

## Exámenes

### Tema 1 - S2: Cuestiones sobre el uso de la jerarquía de una EDA vía import

[Volver a la Lista de Exámenes](#)

Parte 1 de 2 - Introducción

0.0/ 0.0 Puntos

En esta actividad tendrás que trabajar con las siguientes clases del proyecto BlueJ *eda*:

- Programa *TestEdaCola* del paquete *tema1* de *ejemplos*
- Clases *ArrayCola* y *Cola* de, respectivamente, los paquetes *lineales* y *modelos* de *estructurasDeDatos*

No hay preguntas

Parte 2 de 2 - Cuestiones

8.04/ 10.0 Puntos

Preguntas 1 de 4

2.0/ 2.5 Puntos

Al compilar en BlueJ el programa *TestEDACola* se van produciendo una serie de errores de compilación, hasta que llegas a un error que no puedes resolver de forma alguna. Indica qué código has tenido que añadir al programa para solventarlos, recordando que solo será válido aquel que pase el Checkstyle.

- Los 2 primeros errores de compilación se arreglan, en este orden, con las instrucciones...

✓ import librerias.estructurasDeDatos.modelos.Cola;

✓ import librerias.estructurasDeDatos.lineales.ArrayCola;

- Una vez arreglados los 2 primeros errores, tras compilar se producen 2 más. El segundo de ellos se debe a un uso inadecuado de la clase *Cola*, que es de tipo genérico *E* (en notación Java *Cola<E>*). Para arreglarlo debes instanciar *E*... ¿A qué tipo Java?

✓ Integer

Congruentemente, para prevenir futuros lanzamientos de *ClassCastException*, el tipo de *ArrayCola* debe ser...

✓ Integer

- ~~Tras arreglar todo lo anterior, el error de compilación que no hay forma de resolver afecta a la variable de nombre...~~

✗ g

**Respuesta correcta:**import librerias.estructurasDeDatos.modelos.Cola;; import librerias.estructurasDeDatos.lineales.ArrayCola;; Integer, Integer, talla

El atributo *talla* ha sido declarado como *protected* en *ArrayCola* y, por tanto, no se puede acceder a él desde *TestEDACola*. El origen del problema y su solución general asociada son...

- ☐ Que el método *talla()* no esté definido en *ArrayCola*; si lo estuviera, el problema desaparecería si, además, en el *main* de *TestEDACola* se escribiera *q.talla()*, en lugar de *q.talla*.
- ☐ Que *talla* se haya declarado *protected* en *ArrayCola*, por lo que con declararlo *public* el problema desaparece.
- ☒ Que *TestEDACola* NI ES una *Cola* NI ES un *ArrayCola*; por lo tanto, aunque el método *talla()* sea un método público de *ArrayCola*, desde el *main* de *TestEDACola* NO se tiene acceso NI a los métodos públicos de *ArrayCola* que no estén definidos en la interfaz *Cola* NI a los atributos *protected* de *ArrayCola*. La única solución es definir una variable *tallaQ* en el *main* de *TestEDACola* e inicializarla y actualizarla convenientemente.

Respuesta correcta:C

Una vez solucionado el problema de compilación de *TestEDACola*, ejecútalo y completa los huecos que aparecen en los siguientes enunciados:

- En el *main* de *TestEDACola* se invoca al método *toString()* definido en *ArrayCola*, pero que no figura en la interfaz *Cola* ¿Por qué esta invocación no provoca ningún error de compilación o ejecución? Porque, al igual que *equals()*, el método *toString()* se ☒ hereda de la clase ☒ Object.
- Accede a *ArrayCola* y comenta su método *toString()*; luego, recompila y ejecuta *TestEDACola*. El resultado de la ejecución del método *toString()* es *q =* ☒ librerias.estructurasDeDatos.lineales.ArrayCola, i.e. el del método *toString()* definido en la clase ☒ Object.

Respuesta correcta:hereda, Object, librerias.estructurasDeDatos.lineales.ArrayCola@\*, Object

Poniendo V o F en los huecos que figuran delante de ellas, indica cuáles de las siguientes afirmaciones sobre la variable *q* de *TestEDACola* son Verdaderas o Falsas:

- ☒ V Los tipos Estático y Dinámico de *q* son, respectivamente, *ArrayCola* y *Cola*.
- ☒ V El tipo Estático de *q* es su tipo de declaración, i.e. *Cola*.
- ☒ V Los tipos Estático y Dinámico de *q* son, respectivamente, *Cola* y *ArrayCola*.
- ☒ V El tipo Dinámico de *q* es el del objeto al que referencia, i.e. *ArrayCola*.
- ☒ V El tipo Dinámico de *q* también podría ser cualquiera compatible con *Cola*.
- ☒ V El tipo Dinámico de *q* también puede ser *Cola*, por lo que la siguiente instrucción no provocaría ningún error de compilación y/o ejecución: *Cola<Integer> q = new Cola<Integer>();*
- ☒ V Si la clase estuviera en el paquete *lineales*, el tipo Dinámico de *q* también podría ser *LEGCola*, por lo que la siguiente instrucción no provocaría ningún error de compilación y/o ejecución: *Cola<Integer> q = new LEGCola<Integer>();*
- ☒ V Si *Cola* fuese una clase abstracta, en vez de una interfaz, la siguiente instrucción no provocaría ningún error de compilación y/o ejecución: *Cola<Integer> q = new Cola<Integer>();*
- ☒ V *q* es una variable Polimórfica a la que, por Enlace Dinámico, se le aplicarán los métodos de la interfaz *Cola* tal y como estén implementados en la clase de su tipo Dinámico (*ArrayCola*, *LEGCola*, etc.).

Respuesta correcta:F, V|T, V|T, V|T, V|T, F, V|T, F, V|T

- [PoliformaT](#)
- [UPV](#)
- [Powered by Sakai](#)
- Copyright 2003-2020 The Sakai Foundation. All rights reserved. Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen.