en un array y en una lista, e imprima sus valores.

EJERCICIO 4

Cree un programa que almacene la nota de una asignatura de varios alumnos en una colección de tipo Map. El alumno se identificará por su nombre.

Cada elemento del mapa estará formado por nombre de tipo String y nota de tipo Integer. Utilizar tipos genéricos.

EJERCICIO 5

Cree un programa que implemente un diccionario. El diccionario debe contener palabras en castellano y una lista de posibles significados de cada palabra.

Crear la clase Palabra que permita almacenar la información de una palabra, junto con sus definiciones.

Utilizar un objeto HashMap para almacenar palabras en la clase Diccionario.