1 Una clase es lo mismo que un objeto

a) Verdadero

b) Falso

2 Las clases permiten que el programador modele objetos

que contienen atributos y comportamientos

a) Verdadero

b) Falso

3 En C# cuando se declara un objeto, este se crea automaticamente

a) Verdadero

b) Falso

4 Solo los metodos pueden ser declarados public

a) Verdadero

b) Falso

5 Solo los atributos pueden ser declarados private

a) Verdadero

b) Falso

6 Cuando se declara un dato o funcion despues de en la clase, este puede ser visible para cualquier funcion que tenga acceso a dicha clase

a) Verdadero

b) Falso

7 El constructor se invoca cuando se crea una instancia de una clase

a) Verdadero

b) Falso

8 El constructor se invoca cuando se crea un objeto de de una clase

a) Verdadero

b) Falso

9 Los datos o funciones privados de una clase son visibles desde fuera de la clase

a) Verdadero

b) Falso

10 Solo los miembros de una clase tienen acceso directo a los atributos privados

a) Verdadero

b) Falso

11 Si se declara un atributo privado, puede ser inicializado desde fuera de la clase con el operador de asignacion de igualdad (=)

a) Verdadero

b) Falso

12 Solo los metodos pueden ser declarados private

a) Verdadero

b) Falso

13 Solo los atributos pueden ser declarados public

a) Verdadero

b) Falso

14 Un arreglo NO puede ser un atributo de una clase

a) Verdadero

b) Falso

15 Los constructores deben prototiparse

a) Verdadero

b) Falso

16 Los destructores deben prototiparse

a) Verdadero

b) Falso

17 Se refiere a "quitar" propiedades y metodos de un objeto y quedarse solo con aquellos que sean necesarios (relevantes para el problema a solucionar)

a) Herencia

b) Abstraccion

c) Encapsulamiento

d) Sobrecarga

18 Se centra en las caracteristicas esenciales de algun objeto, en relacion a la perspectiva del observador

a) Herencia

b) Abstraccion

c) Encapsulamiento

d) Sobrecarga

19 Es una plantilla, plano o definicion que contiene la descripcion de las caracteristicas y acciones que se desean modelar

a) Objeto

b) Clase

c) Metodo

d) Propiedad

20 A partir de una clase se pueden crear muchos objetos independientes con las mismas caracteristicas

a) Verdadero

b) Falso

21 Es una unidad o elemento que combina datos y funciones

a) Objeto

b) Clase

c) Metodo

d) Propiedad

22 Es creado a partir de una clase

a) Objeto

b) Atributo

c) Metodo

d) Propiedad

23 Son las caracteristicas o datos de los objetos

a) Funciones

b) Procedimientos

c) Metodos

d) Atributos

24 Son las acciones u operaciones a realizar por un objeto

a) Clases

b) Caracteristicas

c) Metodos

d) Atributos

25 Se refiere al valor de las caracteristicas de un objeto

a) Estado

b) Comportamiento

c) Identidad

26 Se refiere a las acciones que puede realizar un objeto

a) Estado

b) Comportamiento

c) Identidad

27 Se refiere a que un objeto tiene un nombre unico y pertenece a una clase

a) Estado

b) Comportamiento

c) Identidad

28 Es la instancia de una clase

a) Objeto

b) Atributo

c) Metodo

d) Propiedad

29 Permite incluir en una sola entidad la informacion y operaciones que manipulan dicha informacion

a) Herencia

b) Abstraccion

c) Encapsulamiento

d) Sobrecarga

30 Permite "eliminar" caracteristicas y acciones irrelevantes para el problema que se desea modelar

a) Herencia

b) Abstraccion

c) Encapsulamiento

d) Sobrecarga

31 Permite definir restricciones de acceso (como publico oprivado) a los componentes de los objetos

a) Herencia

b) Abstraccion

c) Encapsulamiento

d) Sobrecarga

32 Permite ocultar los detalles de implementacion de los objetos

a) Herencia

b) Abstraccion

c) Encapsulamiento

d) Sobrecarga

33 En C# cuando se declara un objeto, automaticamente se crea una instancia de la clase a la que pertenece

a) Verdadero

b) Falso

34 Se utiliza para crear un objeto y/o inicializar su estado

a) Constructor

b) Destructor

35 Se utiliza para liberar el estado de un objeto

a) Constructor

b) Destructor

36 Capacidad para utilizar caracteristicas previstas en antepasados o ascendientes

a) Herencia

b) Abstraccion

c) Encapsulamiento

d) Sobrecarga

37 Permite crear nuevas clases a partir de otras ya existentes, permitiendo que estas transmitan sus propiedades

a) Herencia

b) Abstraccion

c) Encapsulamiento

d) Sobrecarga

38 La clase base tambien se conoce como super clase

a) Verdadero

b) Falso

39 La clase base tambien se conoce como clase madre

a) Verdadero

b) Falso

40 La clase base tambien se conoce como clase derivada

a) Verdadero

b) Falso

41 Una clase derivada hereda de una clase base

a) Verdadero

b) Falso

42 Una clase base hereda de una clase derivada

a) Verdadero

b) Falso

43 A las clases derivadas tambien se les conoce como

clases hijas

a) Verdadero

b) Falso

44 Una subclase es una clase derivada

a) Verdadero

b) Falso

45 Una subclase hereda el comportamiento y la estructura

de una clase base

a) Verdadero

b) Falso

46 En C# se permite la herencia simple

a) Verdadero

b) Falso

47 En C# se permite la herencia multiple

a) Verdadero

b) Falso

48 Se pueden generar instancias de una clase abstracta

a) Verdadero

b) Falso

49 Se pueden crear objetos de una clase abstracta

a) Verdadero

b) Falso

50 Se pueden crear objetos de una clase derivada de una clase abstracta

a) Verdadero

b) Falso

Ejercicios

1 Crear una clase Libro que contenga los siguientes atributos:

– ISBN

– Titulo

– Autor

– Número de páginas

Crear sus respectivos métodos get y set correspondientes para cada atributo.

