



Vending Machine

C# fundamentals

Vending Machine

Your Workshop task for this week is to make a vending machine. It should be able to take requests from the user as to what product it is supposed to return. It should also take payment and send the product back to the user with an appropriate amount of change.

Required Features:

- Money should be input in fixed denominations:
 - 1kr, 5kr, 10kr, 20kr, 50kr, 100kr, 500kr and 1000kr.
- The user should be able to put any amount of money in, adding to the “money pool”.
 - The money inserted must be of a valid denomination.
- The user should be able to buy any number of products, if they have the money for it in the machine.
- When the user decides to stop buying things, the remaining money should be returned as change.
- The vending machine should be able to have different types of products.
- Once a product has been purchased, the user should be able to use it, showing a message about how it is used (drink the drink, eat the snack etc.)

Code Requirements:

- Money denominations should be defined as an array of integers (readonly).
- abstract Product class
 - the base class type the vending machine itself should use.
 - abstract Examine, to show the product’s information including the unique data.
 - abstract Use, to put the product to use once it has been purchased (output a string message how to use the product)
- The VendingMachineService class shall implement the following interface IVending:
 - Purchase, to buy a product, get product if able to buy otherwise return null or throw exception.
 - ShowAll, show all products, get list of string’s with Id/Name/Cost of product.
 - Details, show detailed information of a selected product, get a string with information about the product including its unique data.
 - InsertMoney, add money to the pool.
 - EndTransaction, returns money left in appropriate amount of change (Dictionary<int, int>).
- Minimum of 3 product types and each should be its own class and be child class of abstract Product class.
 - Must have at least one unique field/property more than its parent class.
- Unit test your code.



Vending Machine

C# fundamentals

Resources:

- <https://msdn.microsoft.com/en-us/library/ms173152.aspx> MSDN - Polymorphism
- <https://msdn.microsoft.com/en-us/library/ms173149.aspx> MSDN - Inheritance

Subjects Covered:

- Inheritance
- Polymorphism
- Interfaces
- Abstract Classes

Good luck!