Sesión 04: Castillos

Hoja de problemas

Programación 2

Ángel Herranz
aherranz@fi.upm.es
Universidad Politécnica de Madrid

2019-2020

Ejercicio 1. Esta hoja de ejercicios va a girar alrededor de los números racionales. Los números racionales son aquellos números que pueden expresarse como el cociente de dos números enteros:

$$\frac{p}{q}$$

Un **numerador** p y un **denominador** q **distinto de 0**. ¿Recuerdas estas reglas?

$$\frac{a}{b} + \frac{c}{d} = \frac{a \times d + c \times b}{b \times d} \tag{1}$$

$$\frac{a}{b} - \frac{c}{d} = \frac{a \times d - c \times b}{b \times d} \tag{2}$$

$$\frac{a}{b} \times \frac{c}{d} = \frac{a \times c}{b \times d} \tag{3}$$

$$\frac{a}{b} : \frac{c}{d} = \frac{a \times d}{b \times c} \tag{4}$$

Ejercicio 2. Queremos acabar implementando una clase capaz de representar y operar con números racionales para resolver *castillos* como este:

$$\frac{(\frac{3}{5} - \frac{1}{4} + \frac{1}{10}) \times \frac{3}{2} - \frac{1}{5}}{(\frac{2}{6} + \frac{1}{3} - \frac{6}{4}) : \frac{2}{3} + \frac{1}{6}}$$

Vamos a suponer que alguien ya ha implementado la clase `Racional` y que han implementado ciertos comportamientos de tal forma que podemos escribir el siguiente código que representa simplemente el racional $\frac{1}{4}$:

```
public class Castillos {
  public static void main(String[] args) {
    Racional r;
    r = new Racional(1,4);
    System.out.println(r);

  }
}
```

El resultado que esperamos ver en pantalla es:

```
1 / 4
```

Un poco más elaborado, un código que resolverá la operación

$$\frac{2}{6} + \frac{1}{3}$$

```
public class Castillos {
  public static void main(String[] args) {
    Racional r;
    r = new Racional(1,4);
    System.out.println(r);

    Racional mini;
    mini = new Racional(2,6);
    mini.sum(new Racional(1,3));
    System.out.println(mini);

  }
}
```

Para ver en pantalla

```
1 / 4
12 / 18
```

Ejercicio 3. Tienes que programar la clase `Racional`. Recuerda lo que hemos aprendido en las clases anteriores. Parece que los objetos de tipo `Racional` tienen que tener dos enteros: numerador y denominador. Empieza por ahí.

Ejercicio 4. También parece que queremos ser capaces de crear objetos usando la expresión **`new`** `Racional(`p`, `q`)`.

Ejercicio 5. Llegó el momento de implementar la operación de suma: `sum`. Como habrás visto en el código, la forma en la que funcionan las sumas es la siguiente: si se tienen dos objetos de tipo `Racional`, digamos que r_1 y r_2, se puede ejecutar la siguiente operación:

$$r_1\texttt{.sum(}r_2\texttt{);}$$

El efecto tiene que ser que el objeto r_1 representa el racional resultado de sumar los racionales representados por r_1 y r_2 **antes** de ejecutar la operación (y r_2 no cambia).

Ejercicio 6. Antes de continuar con el resto de las operaciones, vamos a implementar una operación que nos permita ver una representación de un objeto del tipo `Racional`. Para ello vamos a enriquecer la clase con un tercer comportamiento que va a devolver un `String`. El comportamiento lo vamos a llamar `toString` tal y como vimos con las canciones.

Ejercicio 7. Implementar el resto de las operaciones.

Ejercicio 8. Probar la implementación con este código:

```
public class Castillos {
  public static void main(String[] args) {

    Racional castillo;
    castillo = new Racional(3,5);
    castillo.res(new Racional(1,4));
    castillo.sum(new Racional(1,10));
    castillo.mul(new Racional(3,2));
    castillo.res(new Racional(1,5));
    Racional divisor;
    divisor = new Racional(2,6);
    divisor.sum(new Racional(1,3));
    divisor.res(new Racional(6,4));
    divisor.div(new Racional(2,3));
    divisor.sum(new Racional(1,6));
    castillo.div(divisor);
    System.out.println(castillo);
  }
}
```

Dicho código representa el *castillo*

$$\frac{(\frac{3}{5} - \frac{1}{4} + \frac{1}{10}) \times \frac{3}{2} - \frac{1}{5}}{(\frac{2}{6} + \frac{1}{3} - \frac{6}{4}) : \frac{2}{3} + \frac{1}{6}}$$

Ejercicio 9. Vamos a enriquecer la clase con el comportamiento de crear un número racional que sea un entero. Por ejemplo, el número -42 se puede representar como una fración

$$\frac{-42}{1}$$

Queremos poder hacer **`new`** `Racional(-42)`.

Ejercicio 10. Seguro que a estas alturas tienes un montón de dudas y mejoras que realizar. ¿Quizás cosas como estas?

- ¿Cómo voy a representar el 0?
- ¿Puede pasar que el denominador acabe siendo 0? ¿De qué forma? ¿Qué deberíamos hacer?
- ¿No deberíamos simplificar? ¿Cómo?

Ejercicio 11. Añade a tu clase `Racional` el comportamiento de simplificación.

Ejercicio 12. Modifica todas las operaciones para que dejen el número simplificado.

Ejercicio 13. A veces la misma operación se aplica a un montón de racionales, un ejemplo podría ser

$$\frac{2}{6} + \frac{1}{3} + \frac{6}{4}$$

También a veces, tenemos operaciones entre racionales y enteros (que al fin y al cabo son racionales), un ejemplo podría ser

$$5 + \frac{1}{3}$$

- ¿Podemos hacer versiones de los comportamientos con otro parámetro para poder sumar o multiplicar tres racionales?
- ¿Podemos hacer versiones de los comportamientos que admitan un entero en vez de un racional?