

## 202 - מבנה נתונים

### קורסי קדם חובה :

1. תכנות ואלגוריתמיקה ( שפת JAVA או C# או שפת תכנות אחרת)
2. מומלץ- תכנות מונחה עצמים ( או נלמד במקביל)
3. תגבור מתמטיקה

**היקף שעות :** 80 ( הסליבוס כולל 90 שעות, ההרחבה מומלצת)

**מטרת הקורס :** היכרות מאפייני טיפוס מבני נתונים, פעולות על מבני הנתונים, שילוב מבני נתונים בפתרון בעיה אלגוריתמית ביעילות טובה ככל שניתן.

### תקציר הקורס

תכנית מחשב = אלגוריתם + מבנה נתונים.

המשמעות היישומית של משוואה זו : כתיבה של אלגוריתם לפתרון בעיה, דורש חשיבה על דרישות הקלט, פירוק הפתרון למרכיבים, וגישה לנתונים, גישה שביצוגה הפשוט, כותב התכנית נדרש לשימוש במשתנים בסיסיים, בבעיות מורכבות יותר, כותב האלגוריתם נדרש לאחסן אוסף נתונים, נדרש לגישה לאוסף הנתונים, לחיפוש נתון באוסף, מחיקה או אוספת נתון לאוסף, מיון אוסף הנתונים ופעולות נוספות. כלל פעולות אלו המתבצעות על אוסף נתונים גדול מאוד, עלול לדרוש זמן רב, והיעד של מעצב האלגוריתם הוא להקטין את הזמן, מאגרי הנתונים במערכות רבות הם עצומים בגודלם, מיליארדי רשומות של נתונים, סיבוכיות האלגוריתמים לעיבוד מאגרי ענק אלו, חשובה מאוד וכלל הפעולות על מאגרי הנתונים צריכות להתבצע בזמן סביר. הקורס מציג מבני נתונים שונים, ורכישת ידע על מבנים אלו ורכישת מיומנויות כתיבת קוד שיעדם, ביצוע פעולות על מבני הנתונים באופן יעיל. כמה יעיל, האם יעילות מסויימת היא היעילות הטובה ביותר שניתן להשיג, איך אנו משווים יעילות של אלגוריתמים, שאלות אלו ושאלות ונושאים נוספים עומדים בליבת קורס חשוב זה.

### סביבות עבודה : חומרה ותוכנה נדרשות

#### כלי תוכנה נדרשים

מס'	שם התוכנה	תיאור קצר	הערות
1	Eclipse	סביבת עבודה ב-JAVA	הקורס ניתן ליישום ב-C#
2	Jeliot	סביבת הדמייה ומעקב אחר קוד ב-JAVA	אתר : <a href="http://cs.joensuu.fi/jeliot/">http://cs.joensuu.fi/jeliot/</a>

### חומרה נדרשת

מס'	שם החומרה	תיאור קצר	הערות
1	מחשב בסיסי עם חיבור לרשת		

### תוכן הקורס

מס' סעיף	תת סעיף	נושא	שעות הרצאה	שעות תרגול	מיומנות נלמדת
1		חזרה ותרגול בתמ"ע, דגש על שימוש במחלקה נתונה על בסיס ממשק הפעולות	2	4	כתיבת מחלקה, הגדרת טיפוס נתונים
	1.1	הצגת 2-3 מחלקות, תרגול ביצירת עצמים, זימון פעולות.	1	2	זימון פעולות בגישת "ייחוס נקודה" - dot notation
	1.2	רענון מערכים, יצירה, גישה לאיבר, סריקה. מומלץ התנסות במערך של עצמים : מערך מחרוזות, מערך של עצמים מסוג point תרגול בגישה לאיבר במערך(עצם) וזימון פעולות	1	2	הגדרה הפנייה למערך. יצירת מערך. גישה לאורך המערך. הגדרה ויצירת מערך הפניות.
2		פעולות על מערך - טיפוס נתונים סדרתי	5	5	
	2.1	פעולת גישה לאיבר במערך, פעולת סריקה לינארית של מערך, מחיקת איבר, עדכון איבר	2		ביצוע פעולות על מערך : סריקה, חיפוש, חישוב מקסימום, מינימום, סכום, מנייה ועוד. כתיבת פעולה שמקבלת הפנייה למערך, פעולה שמחזירה הפנייה למערך
	2.2	חיפוש לינארי במערך	1	1	כתיבת אלגוריתם חיפוש.
	2.3	חיפוש בינארי במערך ממוין, השוואה "אינטואיטיבית" בין חיפוש ערך עם סריקה לינארית, לבין חיפוש בינארי במערך ממוין	1	2	מימוש אלגוריתם חיפוש בינארי בפעולה המקבלת מערך ממוין וערך לחיפוש

