

Design a simple `MobilePhone` class. A mobile phone can make calls and can send texts, both at a fixed cost. This cost should be set or fixed on construction to default values of €0.20 per call and €0.10 per text or, to some other values determined by the user. All phones have an initial balance of €0.00 or can have a balance specified on construction. The phone balance can be topped up and it should be possible to see the balance on the phone in euro. The user may also need to see the cost per text and cost per call. It should *not* be possible to change the price of calls or texts after the phone object has been created. It should *not* be possible to make calls or send texts if there is insufficient balance.

How to approach this question: *(These are guidelines only)*

- What methods would you supply for a `MobilePhone` class? Describe them informally.
 - Translate these informal descriptions into Java method
 - Think of the method signature in terms of the name, parameter lists, and return types of the each methods or operation.
 - What instance variables would you supply?
 - Think about constructors, how can a mobile phone object be constructed?
 - Consider the state of the instance variables when the object is created.
 - Consider what happens when a user makes a call, or send text. What, if anything, will happen to the instance variables?
1. Draw the UML diagram.
 2. From the UML, write the method signatures in JGrasp.
 3. Implement all of the methods of your class (i.e. write the method body).
 4. Now test your class by writing a basic tester program `MobilePhoneTester`.
 5. Using the `MobilePhoneTester` class defined write a program that will allow the user to select the following options from a menu until the user choses to quit (enters 0). *The user should not be able to make a call or send a text if there is insufficient balance on the phone.*

Mobile Phone Menu

1. Make a call
2. Send a text
3. Top up
4. Check balance
5. Check cost of text
6. Check cost of call
0. Quit

Please select your option: