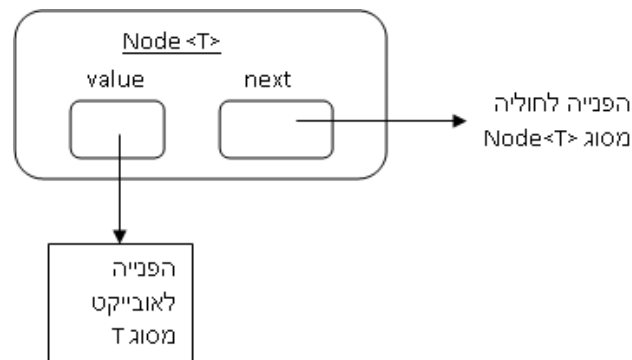


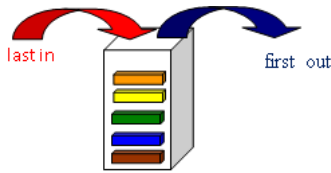
מבנים נתונים

בתכנית הלימודים החדשה

החוליה הגנרית Node<T>

סיבוכיות	חתימת הפעולה	תיאור הפעולה
		בנאים:
O(1)	Node (T x)	פעולה הבונה חוליה שבערך value שלה יהיה x וב- next שלה יהיה הערך null
O(1)	Node (T x , Node<T> next)	פעולה הבונה חוליה שבערך value שלה יהיה x וב- next שלה יהיה הערך next (ערך next המועבר כפרמטר יכול להיות גם null)
		שאלות:
O(1)	T GetValue ()	אם T מחלקה עוטפת לטיפוס בסיסי (Integer, Double, Character) יוחזר ערך החוליה, ואם הפניה לעצם, תוחזר הפניה לעצם זה
O(1)	Node<T> GetNext ()	מוחזרת הפניה לחוליה הבאה
O(1)	bool HasNext ()	פעולה המחזירה אמת אם next מפנה לחוליה נוספת (כלומר אינו null) ושקר אחרת
O(1) (*)	string ToString ()	פעולה המחזירה מחרוזת המתארת את מצב העצם (*) סיבוכיות: אם T עצם מטיפוס פשוט $O(1)$ ואם T מייצג אוסף $O(T)$ (אורך האוסף כפונקציה של n)
		פקודות:
O(1)	void SetValue (T x)	הפעולה משנה (מעדכנת) את ערך ה- value של החוליה ל- x
O(1)	void SetNext (Node<T> next)	הפעולה משנה את ערכו של next להיות next חדש (ערך next המתקבל כפרמטר יכול להיות גם null)





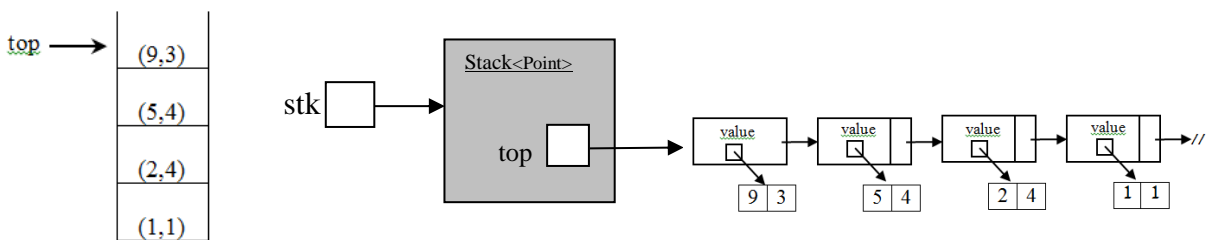
מחסנית גנרית Stack<T>

סיבוכיות	חתימת הפעולה	תיאור הפעולה
		בנאי:
O(1)	Stack ()	פעולה הבונה מחסנית ריקה
		שאלות:
O(1)	bool IsEmpty ()	מחסנית-ריקה? פעולה המחזירה אמת אם המחסנית ריקה ושקר אחרת
O(1)	T Top ()	הצף-למחסנית () הפעולה מחזירה את האיבר שבראש המחסנית מבלי להוציאו. תנאי קדם: המחסנית אינה ריקה
O(n· T) (*)	string ToString ()	פעולה המחזירה מחרוזת המתארת את מצב המחסנית באופן הבא: $[x_1, x_2, x_3, \dots, x_n]$ כך ש- x_1 בראש המחסנית ו- x_n בתחתיתה. (*) סיבוכיות: n מייצג את מספר האיברים במחסנית אם T עצם מטיפוס פשוט $O(n)$ ← ואם T מייצג אוסף באורך כלשהו $O(n \cdot T)$ ← (למשל: T הוא עצם מסוג תלמיד שלו אוסף ציונים) T אורך T
		פקודות:
O(1)	void Push (T x)	דחוף-למחסנית (x) הפעולה דוחפת/מוסיפה את x לראש המחסנית
O(1)	T Pop ()	שלוף-ממחסנית () הפעולה מוציאה ומחזירה את האיבר שבראש המחסנית. תנאי קדם: המחסנית אינה ריקה

משמעות **תנאי קדם**: הפעולה מניחה את תקינות הנתונים. התכנית/פעולה המזמנת את הפקודה חייבת לבדוק זאת. לדוגמה:

```
if (! s.IsEmpty() )
    x = s.Pop();
```

