

UD05.EXAME Práctico - REC

DAM1-Programación 2022-23

03/03/2023

1. Examen Test (5)

1

2. Examen Evaluable (5)

3

- Puedes utilizar apuntes y materiales que consideres pero deberás realizar los programas individualmente. En caso contrario se retirará el examen.
- Realiza programas bien estructurados, legibles, con comentarios, líneas en blanco, identificadores adecuados, etc.
- Cuida la interacción con el usuario, presentando la información de forma clara y ordenada.
- Nombra los proyectos y ficheros con nombres adecuados.
- Indica la autoría del código incluyendo un comentario con tu nombre y apellidos.

1. Descarga el proyecto Java del repositorio (o donde te indique el profesor) y configúralo como un nuevo proyecto Java en tu entorno de desarrollo.
2. En la carpeta /src crea las nuevas clases que se solicitan a continuación.
3. **Incluye un comentario con tu nombre y apellidos al inicio de cada clase.**
4. **Entrega la carpeta /src** del proyecto comprimida.
5. **Tiempo máximo:** 2 sesiones
6. La puntuación máxima de cada ejercicio se indica entre paréntesis.

1. Examen Test (5)

Implementa un modelo de clases en Java que permita la creación de exámenes tipo test. El modelo debe incluir las siguientes clases:

1. La clase "**Pregunta**", que representará una pregunta de tipo test con varias opciones de respuesta y una única opción correcta, y que tendrá los siguientes
 - a. atributos:
 - i. Enunciado de la pregunta
 - ii. Lista de respuestas
 - iii. Índice de la respuesta correcta
 - b. métodos:
 - i. un constructor que validará que la pregunta tenga al menos 2 respuestas diferentes y un índice de opción correcta válido, es decir, dentro de los límites del número de respuestas..
 - ii. **corregir()** que recibe el índice de la opción elegida y devuelve true si es la opción correcta y false si no lo es.
 - iii. el método redefinido **toString()**, que devolverá un String con el enunciado de la pregunta y en las siguientes líneas las posibles respuestas precedidas de las letras consecutivas del alfabeto, es decir, a), b), c), d)...
 - iv. el método redefinido **equals()** considerará iguales dos preguntas que tengan el mismo enunciado y la misma lista de respuestas.
2. La clase "**Examen**", que contendrá los siguientes
 - a. atributos:
 - i. un título del examen
 - ii. un conjunto de preguntas y los siguientes métodos:
 - b. métodos:
 - i. Un constructor que permita indicar el título.
 - ii. **addPregunta()** añade una pregunta al examen comprobando que no se haya añadido ya una igual.
 - iii. **numPreguntas()** devuelve el número de preguntas del examen.
 - iv. el método redefinido **toString()**, que devolverá un String con el título del examen y todas las preguntas del examen numeradas, con su enunciado y sus respuestas pero sin indicar la opción correcta.

Añade el código necesario para que funcione el programa principal de la clase **App1Examen.java** y produzca la siguiente salida por consola:

```
EXÁMENES TIPO TEST
=====

EXAMEN TEST - POO básica en Java: Clases y Objetos

1. Dos clases se consideran vecinas siempre y cuando:
  a) Sean visibles.
  b) Ambas dispongan del mismo número de constructores.
  c) Pertenezcan al mismo paquete.
```

- d) Todo lo anterior ha de cumplirse para que dos clases sean vecinas.
2. La ocultación de atributos puede definirse como:
- a) El proceso en el que un atributo pasa de ser público a privado.
 - b) El proceso en el que se define una variable local (en un método) con el mismo identificador que un atributo.
 - c) El proceso en el que un atributo estático deja de serlo.
 - d) Todas las respuestas anteriores son correctas.
3. Un miembro cuyo modificador de acceso es private será visible desde:
- a) Todas las clases vecinas.
 - b) Todas las clases externas.
 - c) Es indistinto el paquete, pero será visible siempre que se importe la clase que lo contiene.
 - d) Ninguna de las respuestas anteriores.
4. Si desde un constructor queremos invocar a otro constructor de la misma clase, tendremos que usar:
- a) `set()`.
 - b) `get()`.
 - c) `this()`.
 - d) `this`.

2. Examen Evaluable (5)

Crea la clase **ExamenEvaluable** que extiende la clase **Examen** añadiendo los siguientes

1. métodos:
 - a. **responderConsola()** que interactuará con el usuario mostrando cada pregunta del examen y recogiendo la respuesta del usuario para continuar con la siguiente pregunta y así sucesivamente hasta que se acaben. El método deberá detectar respuestas inválidas del usuario (no confundir con respuestas incorrectas) y en esos casos ofrecer al usuario repetir la entrada.
 - b. **corregir()** que recibirá un listado de respuestas y devolverá el número de respuestas correctas.
 - c. **puntuacionSobre10()** que recibirá el número de respuestas correctas y devolverá la puntuación final sobre 10 que podrá tener decimales y consistirá en la proporción entre las respuestas correctas y el número de preguntas del examen.

1. Añade el código necesario para que funcione el programa principal de la clase **App2ExamenEvaluable.java** y produzca la siguiente salida por consola:

```
EXÁMENES TIPO TEST CON CORRECCIÓN
=====

EXAMEN TEST - POO básica en Java: Clases y Objetos

1. Dos clases se consideran vecinas siempre y cuando:
  a) Sean visibles.
  b) Ambas dispongan del mismo número de constructores.
  c) Pertenezcan al mismo paquete.
  d) Todo lo anterior ha de cumplirse para que dos clases sean vecinas.

Respuesta? c

(...)

RESULTADOS:
* Respuestas correctas: 3/4
* Puntuación Final (sobre 10): 7,5
```

Notas:

- **c** equivale a la entrada introducida por el usuario.
- **(...)** equivale al resto de preguntas del examen test.