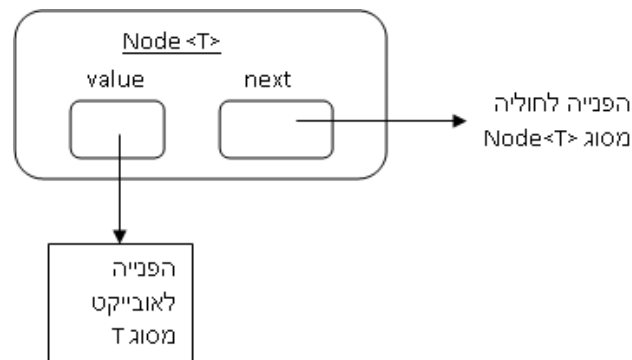


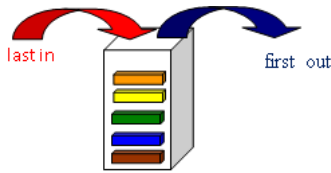
## מבנים נתונים

### בתכנית הלימודים החדשה

#### החוליה הגנרית Node<T>

סיבוכיות	חתימת הפעולה	תיאור הפעולה
		<b>בנאים:</b>
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 המועבר כפרמטר יכול להיות גם null)
		<b>שאלות:</b>
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)	boolean 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 <b>(*) סיבוכיות:</b> אם T עצם מטיפוס פשוט $O(1)$ ← ואם T מייצג אוסף $O( T )$ ← (אורך האוסף כפונקציה של n)
		<b>פקודות:</b>
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)





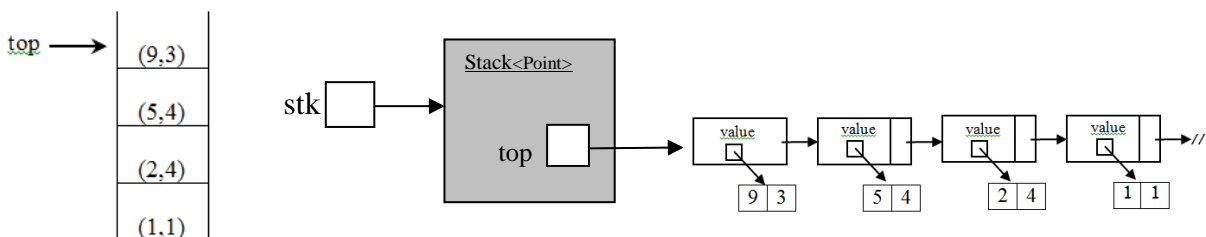
מחסנית גנרית Stack<T>

סיבוכיות	חתימת הפעולה	תיאור הפעולה
		<b>בנאי:</b>
O(1)	Stack ()	פעולה הבונה מחסנית ריקה
		<b>שאלות:</b>
O(1)	boolean 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 )
		<b>פקודות:</b>
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
```





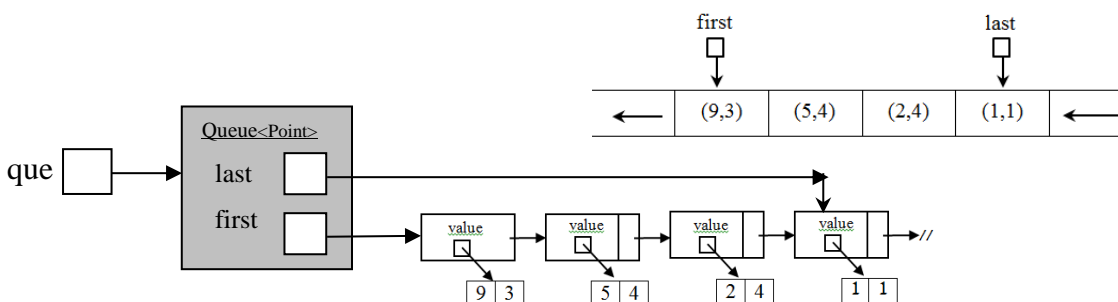
תור גנרי Queue<T>

סיבוכיות	חתימת הפעולה	תיאור הפעולה
		<b>בנאי:</b>
O(1)	Queue ()	פעולה הבונה תור ריק
		<b>שאלות:</b>
O(1)	boolean isEmpty ()	תור-ריק? פעולה המחזירה אמת אם התור ריק ושקר אחרת
O(1)	T head ()	ראש-התור () הפעולה מחזירה את האיבר שבראש התור מבלי להוציאו. תנאי קדם: התור לא ריק
O(n· T )  (*)	String toString ()	פעולה המחזירה מחרוזת המתארת את מצב התור באופן הבא: [x <sub>1</sub> , x <sub>2</sub> , x <sub>3</sub> , ..., x <sub>n</sub> ] כך ש-x <sub>1</sub> בראש התור ו-x <sub>n</sub> בסופו. (איבר חדש יוכנס אחרי x <sub>n</sub> )  (*) סיבוכיות: n מייצג את מספר האיברים בתור אם T עצם מטיפוס פשוט ← O(n) ואם T מייצג אוסף באורך כלשהו ← O(n· T ) (אורך T) (למשל: תור הדפסות: T הוא עצם מסוג מסמך להדפסה שיש בו n דפים)
		<b>פקודות:</b>
O(1)	void insert (T x)	הכנס-לתור (x) הפעולה מוסיפה את x לסוף התור
O(1)	T remove ()	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 () ← x
```





### חוליה בינארית <T> BinNode

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
		<b>בנאי:</b>
O(1)	BinNode (T x)	פעולה הבונה חוליה בינארית שערכה x ושתי ההפניות null
O(1)	BinNode (BinNode<T> left, T x, BinNode<T> right)	פעולה הבונה חוליה בינארית שערכה x, ושתי ההפניות שלה הן left ו- right (ערכם של הפרמטרים left ו- right יכול להיות null)
		<b>שאלות:</b>
O(1)	T getValue ()	אם T מחלקה עוטפת לטיפוס בסיסי (Integer, Double, Character) יוחזר ערך החוליה, ואם הפניה לעצם, תוחזר הפנייה לעצם זה
O(1)	BinNode <T> getLeft ()	פעולה המחזירה הפנייה לחוליה אליה מפנה left.
O(1)	BinNode <T> getRight ()	פעולה המחזירה הפנייה לחוליה אליה מפנה right.
O(1)	boolean hasLeft ()	פעולה המחזירה אמת אם יש חוליה משמאל, ושקר אחרת
O(1)	boolean hasRight ()	פעולה המחזירה אמת אם יש חוליה מימין, ושקר אחרת
O( T )	String toString ()	פעולה המחזירה את מצב החוליה כמחרוזת. <b>(*) סיבוכיות:</b> אם T עצם מטיפוס פשוט $\leftarrow O(1)$ ואם T מייצג אוסף באורך כלשהו $\leftarrow O( T )$
		<b>פקודות:</b>
O(1)	void setValue (T x)	הפעולה משנה (מעדכנת) את ערך החוליה ל- x
O(1)	void setLeft (BinNode <T> left)	הפעולה משנה את ערכה של ההפניה left ל- left הפרמטר left יכול להיות גם null
O(1)	void setRight (BinNode <T> right)	הפעולה משנה את ערכה של ההפניה right ל- right הפרמטר right יכול להיות גם null