מס' סעיף	תת סעיף	נושא	שעות הרצאה	שעות תרגול	מיומנות נלמדת
	2.4	פעולת מיון מערך: מיון הכנסה, מיון בועות, מיוזוג מערכים ממוינים	1	2	תרגול של לפחות 2 אלגוריתמי מיון.
3		<b>מחלקה גנרית</b>	2	2	
	3.1	רציונל ונחיצות של מחלקה גנרית, דוגמאות	2		
	3.2	שימוש במחלקה גנרית עם תכונה אחת יצירת עצמים של המחלקה עם טיפוסים שונים		2	יצירת עצמים ממחלקה גנרית כאשר כל עצם type שונה
4		<b>רקורסיה</b>	5	7	
	4.1	הצגת רעיון של פעולה שקוראת לעצמה דוגמאות בסיסיות לפעולות רקורסיביות שלא מחזירות ערך	2		
	4.2	רקורסיה זנב: מאפיין: זימון רקורסיבי בשורה האחרונה של הפונקציה. דוגמאות: הדפסת תווים של מחרוזת בסדר הפוך. הדפסת ערכים עבור קלט n: n..1 או 1...n ביצוע מעקב אחר רקורסיה באמצעות טבלת מעקב או עץ קריאות, מעקב עם שימוש בסביבת האימוולציה Jeliot		3	יכולת לבצע מעקב אחר ביצוע רקורסיה באמצעות: 1: טבלת מעקב 2: עץ קריאות ניתוח מעיה, כתיבה ניסוח רקורסיבי לפתרון רקורסיבי (רקורסיית זנב) יישום הפתרון בשפת תכנות בדיקת נכונות

מס' סעיף	תת סעיף	נושא	שעות הרצאה	שעות תרגול	מיומנות נלמדת
	4.3	<p>פונקציה רקורסיבית "הלוך ושוב"</p> <p>פונקציה בה הקריאה הרקורסיבית אינה ההוראה האחרונה בקוד.</p> <p>מעקב אחר פונקציה רקורסיבית "הלוך ושוב"</p> <p>דוגמא:</p> <p>חישוב עצרת</p> <p>חישוב סכום של סדרת הערכים <math>n..1</math></p> <p>פונקציה רקורסיבית חיפוש תו במחרוזת:</p> <pre>public static Boolean seek(String s, char t){     if(st.length==0)         return false;     if(st.charAt(0) == t)         return true;     return seek(s.substring(1,tav) }</pre> <p>תרגול הדגמה:</p> <p>פעולה רקורסיבית המקבלת מספר שלם, מחזירה אמת, אם כל הספרות במספר זוגיות</p> <p>פעולה רקורסיבית שמקבלת שני מספרים <math>a, b</math></p> <p>הפעולה מחזירה אמת אם <math>b</math> מחלק את <math>a</math> ללא שארית, שקר אחרת.</p> <p>פעולה רקורסיבית המחזירה את המחלק המשותף הגדול ביותר של שני מספרים שלמים</p> <p>פעולה רקורסיבית המקבלת מספר שלם ומחזירה את סכום ספרותיו</p>	2	3	<p>פתרון בעיות על מספרים שלמים ברקורסיה:</p> <p>סכום ספרות במספר</p> <p>מספר ספרות במספר</p> <p>מציאת מחלק משותף.</p> <p>סיפרה הגדולה ביותר במספר.</p> <p>כתיבת רקורסיה עם מחרוזות:</p> <p>הדפסה מדורגת של מחרוזות.</p> <p>האם מחרוזות היא פילנדרום?</p>

