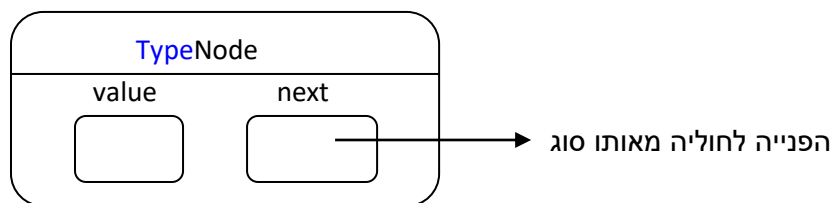


## מחלקה הקונקרטית למחלקה הבסיסית

<p><b>IntNode</b></p> <p><b>IntNode (int x)</b>  <b>IntNode (int x, IntNode next)</b></p> <p><b>int</b> GetValue ()  <b>IntNode</b> GetNext ()</p> <p><b>boolean</b> HasNext ()</p> <p><b>void</b> SetValue (<b>int</b> x)  <b>void</b> SetNext (<b>IntNode</b> next)</p> <p><b>override string</b> ToString ()</p>	<p><b>StringNode</b></p> <p><b>StringNode (string x)</b>  <b>StringNode (string x, StringNode next)</b></p> <p><b>string</b> GetValue ()  <b>StringNode</b> GetNext ()</p> <p><b>boolean</b> HasNext ()</p> <p><b>void</b> SetValue (<b>string</b> x)  <b>void</b> SetNext (<b>StringNode</b> next)</p> <p><b>override string</b> ToString ()</p>	<p><b>CharNode</b></p> <p><b>CharNode (char x)</b>  <b>CharNode (char x, CharNode next)</b></p> <p><b>char</b> GetValue ()  <b>CharNode</b> GetNext ()</p> <p><b>boolean</b> HasNext()</p> <p><b>void</b> SetValue (<b>char</b> x)  <b>void</b> SetNext (<b>CharNode</b> next)</p> <p><b>override string</b> ToString ()</p>
<p><b>DoubleNode</b></p> <p><b>DoubleNode (double x)</b>  <b>DoubleNode (double x, DoubleNode next)</b></p> <p><b>double</b> GetValue ()  <b>DoubleNode</b> GetNext ()</p> <p><b>boolean</b> HasNext ()</p> <p><b>void</b> SetValue (<b>double</b> x)  <b>void</b> SetNext (<b>DoubleNode</b> next)</p> <p><b>override string</b> ToString ()</p>	<p><b>PointNode</b></p> <p><b>PointNode (Point x)</b>  <b>PointNode (Point x, PointNode next)</b></p> <p><b>Point</b> GetValue ()  <b>PointNode</b> GetNext ()</p> <p><b>boolean</b> HasNext ()</p> <p><b>void</b> SetValue (<b>Point</b> x)  <b>void</b> SetNext (<b>PointNode</b> next)</p> <p><b>override string</b> ToString ()</p>	<p><b>Node&lt;T&gt;</b></p> <p><b>Node (T x)</b>  <b>Node (T x, Node&lt;T&gt; next)</b></p> <p><b>T</b> GetValue ()  <b>Node&lt;T&gt;</b> GetNext ()</p> <p><b>boolean</b> HasNext ()</p> <p><b>void</b> SetValue (<b>T</b> x)  <b>void</b> SetNext (<b>Node&lt;T&gt;</b> next)</p> <p><b>override string</b> ToString ()</p>



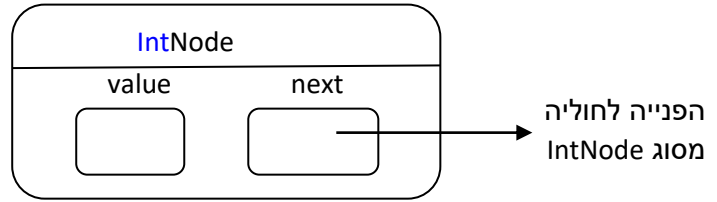
שכפול קוד: לכל סוג רשימה יש להחזיק מחלקה מתאימה

המחלקה הקונקרטית

```

Class IntNode
{
    private int value ;
    private IntNode next;
    :
}

