

Práctica 2

“Generador automático de exámenes”

1. Objetivos

El objetivo es que el alumno, una vez asimilados los conocimientos básicos durante la realización de la práctica 1, aplique conocimientos avanzados de POO vistos en las clases de teoría.

Mediante esta práctica se desea terminar de afianzar los conocimientos básicos de POO e introducir nuevos de carácter avanzado (herencia y polimorfismo). El alumno necesitará realizar un razonamiento de qué técnicas avanzadas de POO debe utilizar con el fin de realizar un código lo más eficiente y que permita el mejor mantenimiento posible.

2. Parte I (8 puntos):

Se trata de realizar un programa que permita crear, almacenar e imprimir exámenes de convocatorias de asignaturas. La funcionalidad básica será:

- Dar de alta preguntas.
- Dar de alta exámenes.
- Ver exámenes guardados.
 - o Imprimir sin respuestas
 - o Imprimir con respuestas

Lo primero que realizará el programa será solicitar que se desea hacer:

Crear pregunta.
Crear examen.
Ver exámenes guardados.
Salir del programa.

La práctica deberá estar preparada para tener varias asignaturas, pero para hacerla más fácil obviaremos el alta de las mismas, por tanto, estableceremos que hay 2 asignaturas por defecto:

- **Código:**POO, **Título:**Programación Orientada a Objetos.
- **Código:**DADM, **Título:**Desarrollo de aplicaciones para dispositivos móviles.

2.1. Crear pregunta:

En el caso de alta de pregunta, se almacenará (Tipo de pregunta: Teórica, Test, Desarrollo de código y Rellenar), texto de la pregunta, nota de la pregunta. Por lo que lo primero que haremos será preguntar el tipo de pregunta y una vez seleccionado esto, le solicitaremos el resto de los datos genéricos comentados.

Además de estos datos habrá que recuperar también los datos específicos de cada pregunta, los cuales se especifican en la sección 2.6 del presente documento.

2.2. Crear examen:

En el caso de alta de examen, lo primero que le hará es solicitar el nombre de la persona que va a generar el examen, después mostrará un listado de las asignaturas guardadas para que se seleccione una. Posteriormente seleccionará la convocatoria (junio, septiembre o diciembre) y el curso (por ejemplo 2021-2022). La fecha de realización se establecerá directamente con el día actual.

Una vez introducidos esos datos le solicitará al usuario que tipo de examen quiere (teórico, test, práctico o mixto). Si se selecciona teórico el examen contendrá solo preguntas teóricas, entre las almacenadas para esa asignatura. Si se selecciona test solo contendrá preguntas de tipo test y rellenar. Si se selecciona práctico el examen solo tendrá preguntas de tipo desarrollo / práctico y escribir el resultado de código. Y si se selecciona mixto contendrá todo tipo de preguntas.

Evidentemente la selección de estas asignaturas será aleatoria entre la totalidad de preguntas que cumplen el criterio.

Sea cual sea el tipo de examen elegido, se solicitará el número de preguntas que tendrá el examen. Teniendo en cuenta lo siguiente:

- Si el examen es de tipo mixto existirá una pregunta de tipo Desarrollo / práctico con valor 4 sobre 10, el resto de preguntas podrán ser de cualquier tipo y se repartirán la nota restante (6) por igual entre las mismas.
- Para cualquier otro tipo de examen el valor de cada pregunta se repartirá por igual entre todas las preguntas.

2.3. Ver exámenes guardados:

Desde esta opción se listarán los exámenes guardados en la aplicación, se mostrarán los siguientes campos:

Fecha – RealizadoPor – Asignatura – Convocatoria - Curso - Nº Preguntas

...

Una vez listados se preguntará al usuario si desea seleccionar alguno de ellos para imprimirlo por pantalla. En caso afirmativo, se le solicitará si lo quiere con las respuestas o por el contrario lo desea en modo examen real. En el primer caso mostrará las respuestas y en el segundo caso lo hará solo con las preguntas como si fuera un examen real.

