Corso di Laboratorio di Programmazione
A.A. 2025/26

Laboratorio 2
Classi e overloading degli operatori

Nota: i quesiti e gli esercizi seguenti sono tratti (ma non tradotti) dal libro di testo.

Discussione

A coppie, rispondete alle seguenti domande (Review, cap. 9, p. 338 sgg.):

1. What are the two parts of a class?
2. What is the difference between the interface and the implementation in a class?
3. Why is a constructor used for the `Date` type instead of an `init_day()` function?
4. What is an invariant? Give examples.
5. When should functions be put in the class definition, and when should they be defined outside the class? Why?

Esercizio (#13, pp. 340-341)

Design and implement a rational number class, `Rational`. A rational number has two parts: a numerator and a denominator, for example 5/6 (five-sixths, also known as approximately 0.83333). Look up the definition if you need to.
Provide:

A. a default constructor initializing the `Rational` number to 0,
B. a constructor accepting an `int` argument (representing an integer value that should be represented in `Rational` form),
C. a constructor accepting two `int`s for numerator and denominator,
D. assignment,
E. addition,
F. subtraction,
G. multiplication,
H. division,
I. equality (==), greater than (>) and less than (<), and output to stream (<<) operators. Also, provide a conversion to `double` via a `to_double()` function.

Test all the overloaded operations in the main function.

Regarding the addition operation, try overloading:

A. the sum of two `Rational`s, and
B. the sum of a `Rational` and an `int`

Both overloads shall be implemented in two ways: using a member function and a helper function (they should not be implemented at the same time!). What are the differences if you sum `int + Rational` vs `Rational + int`?

You should pay strong attention to the type returned, and whether it should be a variable or a reference.