מס' סעיף	תת סעיף	נושא	שעות הרצאה	שעות תרגול	מיומנות נלמדת
	4.4	רקורסיה כפולה : רקורסיה הכוללת שני זימונים : פעולה רקורסיבית לחישוב האיבר ה-n של סדרת פיבונצ'י. פעולה רקורסיבית לפתרון בעיית מגדלי הנוי רקורסיה עם מערכים	1	1	יישום : חישוב איבר בסדרת פיבונצ'י יישום פתרון בעיות מגדלי הנוי
5		<b>יעילות</b>	5	6	
	5.1	הגדרה : פעולת יסוד - הוראה המתבצעת קבוע דוגמא : $x=7$ ; הוראה המתבצעת בזמן תלוי n דוגמא : $while(x<n)\{$ $\}$ חישוב פונקציית זמן ריצה עבור קוד נתון – $T_n$ סדר גודל של זמני ריצה $o(n), o(n_2), o(n!), o(\log_2 n), o(n^2)$	2	2	קבלת קוד וחישוב זמן ריצה של הקוד כפונקציה של n. מעבר בביטוי של זמן ריצה לביטוי של דרגת סיבוכיות. יכולת השוואת יעילות בין שני אלגוריתמים
	5.2	דוגמאות קוד וחישוב זמן ריצה $T_n$ וסיבוכיות זמן ריצה. מנייה, סכום. עבור מבנה נתונים מערך.	1	2	ביצוע חישוב עבור אלגוריתמי חיפוש, מנייה, סכום. עבור מבנה נתונים מערך.
	5.3	חישוב סיבוכיות עבור : הזמן הממוצע של ריצת האלגוריתם הזמן הגרוע ביותר של ריצת האלגוריתם הזמן הטוב ביותר של ריצת האלגוריתם הבחירה וההשוואה בין יעילות של אלגוריתמים, מבוסס על זמן ריצה הגרוע ביותר	1		

מס' סעיף	תת סעיף	נושא	שעות הרצאה	שעות תרגול	מיומנות נלמדת
	5.4	<p>שיפור יעילות אלגוריתם :</p> <p>שיפור בקבוע</p> <p>שיפור בסדר גודל.</p> <p>דוגמא : בעיית בדיקה האם מספר שלם כלשהו הוא מספר ראשוני :</p> <p>בדיקה בלולאה : סיבוכיות <math>O(n)</math></p> <p>בדיקה בלולאה, דילוג על בדיקת מספרים זוגיים :</p> <p>זמן הריצה קטן במחצית – סדר הגודל לא השתנה – <math>O(n)</math></p> <p>בדיקה בלולאה עד ל- <math>\sqrt{n}</math>, סיבוכיות <math>O(\sqrt{n})</math></p> <p>שיפור בסדר גודל.</p>	1	2	<p>בהרצאה הצגת דוגמא : בדיקה ראשוניות של מספר.</p> <p>מיומנות השוואה בין אלגוריתמים, חישוב סדרת הפרשים בין אורך קלט משתנה, הבנה מעמיקה של קצב הגידול של פונקציות שונות</p>
	5.5	<p>תרגול בנושא יעילות, כולל על פעולות המתבצעות עם מבני נתונים שונים :</p> <p>אוסף משורשר לינארית, מחסניות ותורים, מערכים, עצים בינארים</p>			<p>תרגול זה מתבצע לאורך כל הקורס עבור אלגוריתמים</p>
6		<p>מחסנית – Stack</p> <p>מחסנית גנרית</p> <p>דוגמא מחסנית שלמים :</p> <p>Stack&lt;Integer&gt; s</p> <p>דוגמא מחסנית מסוג Point</p> <p>Stack&lt;Point&gt; s</p> <p>מדיניות הכנסה והוצאה במחסנית : LIFO</p> <p>פעולות בסיסיות על מחסנית – ממשק פעולות</p> <p>פעולות חיצוניות על מחסנית</p> <p>פתרון בעיות באמצעות מבנה נתונים מחסנית</p> <p>סיבוכיות ומחסניות</p>	5	6	

