

1151044 - Programación Orientada a Objetos
Tarea 7 - 2026-I

Se desea definir la siguiente interfaz:

```
interface FlujoEntrada {
    int examina( ) throws IOException;
    void consume( ) throws IOException;
}
```

El comportamiento esperado de los tipos derivados que implementen las funciones es el siguiente:

- Función miembro `int examina()`;
Devuelve un entero correspondiente al valor del byte actual de la entrada, sin consumirlo. Si ésta ya se terminó, debe devolver el valor entero `-1`.
- Función miembro `void consume()`;
Consumo el siguiente byte de la entrada. Si ésta ya se terminó, no tiene efecto.

Con base en esto, implementa dos tipos de dato `EntradaEstandar` y `EntradaCadena` que implementen la interfaz¹, que sean instanciables (es decir, que se puedan declarar objetos de dichos tipos) y que se comporten como sigue.

```
class EntradaEstandar implements FlujoEntrada {
    //...
}
```

- Constructor `EntradaEstandar()`;
Efecto definido por la implementación.
- Función miembro `int examina()`;
Devuelve un entero correspondiente al valor del byte actual de la entrada estándar, sin consumirlo. Si ésta ya se terminó, debe devolver el valor entero `-1`.
- Función miembro `void consume()`;
Consumo el siguiente byte de la entrada estándar. Si ésta ya se terminó, no tiene efecto.

```
class EntradaCadena implements FlujoEntrada {
    //...
}
```

- Constructor `EntradaCadena(String s)`;
Almacena una referencia a la cadena dada. El objeto comienza apuntando al primer byte de la misma.
- Función miembro `int examina()`;
Devuelve un entero correspondiente al valor del byte actual de la cadena. Si ésta ya se terminó, debe devolver el valor entero `-1`.
- Función miembro `void consume()`;
Avanza un byte sobre la cadena. Si ésta ya se terminó, no tiene efecto.

Además, implementa la siguiente clase y función estática² que sea capaz de leer un entero a partir de un `FlujoEntrada`:

```
class RutinasEntrada {
    public static int leeEntero(FlujoEntrada ent) throws Exception {
        //...
    }
}
```

Si el entero no se pudo leer, la función debe elevar una excepción de un tipo compatible con `Exception`. Tu función solamente necesita poder leer enteros no negativos y puedes suponer que no habrá problemas de sobreflujo (*overflow*) y que todos los caracteres de la entrada estándar o de la cadena usara para construir un `EntradaCadena` son ASCII.

Los tipos de dato no debe exponer públicamente ninguna variable miembro interna y deben implementarse de modo de que sea imposible que un objeto de este tipo adquiera un estado inválido. Tu código no debe intentar determinar el tipo real del `FlujoEntrada` pasado a la función `leeEntero`, por lo que está prohibido el uso de `instanceof`, de

¹Recuerden que las funciones de una interfaz son públicas y que los tipos derivados deben respetar este nivel de acceso.

²Si creen que este estilo es raro, noten que `java.lang.math` es una clase que está llena de funciones estáticas. El constructor de la clase es privado, ya que de otro modo podrían intentar ejecutar `new java.lang.Math()` para crear un objeto de tipo “matemáticas”!

`getClass` y cualquier otra forma indirecta de intentar determinarlo. Tu programa no debe declarar variables estáticas. Un programa que lea a lo mucho cien enteros debe terminar en menos de un segundo. Puedes consultar una página de prueba en <https://racc.mx/uam/trimestre-actual/2026-i/poo/tarea7.html>. El código que envíes no debe contener `main`, no debe contener la definición de `FlujoEntrada` y tampoco debe leer ni imprimir nada directamente.

Envía tu código fuente desde tu cuenta institucional al formulario en <https://forms.gle/4XJ4Zje9bzcGdyja9>. Tu código será evaluado con varios casos de prueba y se espera que cumpla la semántica descrita.

Ejemplo de uso	Ejemplo de salida
<pre>void main() throws Exception { var ent1 = new EntradaEstandar(); int a = RutinasEntrada.leeEntero(ent1); int b = RutinasEntrada.leeEntero(ent1); // suponer que el usuario ingresa "5 7" // desde la entrada estándar var ent2 = new EntradaCadena(" 123 456"); int x = RutinasEntrada.leeEntero(ent2); int y = RutinasEntrada.leeEntero(ent2); System.out.printf("%d %d\n", a, b); System.out.printf("%d %d\n", x, y); }</pre>	<pre>5 7 123 456</pre>