

1. INTRODUCCIÓN

El objetivo de desarrollar un proyecto, cómo práctica de la asignatura, es practicar los conceptos de la Programación Modular y la Orientación a Objetos vistos en clase. Para ello, el profesor proporcionará el enunciado de un proyecto pequeño y un posible diseño mediante diagramas de clase e interacción UML[1], y el alumno implementará el proyecto utilizando la herramienta Greenfoot. La implementación requiere implementar los métodos de las clases/interfaces diseñadas o definir alguna nueva clase/método.

Cualquier información relacionada con la práctica la encontraréis en el Tema 7 (apartado Práctica) de la asignatura “Programación Modular y Orientación a Objetos” de Moodle.

2. DESARROLLO DEL PROYECTO

La realización del proyecto será individual o en parejas, y se llevará a cabo principalmente fuera del horario docente. Aunque también hemos reservado una sesión de laboratorio para la realización del proyecto en horario docente y así, poder aprovechar para resolver las dudas que os surjan.

El proyecto se iniciará el **25 de marzo de 2014** con la publicación del enunciado y finalizará el **2 de mayo de 2014**.

Las tareas a realizar son las siguientes:

1. **Tarea:** Cada grupo debe anotar el nombre del grupo y los nombres de sus miembros en la Wiki Grupos, dentro del apartado ‘Práctica’ en Moodle. Fecha límite: **28 de marzo de 2014**
2. **Tarea:** Antes de la sesión del laboratorio dedicado a la práctica, el alumno habrá tenido que leer el enunciado y analizar el código que se ha proveído.
3. **Tarea:** Desarrollar el proyecto semanalmente (sin dejarlo para la última semana)
4. **Tarea:** Entregar todo el proyecto en el servidor WEB-CAT, antes de la fecha límite.
5. **Tarea:** Entregar un documento con las decisiones tomadas en la implementación de los métodos o clases

3. MATERIAL NECESARIO

Todo el material necesario para la práctica se publicará en el tema "Práctica" del sitio web de la asignatura "Programación Modular y Orientación a Objetos" dentro de la plataforma Moodle. En resumen:

- Los enunciados y el código Java de partida del proyecto.
- Las reglas de estilo en la programación [2]
- Una versión del JDK 6 de Java
- El entorno de desarrollo Greenfoot [3]
- El entorno de desarrollo BlueJ [4], para enviar el proyecto a WEB-CAT

4. ENTREGABLES Y FORMAS DE ENTREGAR

En total se deben de entregar dos entregables:

1. La implementación de la funcionalidad requerida en el proyecto, es decir, todos los ficheros .java que componen el proyecto
2. Un documento (.doc) que detalle los autores y explique las decisiones que se han ido tomando en la implementación de la funcionalidad requerida, además de añadir el código de cada uno.

4.1. FORMAS DE ENTREGAR

El primero de los entregables se enviará al servidor WEB-CAT para validar tanto el estilo de código como la correcta implementación de la funcionalidad. El envío se puede realizar desde BlueJ como directamente desde la web, tal cómo se indica en el apartado 4.2.

El segundo de los entregables se subirá a Moodle, utilizando la correspondiente tarea dentro del tema de la 'Práctica'.

4.2. FORMAS DE ENTREGAR: WEB-CAT VÍA WEB

La dirección web del gestor de tareas WEB-CAT, para enviar la implementación, es la siguiente:
<http://lsi.vc.ehu.es/Web-CAT/WebObjects/Web-CAT.woa>

IMPORTANTE: El código que subáis al gestor de tareas tiene que ser ejecutable (es decir, no debe incluir ningún error de compilación).

Cada uno de vosotros os tenéis que identificar con el mismo *usuario* que tenéis asignado para el correo de la universidad y la *clave* es vuestro número de DNI (por temas de seguridad os aconsejamos cambiarlo). Para los que estáis realizándolo en pareja, el sistema os da la opción de elegir el usuario de vuestra pareja (*Choose a partner*). Por lo tanto, es suficiente que lo envíe uno de ellos.

Cada entregable será un **único** fichero .zip que incluye el directorio del proyecto de *Greenfoot*. Es vuestra responsabilidad subir los entregables de cada tarea (*Assignment*) al apartado correspondiente de dicho servidor dentro de las fechas establecidas. El sistema también os recordará cuáles son dichas fechas en el apartado (*Upcoming Assignment Deadlines*).

Entre las acciones disponibles se encuentran las siguientes:

- *Submit to this assignment*: para subir el entregable de dicha tarea
- *View your most recent results*: para ver los resultados obtenidos
- *Graph your results*: para ver los resultados gráficamente.

Más información sobre el gestor de tareas: <http://web-cat.cs.vt.edu/WCWiki/WebCatCookbook>

Por otro lado, comentar que los entornos de desarrollo Eclipse y/o BlueJ ofrecen la posibilidad de configurar las entregas de las tareas a un servidor (gestor de tareas) y también, integran la vista del resultado de la evaluación o ejecución de las pruebas unitarias en el entorno de desarrollo elegido. La configuración y uso de este tipo de herramientas ya han sido practicadas en los laboratorios.

5. EVALUACIÓN

La nota de la práctica tiene un peso del 25% en la nota final de la asignatura, tal cómo se indico en el programa de la asignatura.

Para el calculo de la nota de la práctica, la evaluación se divide de la siguiente forma:

1. **80%**: Validación de la correcta implementación
2. **10%**: Validación de las reglas de estilo de programación
3. **10%**: Validación de la documentación entregada: decisiones y aspectos de desarrollo del software, tales como, **la modularización**.

6. CONDICIONES

La realización de los laboratorios o proyecto es individual o en pareja. En algún momento, podéis discutir con otros miembros de otros grupos distintos aspectos, pero no se puede copiar el código.

IMPORTANTE: Si se detecta que la solución de los laboratorios o proyecto se ha **copiado** entre varios grupos, se penalizará a **todos** los miembros de dichos grupos con una nota de **0 puntos**.

No se pueden modificar las signatures de los métodos preestablecidos en el código base de cada laboratorio o proyecto. Se pueden definir nuevos métodos si éstos van a ser invocados desde los métodos comentados anteriormente. En este caso, será necesario documentar el nuevo método utilizando sintaxis JavaDoc.

La nota de la práctica se tendrá en cuenta únicamente durante este curso académico.

7. BIBLIOGRAFÍA

1. Resumen de UML, ver 'Material complementario' en Moodle.
2. Nomenclatura en Java, ver 'Material complementario' en Moodle.
3. Greenfoot, <http://www.greenfoot.org>
4. Blue, <http://www.bluej.org/>