```



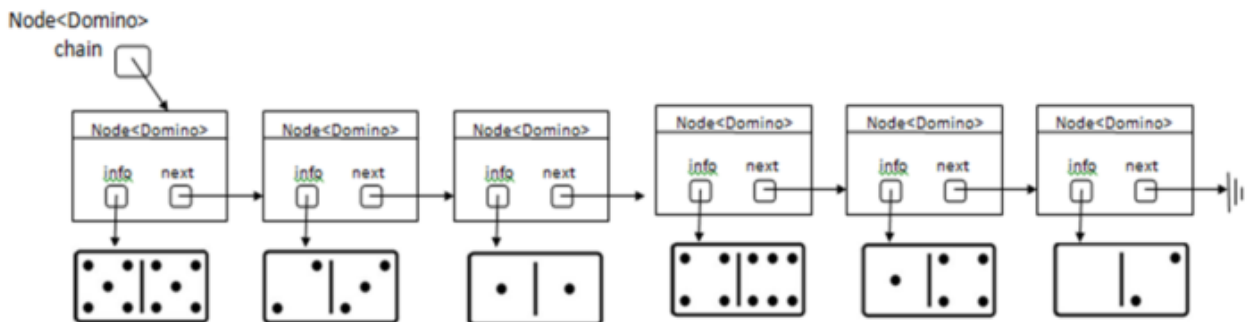
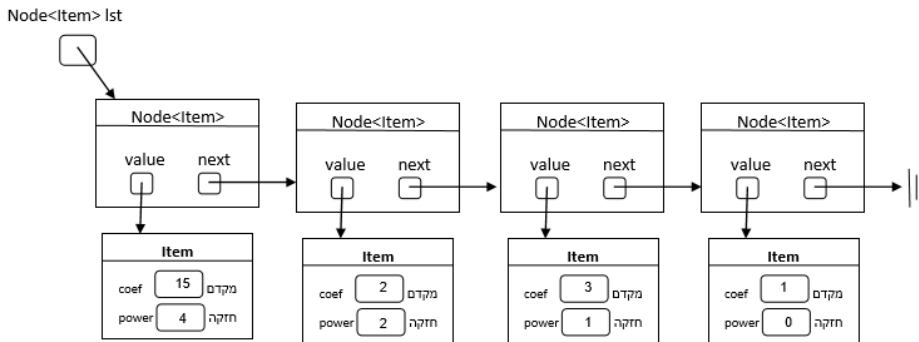
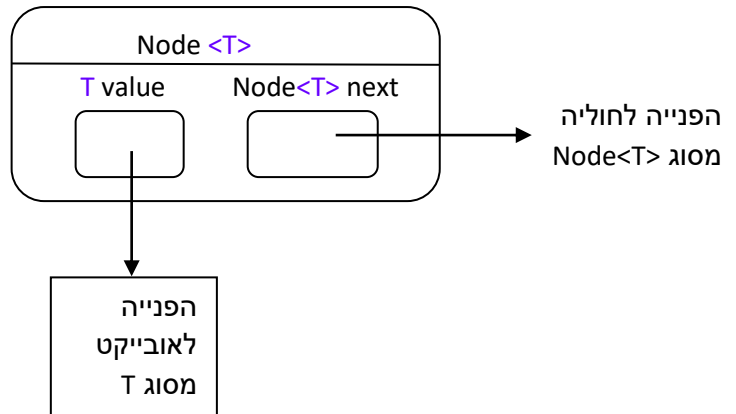
המחלקה הסנכרית

מחלקה אחת מתאימה להרבה טיפוסים נתונים

```

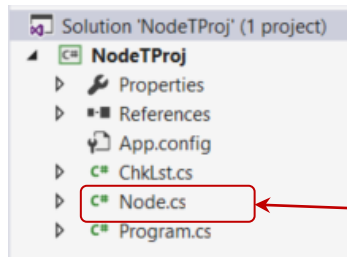
Class Node<T>
{
    private T value ;
    private Node<T> next;
    :
}

```



**ממשק המחלקה: Node**

Node (T x)	הפעולה בונה חוליה. ערך החוליה הוא x ואין לה חוליה עוקבת
Node (T x, Node<T> next)	הפעולה בונה חוליה. ערך החוליה הוא x והחוליה העוקבת לה הוא next. next יכול להיות גם null
T GetValue ()	פעולה המחזירה את ערך החוליה
Node<T> GetNext ()	פעולה המחזירה הפניה לחוליה העוקבת. אם אין חוליה עוקבת תחזיר הפעולה null
void SetValue (T x)	פעולה המשנה את ערך החוליה להיות x
void SetNext ()	פעולה המשנה את ההפניה לחוליה העוקבת להיות next. ערכו של next יכול להיות גם null
bool HasNext ()	פעולה המחזירה אמת אם יש חוליה עוקבת לחוליה הנוכחית, ושקר אחרת
string ToString ()	פעולה המחזירה מחרוזת המתארת את הערך השמור בחוליה



**דגשים בכתיבת המחלקה**

שם המחלקה Node

צור מחלקה בשם Node (! <T>)

**class Node<T>** : כותרת המחלקה

הוסף <T> לכותרת מהחלקה

`class Node<T>`

```
private T value; // הפנייה לאובייקט מסוג T
private Node<T> next; // הפניה לחוליה הבאה
```

//---- פעולות בונות ----

`public Node(T x)`

```
this.value = x;
this.next = null;
```

`public Node(T x, Node<T> next)`

```
this.value = x;
this.next = next;
```

שם הבנאי: **public Node()**  
ללא <T>

## דגים בשימוש במחלקה האנרית

הטיפוס הגנרי T ירשם רק במחלקה Node<T> מחוץ למחלקה יש להשתמש בטיפוס הקונקרטי.

כותרת פעולה היוצרת רשימה / שרשרת חוליות חדשה תיראה כך :

✓ `Node<int> numbers = new Node <int> (5);`

✗ `Node<T> numbers = new Node <T> (5);` ולא כך :

```
Node<int> numbers = new Node <int> (5);
Node<char> word = new Node<char> ('a');
Node <double> lst = new Node <double> (4.25);
Node <Point> dots = new Node <Point> (new Point (3, 6));
```