

Auxiliar 1 - Java, Clases y Strings

Profesores: Nelson Baloian
Jeremy Barbay
Patricio Poblete

Auxiliares: Gabriel Flores, Sven Reisenegger
Elizabeth Labrada, Gabriel Norambuena
Cristóbal Muñoz

P1. Instalar Java

- Descargar e instalar el jdk (java development kit) desde acá: www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- Agregar la variable de entorno `JAVA_HOME=carpeta\del\jdk` (sin backslash al final).
- Editar la variable de entorno `PATH` y agregarle `%JAVA_HOME%\bin`.
- Comprobar que haya quedado todo bien abriendo una consola de comandos y ejecutando:

```
java -version
javac -version
```
- (OPCIONAL) Instalar el IDE eclipse desde <https://www.eclipse.org/downloads/>. Esto facilita mucho la programación con auto-completado y reconocimiento de errores y warnings, además de entregar herramientas para compilar y ejecutar programas (entre muchas otras cosas como ponerle colores a las palabras).
 - (a) Al comenzar la instalación seleccionar IDE for Java Developers.
 - (b) Elegir ubicación de instalación.
 - (c) Al iniciar, elegir el workspace (este es el lugar en el que se guardan todo los proyectos, el default basta).
 - (d) Saltarse el mensaje inicial e ir a la pantalla normal.
 - (e) File → New → Java Project.
 - (f) File → New → Class.
 - (g) Dentro de la clase, escribir `main` y luego presionar `ctrl+espacio` (auto-completar) → `main method`.
 - (h) Dentro del main, escribir `syso`, auto completar `"sysout - print to standard out"`. Rellenar con el argumento `"Hello World!"`.
 - (i) Guardar y presionar el icono de Play, debería aparecer en la consola (parte inferior) `Hello World!`
- (OPCIONAL 2) Existen otros IDEs como IntelliJ IDEA, si desean ocupar otro no hay problema.

P2. Clases de Java

Cree la clase `Fraccion` con la siguiente definición:

- (a) Cree un constructor `public Fraccion(int num, int den)`, que inicializa una fracción usando `num` y `den` como numerador y denominador.
- (b) Cree otro constructor¹ `public Fraccion(String frac)` donde `frac` es un string que representa una fracción por ejemplo `"3/5"`, debe descomponer este string y guardar los números como enteros. (Puede suponer que el string siempre será válido).
- (c) Cree la función `public static int mcd(int a, int b)` que calcula el mcd entre 2 números, para esto use el algoritmo de Euclides este dice que el $mcd(a, b) = mcd(b, a \% b)$ además como caso base se tiene que $mcd(a, 0) = a$.
- (d) Cree el método `public void simplificar()` que simplifica la fracción, para esto use `mcd`.
- (e) Cree el método `public Fraccion suma(Fraccion other)` que devuelve la fracción simplificada que representa la suma de la fracción actual con la fracción `other`.
- (f) Cree el método `public String toString()` que devuelve un string que representa la fracción por ejemplo `"4/7"`.

Utilice la clase anterior en un programa interactivo que sume `n` fracciones. Primero deberá pedir al usuario un numero `n`, y luego deberá pedir que ingrese `n` fracciones, al final deberá mostrar la suma de las `n` Fracciones.

```
n?  
3  
Fraccion 1?  
1/4  
Fraccion 2?  
1/3  
Fraccion 3?  
-1/12  
La suma total es: 1/2
```

P3. Strings

- (a) **Rotación Circular** Un string `r` es la *rotación circular* de otro string `w` si se remueve un carácter del final de `w` y se coloca al principio de éste sucesivamente. Por ejemplo `ritmoalgo` es rotación circular de `algoritmo` y viceversa. Escriba la función en Java `static boolean esRotacionCircular(String r, String w)` que recibe dos strings y retorna `true` si uno es rotación circular del otro (`false` en caso contrario).
- (b) **Palíndromo** Escriba una función que diga si el String `s` es palíndrome o no.

¹Las clases en Java pueden tener más de un constructor siempre que reciban parámetros distintos.

- (c) **Oración palíndrome** Escriba una función que diga si una oración es palíndrome (es decir, el String contiene espacios). Por ejemplo *Anita lava la tina*, *Ana lleva al oso la avellana*.
- (d) **String Completo** Un String s se dice completo, si éste contiene todas las letras del alfabeto inglés (es decir de la 'a' a la 'z' sin la 'ñ'). Escriba una función que reciba un String y retorne **True** si es completo y **False** en caso contrario.