אם המחסנית-לא-ריקה?
שלוף-ממחסנית() ← x



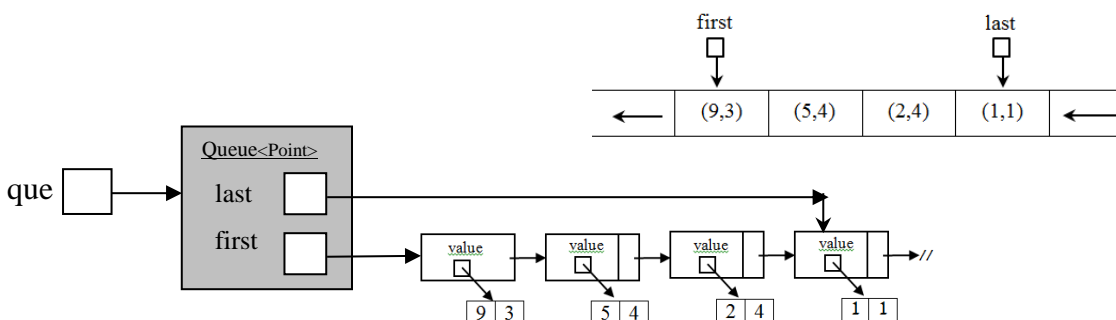
תור גנרי $Queue<T>$

סיבוכיות	חתימת הפעולה	תיאור הפעולה
		בנאי:
$O(1)$	<code>Queue ()</code>	פעולה הבונה תור ריק
		שאלות:
$O(1)$	<code>bool IsEmpty ()</code>	תור-ריק? פעולה המחזירה אמת אם התור ריק ושקר אחרת
$O(1)$	<code>T Head ()</code>	ראש-התור () הפעולה מחזירה את האיבר שבראש התור מבלי להוציאו. תנאי קדם: התור לא ריק
$O(n \cdot T)$ (*)	<code>string ToString ()</code>	פעולה המחזירה מחרוזת המתארת את מצב התור באופן הבא: $[x_1, x_2, x_3, \dots, x_n]$ כך ש- x_1 בראש התור ו- x_n בסופו. (איבר חדש יוכנס אחרי x_n) (*) סיבוכיות: n מייצג את מספר האיברים בתור אם T עצם מטיפוס פשוט $\leftarrow O(n)$ ואם T מייצג אוסף באורך כלשהו $\leftarrow O(n \cdot T)$ אורך T (למשל: תור הדפסות: T הוא עצם מסוג מסמך להדפסה שיש בו n דפים)
		פקודות:
$O(1)$	<code>void Insert (T x)</code>	הכנס-לתור (x) הפעולה מוסיפה את x לסוף התור
$O(1)$	<code>T Remove ()</code>	הוצא-מהתור () הפעולה מוציאה ומחזירה את האיבר שבראש התור. תנאי קדם: התור לא ריק

משמעות **תנאי קדם**: הפעולה מניחה את תקינות הנתונים. התכנית/פעולה המזמנת את הפקודה חייבת לבדוק זאת. לדוגמה:

```
if (! q.IsEmpty ())
    x = q.Remove ();
```

```
אם התור-לא-ריק?
הוצא-מהתור ()  $\leftarrow x$ 
```





חוליה בינארית $\text{BinNode}\langle T \rangle$

חוליה המכילה ערך מטיפוס T והפנייה לשתי חוליות בינאריות. ניתן להשתמש במחלקה זאת כדי לייצג עץ בינארי או שרשרת דו-כיוונית.

סיבוכיות	חתימת הפעולה	תיאור הפעולה
		בנאי:
$O(1)$	$\text{BinNode}(T\ x)$	פעולה הבונה חוליה בינארית שערכה x ושתי ההפניות null
$O(1)$	$\text{BinNode}(\text{BinNode}\langle T \rangle\ \text{left}, T\ x, \text{BinNode}\langle T \rangle\ \text{right})$	פעולה הבונה חוליה בינארית שערכה x , ושתי ההפניות שלה הן left ו-right בהתאמה (ערכם של הפרמטרים left ו-right יכול להיות null)
		שאלות:
$O(1)$	$T\ \text{GetValue}()$	אם T מחלקה עוטפת לטיפוס בסיסי (Integer, Double, Character) יוחזר ערך החוליה, ואם הפניה לעצם, תוחזר הפנייה לעצם זה
$O(1)$	$\text{BinNode}\langle T \rangle\ \text{GetLeft}()$	פעולה המחזירה הפנייה לחוליה אליה מפנה left.
$O(1)$	$\text{BinNode}\langle T \rangle\ \text{GetRight}()$	פעולה המחזירה הפנייה לחוליה אליה מפנה right.
$O(1)$	bool HasLeft()	פעולה המחזירה אמת אם יש חוליה משמאל, ושקר אחרת
$O(1)$	bool HasRight()	פעולה המחזירה אמת אם יש חוליה מימין, ושקר אחרת
$O(T)$	string ToString()	פעולה המחזירה את מצב החוליה כמחרוזת. (*) סיבוכיות: אם T עצם מטיפוס פשוט $\leftarrow O(1)$ ואם T מייצג אוסף באורך כלשהו $\leftarrow O(T)$
		פקודות:
$O(1)$	void SetValue($T\ x$)	הפעולה משנה (מעדכנת) את ערך החוליה ל- x
$O(1)$	void SetLeft($\text{BinNode}\langle T \rangle\ \text{left}$)	הפעולה משנה את ערכה של ההפניה left ל-left הפרמטר left יכול להיות גם null
$O(1)$	void SetRight($\text{BinNode}\langle T \rangle\ \text{right}$)	הפעולה משנה את ערכה של ההפניה right ל-right הפרמטר right יכול להיות גם null

