

RFTP

REQUISITOS

- Elegir tema.

Elegir un tema que se adapte a la funcionalidad y fin de la práctica.

- Crear arrays.

Generar 3 array relacionados en paralelo, con las mismas posiciones para lo que se necesitará tener instalado el programa Eclipse y tener algunos conocimientos básicos.

- Los array deben guardar un máximo de 20 series.

Los arrays deben poder rellenarse con un máximo 20 posiciones, en nuestro caso con series de TV.

- Si el array está lleno se debe informar al usuario.

Si el array se completa con 20 posiciones, la consola debe informar al usuario de ello.

- Si no tiene contenido, no se permitirá modificar o eliminar esa posición.

El programa a su vez, debe informar por consola que no se puede modificar o eliminar una posición inexistente.

- Debe tener un menú que pida al usuario la acción a realizar (añadir, modificar, eliminar, buscar y salir).

El programa debe mostrar un menú con las distintas opciones por consola.

- Análisis Casos de Uso.

Se debe realizar tabla, gráficos y diagrama UML.

- Proyecto subido a GITHUB.

Se tiene que subir el proyecto y todos los avances realizados al programa GITHUB, el cual debemos tener instalado y con una cuenta personal creada.

- Comentar código.

Realizar comentarios sobre el código utilizado en eclipse teniendo en cuenta que no sean

cosas obvias para un programador, y teniendo en cuenta "Diez consejos para mejorar tus comentarios de código fuente" .

- JavaDoc.

Realizar comentario JavaDoc para cada uno de los métodos utilizados.

- Subir formato PDF.

Pasar todo lo que corresponda entregar a formato PDF. Se necesitará Acrobat Reader.

- Realizar un video demostrativo.

A través de PROGRAMA DE VIDEO se debe realizar un video demostrativo de la funcionalidad del programa creado en Eclipse. Se necesitará tener instalado PROGRAMA DE VIDEO.

FUNCIONALIDADES

- Elegir tema.

Elegir un tema relacionado con el ejemplo que se adapte a la actividad.

- Crear arrays.

Realizar arrays 3 arrays paralelos de manera que puedan interactuar con el sistema y relacionado con el tema elegido.

- Los array deben guardar un máximo de 20 series.

Los 3 arrays creados deben tener la capacidad de incluir 20 posiciones cada uno de ellos.

- Si el array está lleno se debe informar al usuario.

Informa al usuario si el array está lleno.

- Si no tiene contenido, no se permitirá modificar o eliminar esa posición.

Informar al usuario de la acción a realizar, borrar y buscar.

- Debe tener un menú que pida al usuario la acción a realizar (añadir, modificar, eliminar, buscar y salir)

Mostrar un menú inicial que permita al usuario seleccionar la acción a realizar.

- Análisis Casos de Uso

Capacidad de representar objetos a través de los diferentes diagramas representados.

- Proyecto subido a GITHUB.

Observar que se ha incluido dentro de la plataforma GITHUB el proyecto realizado.

- Comentar código.

Aclarar a través de comentarios qué código se ha introducido y con qué fin.

- JavaDoc.

Documentar los métodos creados por el programador a fin de que puedan utilizarse y reutilizarse.

- Subir formato PDF.

Facilitar la lectura del trabajo con un programa u formato genérico, en este caso PDF.

- Realizar un video demostrativo.

Se debe observar a través del mismo las pruebas necesarias para comprobar todas las funcionalidades requeridas del programa creado.

TAREA

- Elegir tema.

Para esta práctica y tras observar el ejemplo dado, se toma la decisión de realizarlo con series de TV, actualmente más demandado y con más aceptación entre el público en general.

- Crear arrays.

Se generan 3 arrays relacionados en paralelo, con las mismas posiciones

- Los arrays deben guardar un máximo de 20 series.

Dichos arrays se crean con una capacidad máxima de 20 posiciones.

Si el array está lleno se debe informar al usuario.

Dentro del programa se crea una variable con capacidad de informar por consola al usuario si el array estuviera completo, es decir, sin posiciones disponibles.

- Si no tiene contenido, no se permitirá modificar o eliminar esa posición.

En el apartado de modificar o eliminar, se crea una variable de manera que si el array no tiene contenido, devolver un mensaje por consola de que no es posible modificar o eliminar posición

- Debe tener un menú que pida al usuario la acción a realizar (añadir, modificar, eliminar, buscar y salir)

Crear menú que recoja todas las opciones requeridas.

- Análisis Casos de Uso.

Crear gráficos representativos con DRAWIO y tablas de contenido explicativo de cada uno de los gráficos.

- Proyecto subido a GITHUB.

A través de la mencionada plataforma, hay que subir los ficheros de Eclipse dentro de SRC conforme se van realizando cambios a través de Upload y incorporando los comentarios pertinentes. Así como toda la documentación

- Comentar código.

Comentar las líneas de códigos a fin de aclarar el procedimiento y las variables elegidas si fuera necesario.

- JavaDoc.

Realizar la documentación de los métodos creados y utilizados para el funcionamiento del programa genérico.

- Subir formato PDF.

Pasar toda la documentación a presentar en formato PDF.

- Realizar un video demostrativo.

Crear un video demostrativo con la funcionalidad del programa.

PRUEBAS

- Elegir tema.
- Crear arrays.

Examinamos los array

- Los array deben guardar un máximo de 20 series.

Comprobamos que los array son de 20 posiciones. No obstante y como banco de pruebas se realizar un programa paralelo con 5 posiciones.

- Si el array está lleno se debe informar al usuario.

Introducimos con la opción 1 "añadir" 5 series, 5 plataformas y 5 géneros. Se comprueba que al intentar añadir una posición más de la longitud que le hemos dado a los 3 array, el sistema nos lanza un mensaje por consola indicando que el array está lleno.

Si no tiene contenido, no se permitirá modificar o eliminar esa posición.

Si se selecciona la opción de "modificar" se pide al usuario que introduzca una serie y una nueva entrada, si el array está vacío devolverá un error al usuario por consola al no encontrar coincidencia. Si el usuario selecciona "eliminar" una posición cuando el array esté vacío, lo devolverá al menú inicial.

- Debe tener un menú que pida al usuario la acción a realizar (añadir, modificar, eliminar, buscar y salir).

Comprobamos que todas las acciones recogidas en el menú funcionan correctamente. Accedemos a Menú de pruebas, añadimos 5 series inicialmente y intentamos insertar una más, comprobando que la consola nos devuelve que el array está completo. Eliminamos la primera y tercera serie, modificamos la quinta serie por otra. Buscamos la última serie añadida, y nos indica su posición. Añadimos una serie y comprobamos que se inserta en la primera posición disponible. Para terminar, elegiremos la opción de salir.

- Análisis Casos de Uso.

Comprobamos que todos los casos están redactados y completos.

- Proyecto subido a GITHUB.

Comprobamos que se han añadido el proyecto en main y todas las versiones posteriores en distintas ramas.

- Comentar código.

Comprobamos que el código esta comentado antes de subir la última versión a GitHub.

- JavaDoc.

Comprobamos que Javadoc está comentado antes de subir la última versión a GitHub.

- Subir formato PDF.

Comprobamos que toda la documentación requerida en este formato está completa y en dicho formato.

- Realizar un video demostrativo.

Se realiza un video demostrativo de la funcionalidad del programa a través de programa de pruebas con 5 posiciones para agilizar el video, ya que muchos compañeros han comentado que se alargan en demasía al crear los mismos y sólo tenemos 5 minutos.