מס' סעיף	תת סעיף	נושא	שעות הרצאה	שעות תרגול	מיומנות נלמדת
	6.1	מחסנית – מבנה נתונים מופשט. פעולות על מחסנית: יצירת מחסנית הוצאת איבר ממחסנית דחיפת איבר ממחסנית בדיקת מחסנית ריקה?	2	2	תרגול ביצוע פעולות על מחסנית, יצירת טבלת מעקב אחר מצב המחסנית. סריקת מחרוזת ודחיפת תווי המחרוזת למחסנית. שליפה תווים ממחסנית ויצירת המחרוזת. השוואה בין מצבי המחרוזת. הבנה הרעיון שהכנסת איברים למחסנית והוצאה הופכת את סדר ההופעה
	6.2	כתיבת פעולות חיצוניות למחסנית: מספר איברים במחסנית חיפוש ערך במחסנית שליפת איבר במקום $k$ ממחסנית, שמירה על המחסנית המקורית	1	2	כתיבת כותרת פעולה בהתאם לדרישת המטלה. יישם פעולות שונות על מחסנית. כתיבת פעולה המחזירה מבנה נתונים אחר (לדוגמא מקבלת מחסנית, ומחזירה את איברי המחסנית במערך)
	6.3	פתרון בעיות מיוחדות עם מחסנית חוקיות בעיית נכונות ביטוי מתמטי: {00} - האם סדר הסוגריים בביטוי חוקי בדיקה האם ביטוי מחרוזתי הוא פילנדרום: MADAMIMADAM רקורסיה עם מחסנית חישוב סיבוכיות על אלגוריתמים עם מחסנית	2	2	דגש על פעולות רקורסיה עם מחסנית. הבנת ההבדל כאשר מיקום הקריאה הרקורסיבית משתנה וההשפעה על הריצה. יכולת יצירת מעקב אחר ריצה של הקוד באמצעות טבלת מעקב או עץ קריאות

מס' סעיף	תת סעיף	נושא	שעות הרצאה	שעות תרגול	מיומנות נלמדת
7		<p>תור – Queue</p> <p>תור גנרי</p> <p>דוגמא מחסנית שלמים :</p> <p>Queue&lt;Double&gt;</p> <p>דוגמא תור מסוג String</p> <p>&lt;Point&gt; s</p> <p>מדיניות הכנסה והוצאה בתור : FIFO</p> <p>פעולות בסיסיות על תור – ממשק פעולות</p> <p>פעולות חיצוניות על תור</p> <p>פתרון בעיות באמצעות מבנה נתונים תור</p> <p>רקורסיה עם תורים</p> <p>סיבוכיות ותורים</p>	5	6	
	7.1	<p>תור – מבנה נתונים מופשט.</p> <p>פעולות על תור :</p> <p>יצירת תור</p> <p>הוצאת איבר תור</p> <p>דחיפת איבר תור</p> <p>בדיקת תור ריק?</p>	2	2	<p>תרגול ביצוע פעולות על תור, יצירת טבלת מעקב אחר מצב התור.</p> <p>סריקת מחרוזת ודחיפת תווי המחרוזת לתור.</p> <p>שליפה תוויים לתור ויצירת המחרוזת.</p> <p>השוואה בין מצבי המחרוזת.</p> <p>הבנה הרעיון שהכנסת איברים לתור והוצאתם אינה משנה את סדר ההופעה</p>



מס' סעיף	תת סעיף	נושא	שעות הרצאה	שעות תרגול	מיומנות נלמדת
	7.2	<p>כתיבת פעולות חיצוניות לתור :</p> <p>מספר איברים תור</p> <p>חיפוש ערך תור</p> <p>שליפת איבר במקום – k מתור, שמירה על התור המקורית.</p> <p>סכום תור</p> <p>היפוך איברי תור</p> <p>רקורסיה עם תורים</p>	1	2	<p>כתיבת כותרת פעולה בהתאם לדרישת המטלה.</p> <p>יישם פעולות שונות על תור.</p> <p>כתיבת פעולה המחזירה מבנה נתונים אחר (לדוגמא מקבלת תור, ומחזירה את איברי התור במחסנית</p>
	7.3	<p>מחסנית ותור</p> <p>העברת ערכי תור למחסנית ולהיפך.</p> <p>היפוך סדר איברים בתור ע"י העברה למחסנית, והכנסה לתור.</p> <p>סריקה עץ בינארי לפי רמות (אלגוריתם וקוד שמומלץ להציג בפרק על עצים בינאריים).</p>	2	2	<p>דגש על פעולות רקורסיה עם תור.</p> <p>הבנת ההבדל כאשר מיקום הקריאה הרקורסיבית משתנה וההשפעה על הריצה.</p> <p>יכולת יצירת מעקב אחר ריצה של הקוד באמצעות טבלת מעקב או עץ קריאות.</p> <p>תרגילים הכוללים מבנה נתונים תור, ומבנה נתונים מחסנית שילובים שנבחרים לפי יעד האלגוריתם</p>

