## Упражнение по Софтуерни архитектури Шаблони за проектиране

## Задача 1

Нека е даден динамичен списък, дефиниран по начина, показан по-долу. Създайте итератор за обхождане на списъка, използвайки Iterator шаблон за проектиране.

```
public class Node<T>
     public T Value;
     public Node<T> Next;
 public class MyList<T>
     private Node<T> head;
     private Node<T> tail;
     public MyList()
         head = null;
         tail = null;
     public Node<T> Add(T item)
         if (head == null)
         {
             head = new Node<T>();
             head.Value = item;
             tail = head;
         }
         else
             Node<T> newNode = new Node<T>();
             newNode.Value = item;
             tail.Next = newNode;
             tail = newNode;
         return tail;
     }
     public void PrintList()
```

```
Console.WriteLine("\n List content is:");
var node = head;
while (node != null)
{
        Console.Write(String.Format("{0} ", node.Value));
        node = node.Next;
}
}
```

## Задача 2

Използвайки шаблона Chain of Responsibilities, реализирайте алгоритъм за определяне на брой банкноти от различните купюри, които трябва да се предоставят на потребителя, когато той изтегля пари от банкомат. Банкоматът разполага с касети, в които се съхраняват банкноти с копюри от 5, 10, 20, 50 и 100 евро. При първоначалната инициализация на банкомата се зареждат броя банкноти във всяка касетка. При всяко изтегляне на пари броят на банкноти от съответните касети намалява. Изберете условие А или условие Б

- А. Реализирайте по-лесния алгоритъм, при който се започва от най-големите копюри и се преминава към по-малките, с цел формиране на точната сума. Пример 400 евро се разпределят в 4 банкноти по 100 евро.
- Б. Реализирайте по-сложен алгоритъм, при който сумата се разпределя по-оптимално например при подадена заявка за 400 евро, потребителят получава половината сума в едри банкноти, а другата половина в по-дребни например 2 банкноти по 100 евро и 4 банкноти по 50 евро.