

Home task for developer – Vending machine

2014 / Māris Lazdiņš

Relax, we'll help you.





Par If

- **If P&C Insurance Holding Ltd (If)** ir lielākā riska apdrošināšanas kompānija Ziemeļeiropā, kā arī vadošā apdrošināšanas kompānija Baltijā.
- If Latvijā šobrīd aizņem **11.6 %** no tirgus, ierindojo to **Top 4** vidū. 15 gadu pieredze tirgū.
- If piedāvā **pilnu riska apdrošināšanas pakalpojumu klāstu** gan privātpersonām, gan uzņēmumiem Zviedrijā, Somijā, Norvēģijā, Dānijā, Baltijas valstīs un Krievijā.
- If caur riska kontroli rūpējas par **klientu finansiālo drošību un stabilitāti pirms un pēc** apdrošināšanas gadījuma iestāšanās.

Relax, we'll help you.



Vending machine

Think about it as a physical machine:



Relax, we'll help you.





Task

Implement Vending machine in code using TDD approach

There are following features:

- You can update product list at any time
- You can insert coins, get coins back and get remainder
- You can buy 1 product at once for inserted coins
- Machine accepts following coins: 5¢, 10¢, 20¢, 50¢, 1 € and 2 €

Relax, we'll help you.



We are giving Vending machine interface:

```
public interface IVendingMachine
{
    /// <summary>Vending machine manufacturer.</summary>
    string Manufacturer { get; }

    /// <summary>Amount of money inserted into vending machine. </summary>
    Money Amount { get; }

    /// <summary>Products that are sold.</summary>
    Product[] Products { get; set; }

    /// <summary>Inserts the coin into vending machine.</summary>
    /// <param name="amount">Coin amount.</param>
    Money InsertCoin(Money amount);

    /// <summary>Returns all inserted coins back to user.</summary>
    Money ReturnMoney();

    /// <summary>Buys product from list of product.</summary>
    /// <param name="productNumber">Product number in vending machine product list.
    </param>
    Product Buy(int productNumber);
}
```

We are giving Vending machine interface:

```
public struct Money
{
    public int Euros { get; set; }
    public int Cents { get; set; }
}

public struct Product
{
    /// <summary>Gets or sets the available amount of product.</summary>
    public int Available { get; set; }

    /// <summary>Gets or sets the product price.</summary>
    public Money Price { get; set; }

    /// <summary>Gets or sets the product name.</summary>
    public string Name { get; set; }
}
```

Relax, we'll help you.



Take into account

- User Visual Studio 2012 express or any paid and/or newer version if available
 - Use TDD approach
 - Use C# language
 - Think about OOP design patterns and S.O.L.I.D. principles
 - In case of error throw exception of different type for each situation
 - Comments and code must be in English language
 - No need for UI
-
- As a result I need source code of this task



Have a nice day!

Māris Lazdiņš

Software Development lead

M. 26598742

E-mail: maris.lazdins@if.lv

Relax, we'll help you.