2.4. Definiciones

A continuación, se detallan los actores principales que aparecen en el programa:

- **Asignatura:** Almacenará los datos de cada asignatura de la cual se puede generar un examen. Los datos a almacenar son:
 - Código
 - Título
 - Listado de preguntas de esa asignatura
- **Examen:** Almacenará los datos de cada examen generado. Los datos a almacenar son:
 - Fecha de realización: xxxx
 - Realizado por: xxxxx
 - Asignatura: (al imprimir mostrar código y título de la asignatura)
 - Convocatoria: Junio, septiembre o diciembre.
 - Curso: 2021-2022
 - Tipo examen: teórico, test, práctico o mixto
 - Preguntas: Listado de preguntas del examen.
- **Pregunta:** Almacenará los datos de cada pregunta dada de alta. Los datos a almacenar son:
 - Texto de la pregunta: Pregunta a realizar.
 - Texto aclaratorio.
 - Nota: Nota numérica de la pregunta.
- **Pregunta teórica:** Pregunta – respuesta.
 - o Respuesta correcta – *Evidentemente esta no se imprimirá al imprimir el examen.*
- **Pregunta tipo test:** Varias opciones de contestación.
 - o Indicar si pregunta fallada resta.
 - o Tipos de preguntas de Test
 - **Pregunta de verdadero/falso:**
 - Resultado: Verdadero / falso.
 - **Pregunta de listado de opciones (una o varias correctas)**
 - Número de respuestas.
 - Listado de respuestas posibles.
 - o Respuesta
 - o Opción correcta: Si / No. – *Evidentemente esta no se imprimirá al imprimir el examen.*
- **Pregunta de rellenar:** Texto con x palabras que rellenar.
 - o Frase de la respuesta, que contendrá x caracteres '?'. Cada uno de ellos deberá ser completado al responder el examen.

- Listado de palabras correctas por orden de aparición en la frase de respuesta. – *Evidentemente esta no se imprimirá al imprimir el examen.*
- **Pregunta de Desarrollo / Práctico:** Pregunta – desarrollo de código. Tiene varios apartados evaluados parcialmente respecto a la nota final de la pregunta.
 - Listado de apartados del código a desarrollar.
 - Texto del apartado.
 - % sobre la nota de la pregunta.

3. Consideraciones a tener en cuenta

Para estandarizar la impresión de los datos de cualquier clase se requiere la utilización de una interfaz que llamaremos `Imprimible` que tendrá un método `imprimirSimple` y otro `imprimirCompleto` que realizará la impresión de dichas clases en un formato amigable bien en formato simple con datos reducidos o bien en formato completo con todos los datos. En el caso de las preguntas el formato de impresión simple no tendrá las respuestas y el de impresión completo si las tendrá.

Los datos almacenados en la práctica deben persistirse en fichero, es decir, no se pueden perder los datos entre una ejecución y otra de la práctica. En clase ya vimos los métodos para serializar y deserializar objetos mediante GSON, y métodos para leer y escribir en fichero. También se puede utilizar el interfaz `Serializable` para guardar los datos como array de bytes. En cualquier caso, los ficheros se guardarán en el directorio **files** de la raíz del proyecto. Para dicha persistencia se sugiere utilizar una clase por cada clase a persistir o bien una clase con todo lo necesario para ser persistido. En función de la opción elegida tendremos 1 fichero o 2 de persistencia. En la entrega se deberá incluir el fichero de persistencia para facilitar la corrección de la misma.

Al utilizar polimorfismo, en algunos casos necesitaremos comprobar la clase “real” del objeto, para ello podemos utilizar el método `getClass()` de la siguiente forma:

```
If (caja.getClass().getName().equals("Caja")
```

o también mediante la instrucción **instanceof** de la siguiente forma:

```
If (caja instanceof Caja)
```

4. Parte II (2 puntos)

Para obtener la puntuación máxima de la práctica se deberá realizar una ampliación que consiste en implementar algunos de los siguientes puntos (hasta sumar 2 puntos):

