Homework #2

This work is based on material from Chapter 2 of *Objects First with Java - A Practical Introduction using BlueJ* by David Barnes and Michael Kolling. The goal is to look inside the structure of a Java Class.

Some useful BlueJ tips

- In the editor, use Edit/Auto-layout after making any changes to your code, to correct indentation. Using auto-layout makes it easier to spot bugs.
- Holding down the ctrl key and hitting the space key should allow you to see code completions.
- You can *comment out* code in BlueJ by selecting one or more lines of code and selecting Edit/Comment from the menu.
- You can compile your code with a single shortcut key - ctrl+k on Windows or Linux, cmd+k on MacOS.

Task 1

Download the HW2_code.zip zip file, and the file on your computer. Open it in BlueJ, and compile the code. (See last week's labs for more detailed instructions if you don't recall how to do this.)
Create a `TicketMachine` instance. You will be asked to supply a number that corresponds to the price of tickets (assumed to be in cents) that will be issued by that particular machine. For instance, entering `500` would create a machine which issues tickets costing $5. Take a look at the machine's methods. You should see the following:

`getPrice`, `getBalance`, `insertMoney`, and `printTicket`.

Try the `getPrice` method – what does it return? Use the `insertMoney` method to simulate inserting an amount of money into the machine, and use `getBalance` to check that the machine has kept an accurate record of the amount just inserted. Try inserting several amounts and check what result `getBalance` gives you. Once you have inserted enough money for a ticket, call the `printTicket` method. A facsimile ticket should be printed in the "Terminal" window.

1. What value is returned if you get the machine's balance after it has printed a ticket?

2. Experiment with inserting different amounts of money before printing tickets. Do you notice anything strange about the machine's behavior? What happens if you insert too much money into the machine – do you receive any refund? What happens if you do not insert enough and then try to print a ticket?

3. Create another ticket machine for tickets of a different price; remember that you have to supply this value when you create the machine object. Buy a ticket from that machine. Compare the printed ticket with that from the first machine you created – does it look any different?

Task 2

Take a look at the source code for the `TicketMachine` class. You can do this by double-clicking the class in the main BlueJ window (or by right-clicking it and selecting "`Open Editor`"); a new window should open up, containing the source code for the class.

Task 3

1. Create an accessor method `getTotal` in the `TicketMachine` class; the new method should return the value of the `total` field.

   Start by pasting the following code into the class:

   ```
   /**
    * Return the total amount of money collected by this machine.
    */
   public int getTotal()
   {
       return 0; // replace this with your own code
   }
   ```

   Then replace the line containing `return 0` with your own code. Compile the code; then try creating a `TicketMachine` instance and using it to see if your new method does what you expect.

2. Create a mutator method `setPrice` in the `TicketMachine` class. The new method should take one parameter, and set the `price` field to the value of that parameter.
   Start by pasting the following code into the class:

   ```
   /**
    * Set the price of this machine's tickets to be cost (if reasonable)
    */
   public void setPrice(int cost)
   {
       //add code here
   }
   ```

   Then replace the line that says `//add code here` with your own code. Again, make sure your code compiles, and try making a `TicketMachine` instance to ensure it does what you expect.

Task 4

Try the following, if you have completed the previous tasks.

1. Modify the code of `insertMoney` so that a user can enter only “sensible” amounts of money. What do you think “sensible” should mean? (Try calling `insertMoney` with different values – are there any you think should not be allowed?)

2. Modify the code of `printTicket` so that it will:

   a) Print a ticket only if enough money has been input.

   b) Print the error message "Not enough money to buy a ticket." if there is not enough money.

   c) Subtract only the ticket price from the balance (possibly leaving change to be given) rather than setting the balance to 0.

In both cases, try creating a new `TicketMachine` instance, and checking to see if your code does what you think it does.

Submission

Submit your code, with the revised `getTotal, setPrice, insertMoney, printTicket` methods, as follows. Go to CTL, open the menu Student Activity -> Assignment, click on the button for submitting the report. Attach and submit your code file **`TicketMachine.java`**. Do not submit any other type of file: zip, class, txt, etc. This is the method that will be used to submit your project work for the class.