

Tutorial 3

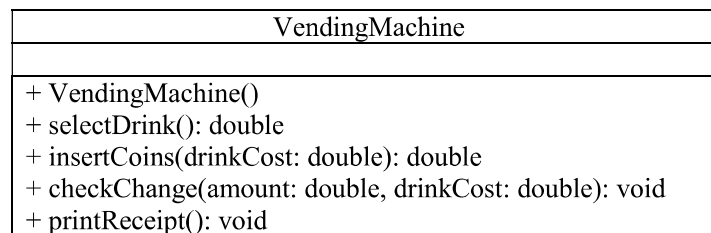
Class Methods & Inheritance

1. Design a program to implement a vending machine for buying drinks. Write a class `VendingMachine` that has the following methods:

```
public class VendingMachine
{
    // constructor
    public VendingMachine() {}

    // get the drink selection, and return the cost of the drink
    public double selectDrink() {...}
    // insert the coins and returns the amount inserted
    public double insertCoins(double drinkCost) {...}
    // check the change and print the change on screen
    public void checkChange(double amount, double drinkCost) {...}
    // print the receipt and collect the drink
    public void printReceipt(){...}
}
```

The UML class diagram for the `VendingMachine` class is given below:



Write an application class `VendingMachineApp` to test the class `VendingMachine`. The program allows users to select the drink to buy, and accept coins inserted by the user to pay for the drink. The program will also print the receipt for user to collect the drink.

- a) Discuss the design of the `VendingMachine` class and how it can be improved.
- b) What will be a relevant class to relate to the `VendingMachine` class?
- c) Suggest how the application can be re-designed to involve the class in (b)?
- d) [Optional] Implement your design.

A sample program run is given below:

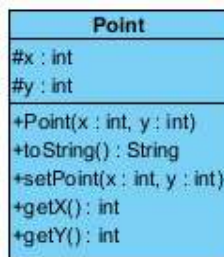
```
===== Vending Machine =====
|1. Buy Beer ($3.00)           |
|2. Buy Coke ($1.00)          |
|3. Buy Green Tea ($5.00)      |
|=====|
Please enter selection:
1
Please insert coins:
===== Coins Input =====
```

```

|Enter 'Q' for ten cents input |
|Enter 'T' for twenty cents input|
|Enter 'F' for fifty cents input |
|Enter 'N' for a dollar input   |
=====
Q
Coins inserted: 0.10
T
Coins inserted: 0.30
F
Coins inserted: 0.80
N
Coins inserted: 1.80
N
Coins inserted: 2.80
N
Coins inserted: 3.80
Change: $ 0.80
Please collect your drink
Thank You !!

```

2. You are given the class diagram for the Point class :



The `toString()` method will return the x and y value in the format “[x , y]”.

Write the code in Java.

Create a class `Circle` to inherit from the `Point` class. The `Circle` class is to have the following methods: **setRadius**, **getRadius**, **toString** and **area**. Reuse whatever you can from the `Point` class.

Create a class `Cylinder` to inherit from the any of the classes above. The `Cylinder` class is to have the following methods: **setHeight**, **getHeight**, **toString**, **area** and **volume**.

Draw the class hierarchy. Create and use instances of a circle and a cylinder to test classes you have created. Do you think that it was a good choice to use `Point` as the base class? Suggest alternatives.