

Abstract classes. Object class.

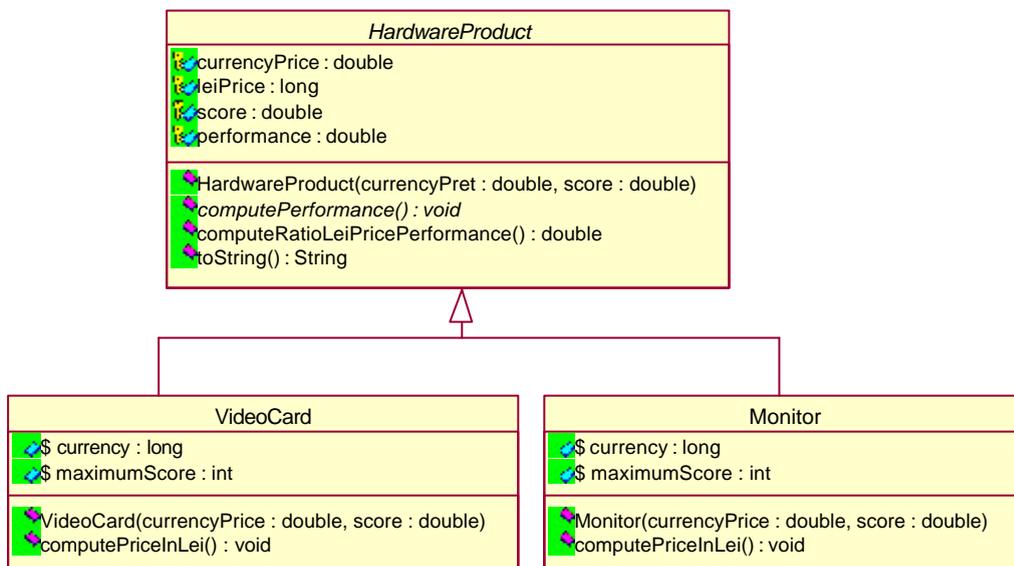
The owner of a hardware products shop has to do a statistic concerning the ration between the price (in lei) of two kinds of its products (video cards and monitors) and the performance of that product. In the case of video cards, the performance criterion is given by the obtained point number in 3DMark. The monitors are evaluated depending by the obtained refresh to a 1152x864 resolution.

Depending on this score, the performance (memorized as a real number) is computed after the following formula:

$$\text{performance} = \text{obtained Score} / \text{maximumScore} * 100,$$

where maximumScore represents the maximum score which can be obtained by all the products from the same category. We assume that the maximum score for the video cards is equal with 100, and, for monitors, is 50.

Use the following UML class diagram associated to application:



HardwareProduct is an abstract class, having the abstract method computePerformance(). In addition, this class contains the protected variables currencyPrice, score, leiPrice and performance. The value of the leiPrice variable will be computed, depending by the currency price and type which is used. We assume that the video cards' price is given in €, and for monitors in \$.

Requirements:

- Implement the class from the above diagram.
- In order to test the classes, implement the class TestShop with a method main that carries out the following tasks:
 - Create an array of ProdusHardware[] products, using the data from the files:
PlaciVideo.txt Monitoare.txt
130 76 249 78
108 77.79 224 95
152 72.55 249 96
53 54.69 279 107
 - Compute the performance for each product from array.
 - Print the relevant information fro each object from array.