

6. Windows Forms Applications

Exercise 06.1 Examine the supplied Console Application

1. You have been supplied with a new version of the console application that you have been writing as an exercise project. This console application has been modified in the following ways:
 - 1.1. The classes Can, CanRack, Coin, CoinBox, PurchasePrice, and VENDBADFLAVORException have been made **public**. This is so they can be used outside of the console application project (this, and existence of a Main() method, is the biggest difference between a class library and a console application).
 - 1.2. CanRack has had an indexer added to it.
 - 1.3. Coin now has several coins added (one for each Denomination), some conversion routines, a static constructor, and a static property that will return a set of all the coin denominations.
 - 1.4. CoinBox now includes methods Withdraw, ContainsExactChangeFor, RemoveCoinDenomination, CanMakeChange, Transfer, and a static constructor.
 - 1.5. The added code mostly does what you might expect (optionally, you can write these routines yourself, but they have been provided to you in the interest of time constraints). If you are unsure what the member does based on its name, read the code.
 - 1.6. If you want to continue to use as much of your code as possible, just cut-and-paste the new methods as needed. You may find that you don't actually need all of the utility provided by the new code.

Exercise 06.2 Create a new Windows Forms Application Vending Machine

1. Create a new Windows Forms Application to give the vending machine a GUI interface.
2. Add a reference in your Windows Forms Application project to the supplied console application with the new operations.
3. On the new form, place the following controls:
 - 3.1. A label to use as a title telling the user to insert money for a can of soda
 - 3.2. A label for stating that exact change is required
 - 3.3. A label showing the total amount of money that has been inserted into the machine
 - 3.4. Three PictureBoxes, one for each flavor of soda
 - 3.4.1. Use Paint to create a simple image file for each flavor
 - 3.5. A button for returning money to the user (i.e., to act as the Coin Return lever)
 - 3.6. Three buttons, each one ejecting a can of the appropriate soda flavor
 - 3.7. Four buttons, one for each denomination of coin (we don't need a button for slugs)
4. The vending machine should work the way vending machines normally work:
 - 4.1. Users should be able to put money into the machine
 - 4.2. User should be able to get their money back without buying a soda

- 4.3. When they have put enough in, users should be able to eject a can of soda of a given flavor, if the machine still has cans of that flavor left
- 4.4. They should be able to get change back if they put in too much
- 4.5. Users should not be able to eject a can of a flavor that is out of stock
- 4.6. There should be an indication if exact change is required (in which case, users would not get change back if they eject a can of soda)
5. Add event procedures to the various buttons to implement the vending machine operation
6. Don't forget to name your control objects prior to giving them event procedures
7. Use a MessageBox object to show a dialog box that says the user got a can of soda and to show that the user is getting their money back or their change back

Suggestions

8. Create two CoinBoxes:
 - 8.1. A main box to hold most of the money, and a temporary box to hold the money that has been inserted as a new user is in the process of buying a soda
 - 8.2. Once a soda can has been paid for, transfer the money from the temporary box to the main box
9. Place a small amount of money in the main box during vending machine initialization
10. Remember that the purpose of the exercise is to create a Windows Forms Application that includes several controls with associated event procedures. Although preferable, it is not necessary to implement all of the vending machine operations