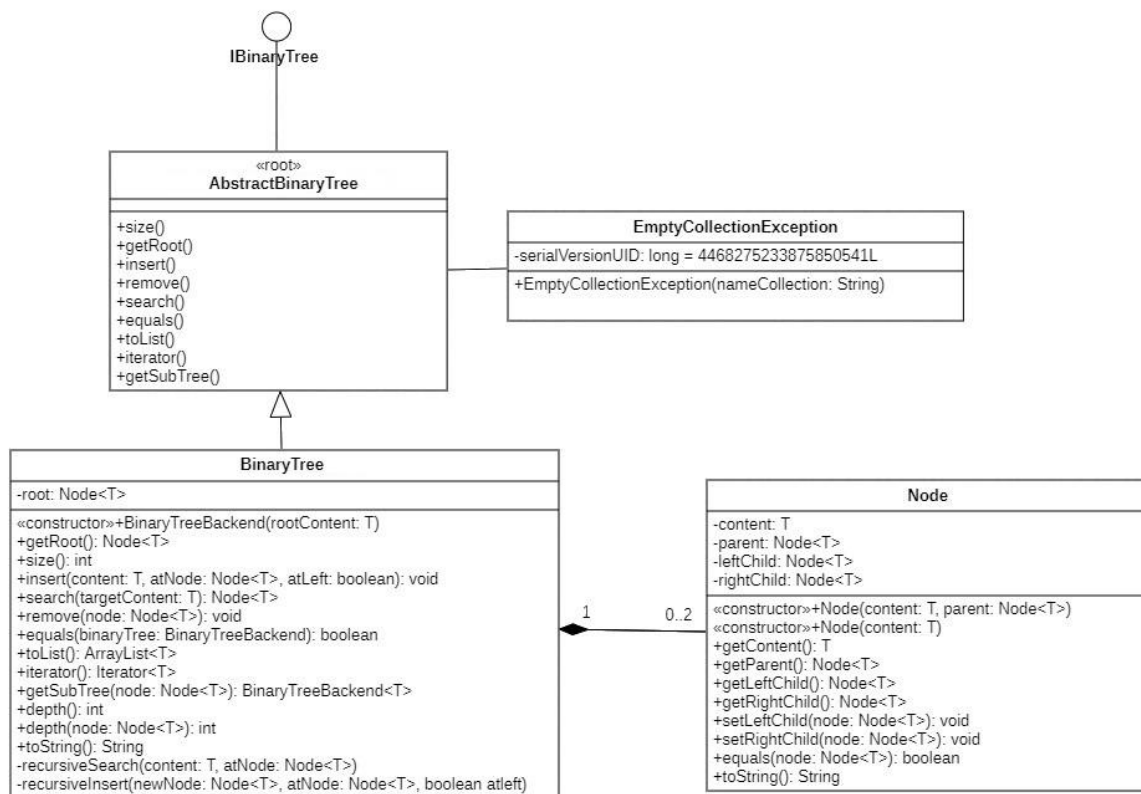


PRÁCTICA 1: PRUEBAS DE CAJA NEGRA

Verificación y Validación 2022-23

Los profesores de la asignatura de Verificación y Validación han implementado un módulo en Java que permite gestionar un árbol binario que puede contener tipos genéricos de datos, desde Integer, String, hasta una lista de clases. Dicho módulo sigue el siguiente modelado de clases:



En la imagen se puede ver que se ha modelado una clase abstracta que implementa una interfaz. La clase implementada (**AbstractBinaryTree**) hereda de la abstracta **BinaryTree**. El objetivo de la práctica es **probar los métodos implementados en la clase BinaryTree** usando Junit5.

Requisitos previos a la realización de la práctica:

- Maven 3.8.6: <https://downloads.apache.org/maven/maven-3/3.8.6/binaries/>
- Java JDK 1.8 o superior
- IDE de desarrollo (Eclipse, Netbeans, IntelliJ, etc.)
- Tener una cuenta en Github (github.com)

Para poder llevar a cabo la práctica debéis configurar vuestro entorno de trabajo de la siguiente manera:

- 1- Un único miembro del equipo debe hacer un “fork” al proyecto base disponible en <https://github.com/VyV-UPM/202223-caja-negra> para crear una copia del mismo en su usuario de Github. Posteriormente debe añadir como colaboradores al resto de compañeros de equipo y al profesor. Se puede hacer el repositorio privado.
- 2- Todos los miembros del equipo se deben descargar de Github el proyecto base forkeado usando Git y abrirlo en el IDE escogido (Eclipse, IntelliJ, etc.). También se puede descargar directamente desde el IDE.
- 3- Añadir las dependencias necesarias para poder usar Junit5 (<https://junit.org/junit5/docs/current/user-guide/>)
- 4- Añadir a las dependencias del proyecto la librería “binarytree.jar” (disponible en Moodle) que contiene las clases implementadas que hemos mencionado anteriormente.
- 5- La documentación proporcionada en “apidocs.rar” corresponde a los métodos de la clase “**AbstractBinaryTree**”, que luego se implementan en la clase **BinaryTree**, por lo que la documentación proporcionada es suficiente para llevar a cabo el ejercicio.

Lo que se pide es realizar pruebas de caja negra en la carpeta “src/test/java” sobre la librería “binaryTree.jar” proporcionada. Toda la información sobre los métodos y lo que hacen está en la documentación Javadoc proporcionada. Lea atentamente la funcionalidad que se proporciona en la documentación, en ella encontrará la información necesaria. Para realizar las pruebas hay que tener en cuenta las siguientes instrucciones:

1. Para homogeneizar las pruebas, vamos a usar elementos de la tabla ASCII. En nuestro caso vamos a utilizar caracteres alfanuméricos. Es decir, consideraremos válido solo las letras (mayúsculas y minúsculas) y números. Cualquier otro símbolo no es válido. Puede ver los elementos de la tabla ASCII en este enlace: <https://elcodigoascii.com.ar/>
2. Utilizando la técnica de clases de equivalencia, identificar los casos de prueba necesarios para probar cada funcionalidad.
3. Se debe probar **TODA** la funcionalidad de la clase **BinaryTreeBackend**, es decir, **TODOS** los métodos. **Justificar la funcionalidad que no cumple requisitos.**
4. Para construir los casos de prueba se utilizará la técnica “**Strong Robust**”.
5. Implementar las pruebas en **JUnit5** para apoyar los casos de prueba descritos, use la funcionalidad de **JUnit5** necesaria: pruebas unitarias, pruebas parametrizadas, creación de suites, etc.
6. Se valorará positivamente la organización de los tests en funcionalidades parecidas. Si por ejemplo tuviéramos un software de clientes, productos y facturas, genere clases con la funcionalidad de los clientes y otro de facturas.
7. Generar un fichero README.MD donde figuren todas las instrucciones Maven necesarias para la ejecución, compilación, empaquetado, etc., del proyecto.

El desarrollo de esta práctica será evaluado mediante preguntas en el **examen final**.