Crear el método toString() para mostrar la información relativa al libro con el siguiente formato:

“El libro con ISBN creado por el autor tiene páginas”

En el fichero main, crear 2 objetos Libro (los valores que se quieran) y mostrarlos por pantalla.

Por último, indicar cuál de los 2 tiene más páginas.

2 Vamos a realizar una clase llamada Raices, donde representaremos los valores de una ecuación de 2º grado. Tendremos los 3 coeficientes como atributos, llamémosles a, b y c.  
Hay que insertar estos 3 valores para construir el objeto.  
Las operaciones que se podrán hacer son las siguientes:

-obtenerRaices(): imprime las 2 posibles soluciones

-obtenerRaiz(): imprime única raíz, que será cuando solo tenga una solución posible.

-getDiscriminante(): devuelve el valor del discriminante (double), el discriminante tiene la siguiente formula, (b^2)-4\*a\*c

-tieneRaices(): devuelve un booleano indicando si tiene dos soluciones, para que esto ocurra, el discriminante debe ser mayor o igual que 0.

-tieneRaiz(): devuelve un booleano indicando si tiene una única solución, para que esto ocurra, el discriminante debe ser igual que 0.

-calcular(): mostrara por consola las posibles soluciones que tiene nuestra ecuación, en caso de no existir solución, mostrarlo también.

Formula ecuación 2º grado: (-b±√((b^2)-(4\*a\*c)))/(2\*a)  
Solo varia el signo delante de –b

3 Queremos representar con programación orientada a objetos, un aula con estudiantes y un profesor.  
Tanto de los estudiantes como de los profesores necesitamos saber su nombre, edad y sexo. De los estudiantes, queremos saber también su calificación actual (entre 0 y 10) y del profesor que materia da.  
Las materias disponibles son matemáticas, filosofía y física.  
Los estudiantes tendrán un 50% de hacer novillos, por lo que si hacen novillos no van a clase pero aunque no vayan quedara registrado en el aula (como que cada uno tiene su sitio).  
El profesor tiene un 20% de no encontrarse disponible (reuniones, baja, etc.)  
Las dos operaciones anteriores deben llamarse igual en Estudiante y Profesor (polimorfismo).  
El aula debe tener un identificador numérico, el número máximo de estudiantes y para que esta destinada (matemáticas, filosofía o física). Piensa que más atributos necesita.  
Un aula para que se pueda dar clase necesita que el profesor esté disponible, que el profesor de la materia correspondiente en el aula correspondiente (un profesor de filosofía no puede dar en un aula de matemáticas) y que haya más del 50% de alumnos.  
El objetivo es crear un aula de alumnos y un profesor y determinar si puede darse clase, teniendo en cuenta las condiciones antes dichas.  
Si se puede dar clase mostrar cuantos alumnos y alumnas (por separado) están aprobados de momento (imaginad que les están entregando las notas).  
NOTA: Los datos pueden ser aleatorios (nombres, edad, calificaciones, etc.) siempre y cuando tengan sentido (edad no puede ser 80 en un estudiante o calificación ser 12).

4 Nos piden hacer un programa orientado a objetos sobre un cine (solo de una sala) tiene un conjunto de asientos (8 filas por 9 columnas, por ejemplo).  
Del cine nos interesa conocer la película que se está reproduciendo y el precio de la entrada en el cine.

De las películas nos interesa saber el título, duración, edad mínima y director.

Del espectador, nos interesa saber su nombre, edad y el dinero que tiene.

Los asientos son etiquetados por una letra (columna) y un número (fila), la fila 1 empieza al final de la matriz como se muestra en la tabla. También deberemos saber si está ocupado o no el asiento.

8 A 8 B 8 C 8 D 8 E 8 F 8 G 8 H 8 I  
7 A 7 B 7 C 7 D 7 E 7 F 7 G 7 H 7 I  
6 A 6 B 6 C 6 D 6 E 6 F 6 G 6 H 6 I  
5 A 5 B 5 C 5 D 5 E 5 F 5 G 5 H 5 I  
4 A 4 B 4 C 4 D 4 E 4 F 4 G 4 H 4 I  
3 A 3 B 3 C 3 D 3 E 3 F 3 G 3 H 3 I  
2 A 2 B 2 C 2 D 2 E 2 F 2 G 2 H 2 I  
1 A 1 B 1 C 1 D 1 E 1 F 1 G 1 H 1 I

Realizaremos una pequeña simulación, en el que generaremos muchos espectadores y los sentaremos aleatoriamente (no podemos donde ya este ocupado).  
En esta versión sentaremos a los espectadores de uno en uno.

Solo se podrá sentar si tienen el suficiente dinero, hay espacio libre y tiene edad para ver la película, en caso de que el asiento este ocupado le buscamos uno libre.  
Los datos del espectador y la película pueden ser totalmente aleatorios.

5 Problema 136 - Números Feos \

Los números feos son números cuyos únicos factores primos son 2, 3, o 5. La secuencia 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, .... Muestra los primeros 11 números feos. Por convención se incluye el 1. Esto equivale a encontrar los números que se pueden formar por 2a3 b5 c . Escriba un programa que encuentre e imprima el número 1500. Entrada y salida.- No existe entrada en este problema. La salida debe consistir de una línea reemplazando número con el número calculado. Ejemplo de salida The 1500’th ugly number is n´umero.