Corso di Laboratorio di Programmazione

Esercitazione 5 – Template, ereditarietà
18/12/2019

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. 19&14):

1. Why don't we just always define a vector with a large enough size for all eventualities?
2. Which two operations define copy for vector?
3. When must we copy vector elements to a new location?
4. What is a template?
5. What is generic programming?
6. How can you make a class abstract?
7. What is a virtual function and how does it differ from a non-virtual function?
8. What is a base class?
9. What makes a class derived?
10. What is the difference between a protected member and a private one?
11. How does a pure virtual function differ from other virtual functions?
12. What does overriding mean?

Esercizi (da svolgere usando l'IDE)

1. Write a template function f() that adds the elements of one vector<T> to the elements of another; for example, f(v1, v2) should do v1[i] += v2[i] for each element of v1.
2. Define a class Int having a single member of class int. Define constructors, assignment and operators +, -, *, / for it. Test it, and improve its design as needed (e.g., define operators << and >> for convenient I/O).
3. Define a class Controller with four virtual functions on(), off(), set_level(int) and show(). Derive a class from Controller: it should be a simple test class where show() prints out whether the class is set to on or off and what is the current level.
4. Build on the Shape class discussed during the last lecture. Extend the interface of the Shape class to include area calculation. Extend the class hierarchy to include the following shapes:
    1. rectangle
    2. square
    3. circle
    4. rhombus
    5. parallelogram