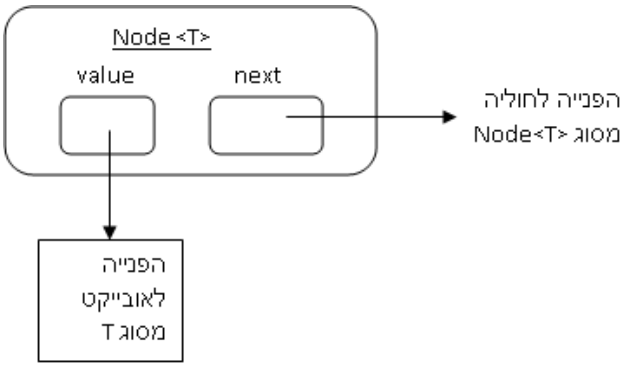
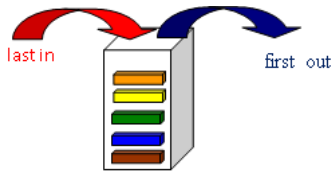
ממשקים למבני הנתונים
בתכנית הלימודים החדשה

החוליה הגנרית Node<T>

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





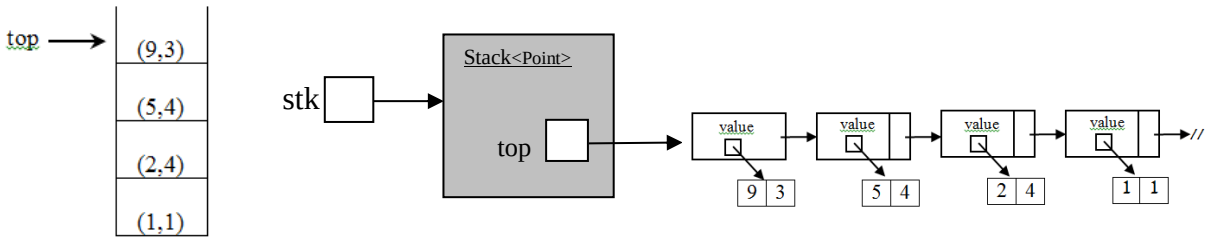
מחסנית גנרית Stack<T>

תיאור הפעולה	חתימת הפעולה	סיבוכיות
בנאי:		
פעולה הבונה מחסנית ריקה	Stack ()	O(1)
שאילתות:		
מחסנית-ריקה? פעולה המחזירה אמת אם המחסנית ריקה ושקר אחרת	**bool** IsEmpty ()	O(1)
הצץ-למחסנית () הפעולה מחזירה את האיבר שבראש המחסנית מבלי להוציאו. תנאי קדם: המחסנית אינה ריקה	T Top ()	O(1)
פעולה המחזירה מחרוזת המתארת את מצב המחסנית באופן הבא: $[x_1, x_2, x_3, \ldots, x_n]$ 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•|T|) (למשל: T הוא עצם מסוג תלמיד שלו אוסף ציונים) |T| אורך T	**string** ToString ()	O(n•|T|) (*)
פקודות:		
דחוף-למחסנית (x) הפעולה דוחפת/מוסיפה את x לראש המחסנית	**void** Push (T x)	O(1)
שלוף-ממחסנית () הפעולה מוציאה ומחזירה את האיבר שבראש המחסנית. תנאי קדם: המחסנית אינה ריקה	T Pop ()	O(1)

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

אם המחסנית-לא-ריקה?
שלוף-ממחנית() ← x

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



תור גנרי Queue<T>

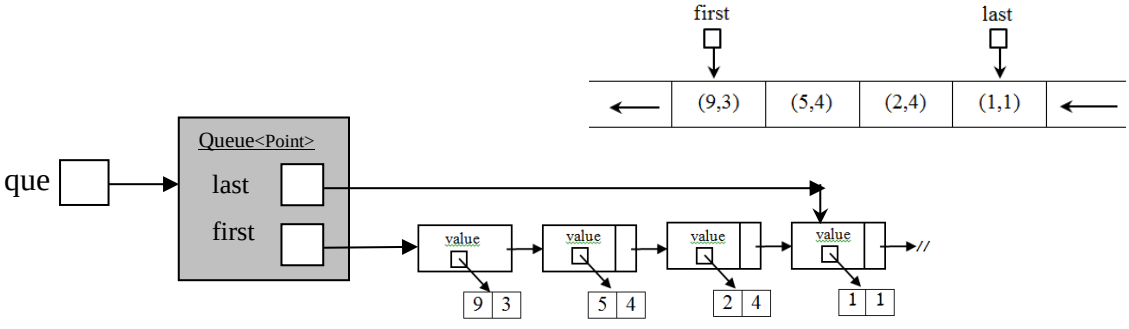


תיאור הפעולה	חתימת הפעולה	סיבוכיות
בנאי:		
פעולה הבונה תור ריק	Queue ()	O(1)
שאילתות:		
תור-ריק? פעולה המחזירה אמת אם התור ריק ושקר אחרת	**bool** IsEmpty ()	O(1)
ראש-התור () הפעולה מחזירה את האיבר שבראש התור מבלי להוציאו. תנאי קדם: התור לא ריק	T Head ()	O(1)
פעולה המחזירה מחרוזת המתארת את מצב התור באופן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

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

אם התור-לא-ריק?
x ← הוצא-מהתור ()

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





חוליה בינארית BinNode<T>

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

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

