Universidad de La Laguna

ESIT – INFORMÁTICA
Departamento de Ingeniería Informática y de Sistemas

Computabilidad y Algoritmia
Curso 2019-2020

PRACTICA 1: La clase Racional

1.1. Objetivos

Se propone que el alumnado utilice este ejercicio para poner en práctica al menos los siguientes aspectos del desarrollo de programas en C++, algunos de los cuales ya debiera conocer a partir de su experiencia en asignaturas de primer curso de la titulación:

- Paradigma de programación orientada a objetos.
- Formato propuesto para la escritura de programas en C++ en esta asignatura.
- Lectura y escritura de ficheros de texto en C++.
- Compilación de programas utilizando `make` [3].
- Edición de ficheros utilizando el editor `vim` [2].

En la sesión de corrección de esta práctica se requerirá conocimientos básicos sobre `vim` y la compilación del programa realizado se hará usando `make`.

Si el alumnado tiene dudas respecto a cualquiera de estos aspectos, debiera acudir al foro de discusiones de la asignatura para plantear allí dichas dudas.

Se espera asimismo que a través de ese foro el alumnado intercambie experiencias y conocimientos, ayudándose mutuamente a resolver dichas dudas.

También el profesorado de la asignatura intervendrá en las discusiones que pudieran suscitarse.

1.2. Introducción

Un número racional [1] tiene un numerador y un denominador de la forma `a/b`, donde `a` es el numerador y `b` el denominador. Por ejemplo, 1/3, 3/4 y 10/4 son números racionales.

Un racional no puede tener denominador 0, pero sí puede ser cero el numerador. Todo número entero `n` es equivalente al racional `n/1`. Los números racionales se utilizan en cálculos precisos que involucran fracciones. Por ejemplo, `1/3 = 0.33333 ...` Este número no puede ser representado de forma precisa en formato de punto flotante

utilizando los tipos float o double. Para obtener resultados precisos es conveniente usar números racionales.

C++ dispone de tipos de datos para enteros y números en punto flotante, pero no para racionales. En esta práctica se propone el diseño de una clase para representar números racionales.

1.3. Ejercicio práctico

Desarrollar un programa cliente `Racionales.cpp` que permita operar con números racionales y haga uso de la clase `Racional` que ha de diseñarse.

Las siguientes deben tomarse como especificaciones del programa a desarrollar

- El programa ha de implementar la clase `Racional` que incluirá al menos métodos para:
  - Crear objetos de tipo `Racional`. En este sentido, se debe implementar un constructor por defecto y uno parametrizado.
  - Leer (por teclado o desde fichero) un objeto de tipo `Racional`.
  - Escribir (a fichero o a pantalla) un objeto de tipo `Racional`.
  - Sumar dos objetos de tipo `Racional`.
  - Restar dos objetos de tipo `Racional`.
  - Multiplicar dos objetos de tipo `Racional`.
  - Dividir dos objetos de tipo `Racional`.
  - Comparar objetos de tipo `Racional`.
- Por otro lado, el programa ha de permitir leer un fichero de texto en el que figuran pares de números racionales separados por espacios de la forma:

  ```
  a/b c/d
  e/f g/h
  ```

  y para cada par de pares, el programa ha de imprimir en otro fichero de salida todas las operaciones posibles con cada uno de los pares de números del fichero de entrada, de la forma:

  ```
  a/b + c/d = n/m
  ```

  ...
- Al ejecutarse en línea de comandos sin parámetros: `./Racionales` el programa mostrará un texto de ayuda indicando la forma de ejecución del programa en la que el usuario puede pasar como parámetros los nombres de los ficheros de entrada y salida que utiliza el programa.

- Todo el código fuente ha de estar escrito de acuerdo a la guía de estilo de Google para C++ [4]. Dentro de esa guía prestaremos particular atención en este ejercicio a los siguientes aspectos:

  - Formateo del código (apartado *Formatting* en [4])
  - Comentarios de código de diverso tipo (apartado *Comments* en [4])
  - Nominación de identificadores, clases, ficheros, etc. (apartado *Naming* en [4])

  Pueden asimismo utilizarse las directrices que se presentan en la *C++ Core Guidelines* [5] en cuanto al estilo de escritura del código.

Bibliografía

[1] Rational Number, https://en.wikipedia.org/wiki/Rational_number

[2] Vim isn't that scary. Here are 5 free resources you can use to learn it, https://tinyurl.com/vim-not-scary

[3] GNU Make, https://www.gnu.org/software/make/manual/make.html

[4] Google C++ Style Guide, https://google.github.io/styleguide/cppguide.html

[5] C++ Core Guidelines, https://github.com/isocpp/CppCoreGuidelines/blob/master/CppCoreGuidelines.md