

Instituto Tecnológico de Costa Rica Ingeniería en Computación Sede Regional San Carlos	Proyecto Programado Lenguajes de Programación
Prof. Oscar Víquez Acuña.	Paradigma OO

Introducción

La orientación a objetos ayuda a tener un modelo más cercano de la realidad de muchos problemas. Mediante un adecuado soporte de abstracciones de datos es posible tener una visión de objetos que tienen operaciones para manipularlos, sin tener acceso a detalles internos y a la información que encapsulan.

Una forma efectiva para implementar polimorfismo y genericidad de tipos en lenguajes orientados a objetos es mediante la definición de interfaces que son luego implementadas por diferentes clases. De esta manera un programa puede trabajar con objetos del tipo de la interfaz sin saber específicamente a que clase pertenecen.

También, un aspecto interesante de estos lenguajes es la posibilidad de cargar dinámicamente una clase e instanciar objetos de esa clase sin necesidad de conocerla.

Descripción del proyecto

El proyecto consiste en crear en Java o en cualquier otro lenguaje Orientado a Objetos, una plataforma de juegos simples que sea capaz de correr cualquier tipo de juego que cumpla con un patrón mínimo de diseño y de funcionamiento. El sistema tratará a los juegos como módulos (objetos) y será capaz de cargar cualquier módulo de juego mediante la carga dinámica de clases a partir de versiones empaquetadas o ejecutables de los mismos (en el caso de Java, archivos JAR).

El sistema debe cumplir las siguientes restricciones no negociables:

- 1) Desde la pantalla principal, a través de un menú, se podrá cargar un juego/módulo y se podrá acceder a la lista de mejores resultados por cada juego. Esta pantalla principal y cada uno de los módulos debe implementarse utilizando Internal Frames

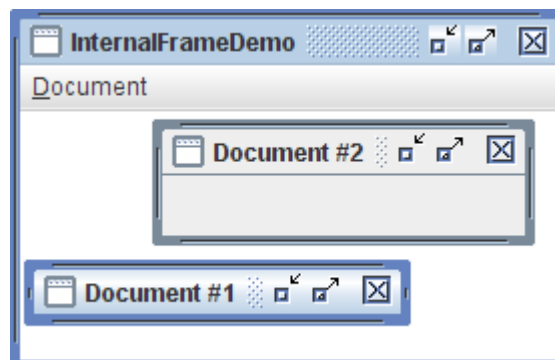


Ilustración 1- Ejemplo de JInternalFrame en Java

- 2) Cada juego debe ser algo simple que permita jugar y que permita mostrar al final del juego alguna estadística relacionada con algo del juego (puede ser el tiempo, el puntaje, cantidades de movimientos, etc) para cada una de las corridas del juego mientras el programa principal se encuentre en ejecución.
- 3) Cada grupo debe al menos implementar un juego y mostrar que es capaz de cargar otros juegos de compañeros a partir de los patrones establecidos
- 4) Cada juego debe implementar la siguiente interfaz que se detalla y que establece los métodos principales de funcionalidad del juego (esta es la clave para la carga dinámica).

```
public interface GameFunction {
    public Stat getStats();
}
```

```

    }

    public class Stat{

        private String clave;
        private String nombre;
        private int valor;

        //constructor

        //setters/getters para cada variable
    }

```

- 5) Se deben respetar los paquetes de implementación de la interfaz y la clase Stat tanto en la consola de juegos como en los juegos como tal.
- 6) El método getStats deberá devolver un objeto Stat con el valor obtenido al jugar el juego donde clave es el nombre del rubro a mostrar (tiempo, puntaje, movimientos, etc), el nombre, el nombre de la persona que lo realizó y el valor, el total obtenido cuando se jugó.
- 7) Cada juego debe estar empaquetado en un JAR de Java. (El JAR debe ser completo e indicar el nombre de la clase principal que debe cargarse)
- 8) No debe ser posible hacer dos instancias diferentes de cada juego. Si ya está cargado, se instancia una vez solamente.
- 9) Las estadísticas de cada juego deben llevarse en la pantalla principal y deben actualizarse cada vez que se mejore el valor de la clave que se tenga como “el más alto”
- 10) El sistema (pantalla principal y módulos) no debe implementar ningún tipo de paquete para sus clases porque de lo contrario alteraría la funcionalidad deseada al momento de cargar clases de forma dinámica al depender de algún directorio específico.

De los Juegos:

Los juegos a implementar deben ser juegos implementados como Internal Frames y deben tener una funcionalidad completa. Deben cumplir con los siguientes requisitos:

- No deben contener ningún modo de funcionamiento asistido por computadora
- Deben indicar por pantalla y no por consola cualquier mensaje que se desee informar al usuario
- Deben indicar cuando el juego se termina o no
- Deben poder reiniciarse cuando el usuario lo determine
- Deben mostrar estadísticas de juego y almacenar de la forma indicada anteriormente el mejor valor de juego establecido en dicha instancia de corrida.

Rubros de calificación:

Se dará énfasis en la calificación a detalles principalmente de funcionalidad, de manera que no será tan importante el código como tal, sino el funcionamiento global del programa. Se tomarán en cuenta lo siguiente:

- Carga dinámica funcional sin que permita cargar nada que no sea implementado por la interface establecida.
- El sistema debe poder cargar los juegos de los demás compañeros del curso.
- Los juegos no deben ser iguales entre compañeros. La originalidad cuenta!!!

Notas Finales para la Tarea:

- El proyecto es para ser realizado **en parejas** y bajo ninguna circunstancia se permitirán copias de trabajos.
- Fecha de entrega: viernes **2 de junio del 2023** antes de las 10:00 pm a través del TECdigital
- La documentación debe seguir el patrón general establecido en los otros proyectos