- (0,5 puntos) Utilizar la librería gratuita `iText` generar los exámenes en formato pdf. En esta ruta puede encontrar un ejemplo de utilización <http://www.java4s.com/core-java/creating-pdf-with-java-and-itext-generating-pdf-using-java-example/>. La librería se puede descargar en <https://sourceforge.net/projects/itext/>.
- (0,5 puntos) Identificar en la práctica un ámbito donde aplicar un patrón de los vistos en clase. Como ejemplo, recomendaría utilizarlo en la clase que haga la persistencia de los objetos ya que el acceso es único y centralizado.
- (1 punto) Realizar una ampliación de la práctica para hacer un formulario con AWT o Swing de mantenimiento de asignaturas (alta, baja y modificación)
- (1 punto) Hacer una ampliación adicional a la práctica (acordarlo previamente con el profesor)

5. Otros temas

Cualquier librería externa utilizada se almacenará en una carpeta **libs** en la raíz del proyecto. Se deberá utilizar la versión 1.7 o superior.

Se deberán controlar todos los errores mediante manejadores de excepciones.

Es necesaria la utilización de paquetes para no tener todas las clases juntas y así tener mejor organizado el código.

Se valorará de forma muy importante la limpieza del código, así como que este esté comentado correctamente.

Es obligatoria la realización de la documentación `JavaDoc`.

6. Normas de entrega

La práctica deberá entregarse antes del **16 de enero de 2022 a las 23:59** mediante la opción de entrega Práctica 2 que aparecerá en el moodle de la asignatura.

Se deberá entregar una memoria explicativa (Word, pdf, ...) en la que conste:

- Un diagrama de clases, la descripción de las mismas y la explicación de las funciones más importantes.
- Un breve resumen de como se ha abordado la práctica.
- Los problemas que se han ido encontrando.
- Detalle de lo implementado de la parte II.

Se recuerda que las prácticas son individuales, por tanto, cualquier copia detectada entre alumnos conllevará que ambos alumnos obtendrán la puntuación de 0.

7. Apéndice I: Detalle de tipos de preguntas.

Para ayudar al alumno en las preguntas y que durante la realización de la práctica afiance conocimientos, se ha proporcionado los exámenes de años anteriores. Las preguntas de los mismos deberán estar almacenadas en la práctica dentro de la una carpeta files agrupadas por asignatura. Esta carpeta tendrá todos los ficheros de persistencia de datos (asignaturas con sus preguntas y exámenes creados), de forma que cuando se entregue la práctica estos estén incluidos en la misma.

7.1. Ejemplo de pregunta teórica

Pregunta 8 (0,5 puntos): ¿Qué es el polimorfismo? Contesta y explica dónde lo has utilizado en la práctica de asignatura.

7.2. Ejemplo de pregunta test

Pregunta 4 (0,5 puntos): ¿Qué significa instanciar una clase? Selecciona la respuesta correcta

- a) Crear la definición de la clase.
- b) Crear un objeto a partir de una clase.
- c) Crear clases a partir de otras clases.
- d) Todas las anteriores son correctas.

7.3. Ejemplo de pregunta rellenar

Pregunta 4 (0,5 puntos): Completa la siguiente frase:

Suponiendo que la clase **B** hereda de la clase **A**:

- | | | |
|---------------------------------------|-----------------------|---|
| A la clase A se le llama clase | <input type="radio"/> | . |
| A la clase B se le llama clase | <input type="radio"/> | . |

7.4. Ejemplo de pregunta desarrollo / práctico

Pregunta 15 (3 puntos):

a) Escribe el código en Java de una clase llamada Persona que tenga como atributos dni, nombre y apellidos (todos ellos de tipo string). Incluye los métodos de acceso de lectura y escritura para todos ellos, un constructor que asignará solo dni y otro que asigne todos los atributos.

b) Crea una clase Asignatura que contenga el código de la asignatura, el nombre (tipo String) y la nota de aprobado (tipo int).

c) Crea una clase Alumno que herede de la clase Persona y que agregue un nuevo atributo del tipo array de asignaturas. Añadir un método que permita agregar asignaturas al alumno.