SESIÓN 01

La práctica corresponde al:
Tema 1: Introducción a EDAT en Java (Colecciones Abstractas)

Peso en la nota final:
0%

Objetivos

El objetivo de esta práctica es que el alumno desarrolle habilidades en la implementación de tipos abstractos de datos (TAD) en Java. En esta práctica, se implementará un TAD llamado `EnteroEnRango` que almacena un número entero dentro de un rango específico y realiza operaciones aritméticas que respetan los límites del rango.

Conceptos Teóricos

Los conceptos relacionados con los Tipos Abstractos de Datos (TAD) y cómo se relacionan con los Interfaces en Java se estudian en el Tema 1. El concepto de Iteración sobre una Colección, se explica en la correspondiente sesión de práticas (Sesión 01).

Trabajo del Alumno

1. Programa

El alumno deberá programar una clase en Java que implemente el TAD `EnteroEnRango` debe cumplir con las siguientes especificaciones:

1. **Atributos**:
   - `valor`: Un número entero que representa el valor almacenado.
   - `minimo`: Un número entero que representa el valor mínimo permitido.
   - `maximo`: Un número entero que representa el valor máximo permitido.
2. **Constructores**:
   - Un constructor que inicialice el `valor`, `minimo` y `maximo`. Si el `valor` está fuera del rango `[minimo, maximo]`, debe lanzarse una excepción.
3. **Métodos**:
   - **getValor**: Devuelve el valor almacenado.
   - **getMinimo**: Devuelve el valor mínimo permitido.
   - **getMaximo**: Devuelve el valor máximo permitido.

- **suma**: Suma el valor de otro `EnteroEnRango` y devuelve un nuevo `EnteroEnRango` con el resultado. Si el resultado supera el `maximo`, se ajusta al `maximo`.

  ```
  java public EnteroEnRango suma(EnteroEnRango otro)
  ```

- **resta**: Resta el valor de otro `EnteroEnRango` y devuelve un nuevo `EnteroEnRango` con el resultado. Si el resultado es menor que el `minimo`, se ajusta al `minimo`.

  ```
  java public EnteroEnRango resta(EnteroEnRango otro)
  ```

- **Iteración**: El tipo de datos será iterable. Al solicitar realizar la iteración, devolverá uno a uno los enteros contenidos en el rango que representa, de menor a mayor. El iterador NO podrá modificar el contenido de la Estructura de Datos (no implementará el método `remove()`). Por lo tanto, la clase deberá implementar el interfaz Iterable e incluir las clases internas y métodos a los que ésta obliga.

2. Informe

No se solicita en esta práctica.

Condiciones de Entrega

- La práctica se realizará en grupos de 1 o 2 personas.
- La entrega se hará SOLAMENTE por medio de la plataforma UBUVirtual. Cada miembro del grupo deberá subir su propia copia de la solución. No se admitirán soluciones fuera de plazo o por otros medios de entrega (e-mail, mensajería interna, …).
- La entrega incluirá la información de quienes son los autores de la misma, tanto en los ficheros de código fuente como en los documentos asociados.
- Cada entrega consistirá en un fichero comprimido (formato .zip), con la estructura de nombre
  "`Apellidos1Nombre1_Apellidos2Nombre2`"
- Qué deberá incluir el fichero comprimido:
  - Código fuente de la solución (dentro de la estructura de paquetes necesaria)
  - Documentación en formato Javadoc
  - Informe en formato PDF (en aquellas prácticas que se solicite).
- No hace falta entregar ficheros binarios

Criterios de Evaluación

- **Corrección del funcionamiento**. Se valorará como aspecto más importante que el funcionamiento de la estructura de datos implementada cumpla con todas las condiciones solicitadas, tanto en la documentación oficial como en el enunciado. Se facilitan al alumno un subconjunto de los tests que se emplearán en la corrección, pero se aconseja que el alumno realice de forma adicional los que crea conveniente.
- **Corrección del código**. Ausencia de *warnings* u otros elementos no deseables como variables no utilizadas, código que no se emplea, catch vacíos, ausencia de marcas sobre los métodos (*@Override*, @...) etc.
- **Documentación**. Las explicaciones y razonamientos que se incluyan a modo de comentarios en el código que entrega el alumno. Se solicita que el alumno entregue adicionalmente la documentación generada en formato javadoc, junto con los ficheros de código fuente.
- En este caso, no se evaluará complejidad algorítmica del código.