מס' סעיף	תת סעיף	נושא	שעות הרצאה	שעות תרגול	מיומנות נלמדת
8		<p>המחלקה הגנרית Node – מחלקה גנרית ייצוג חוליה בסיסית.</p> <p>דוגמא חוליה מסוג שלם :</p> <p>Node&lt;Integer&gt; p</p> <p>פעולות בסיסיות במחלקה Node :</p> <p>getValue()</p> <p>getNext()</p> <p>setValue(Type x)</p> <p>setNext( Node next)</p> <p>יצירת אוסף לינארי משורשר</p>	2	6	ראה מחלקה בנספח
	8.1	<p>יצירת חוליה ראשונה באוסף</p> <p>הוספת חוליה לאוסף לינארי במיקום נתון ( מתקבלת הפנייה )</p> <p>מחיקת חוליה ממיקום כלשהו( מתקבלת הפניה)</p> <p>סריקת אוסף לינארי</p> <p>שרשור שני אוספים לינארים</p> <p>מניית מספר חוליות באוסף לינארי</p> <p>סיבוכיות סריקה של אוסף לינארי</p>	2	3	<p>הבנת תהליך יצירת אוסף לינארי עם החוליה הראשונה.</p> <p>תרגול ושליטה בכתיבת קוד להוספת חוליה( בתחילת האוסף, בסוף האוסף, בכל מיקום נדרש).</p> <p>כתיבת קוד למחיקת חוליה מאוסף.</p> <p>סריקת אוסף לינארי של חוליות עם תנאי של שוויון לערך null</p>
	8.2	<p>פתרון בעיות עם אוספים לינארים, חישוב סיבוכיות, כתיבת פעולות שמחזירות ערך ולא מחזירות.</p> <p>שימוש בערך ההחזרה בפונקציה המזמנת ( לדוגמא קבלת הפנייה לאוסף שנוצר בפעולה )</p>	1	3	<p>חיפוש ערך באוסף</p> <p>חיפוש איברים משותפים בשני אוספים.</p> <p>היפוך רשימה</p> <p>העברת רשימה למבנה נתונים אחר, חישובי סיבוכיות</p>
9		חוליה בינארית - רשימות מקושרות דו כיווניות, עץ בינארי	7	10	

מס' סעיף	תת סעיף	נושא	שעות הרצאה	שעות תרגול	מיומנות נלמדת
		המחלקה הגנרית BinNode יצירת אוסף לינארי משורשר דו-כיווני	1	1	
	9.1	פעולות על עצם מסוג BinNode יצירת רשימה דו כיוונית סריקת רשימה דו כיוונית פעולות על רשימה דו כיוונית בהנתן הפנייה לחוליה הראשונה	2	3	מימוש אלגוריתם להוספת איבר לרשימה דו כיוונית מימוש אלגוריתם למחיקת איבר מרשימה דו כיוונית, סריקת רשימה דו כיוונית בשני הכיוונים.
	9.2	יצירת עץ בינארי סריקות של עץ בינארי ( תוכית, תחילית, סופית) פעולות על עץ בינארי : סכום מספר צמתים מקסימום ומינימום חיפוש ערך בעץ בינארי ממזין עץ חיפוש בינארי סריקת עץ חיפוש בינארי – סיבוכיות חיפוש בעץ חיפוש בינארי סריקת עץ לפי רמות	4	7	פעולות לחיפוש בעץ בינארי פעולות בוליאניות על עץ בינארי – לבדיקת קיום תכונה פעולות על עץ חיפוש סריקה לפי רמות באמצעות תור עזר
<b>סה"כ שעות קורס</b>			<b>38</b>	<b>52</b>	<b>סה"כ - 90</b>

**מרכיבי הטמעה ושילוב של מיומנויות המאה 21, בתהליכי למידה ותרגול**

תיאור המיומנות המוטמעת	תיאור היישום
למידה עצמאית	מומלץ : למידה מקדמית של נושאים עצמאית על בסיס מקורות שונים : סרטונים, מצגות וכו..
עבודת צוות	ביצוע מטלות סיכום בצוותי תלמידים ( 1-3 ) סטודנטים.

תיאור המיומנות המוטמעת	תיאור היישום
שפה זרה	הפנייה והצגת מקורות מידע באנגלית, שימוש בהרצאות נבחרות באנגלית

**ביבליוגרפיה לסטודנט ( לינקים, ספרים מקצועיים דיגיטלים/מסורתיים )**

מס'	תיאור	אופן הגישה(לינק, פרטי הוצאה)	הערות
1	ספרות מקצועית הזמינה בגירסאות מודפסות ודיגיטליות של כותבים שונים והוצאות לאור שונות		
2	Data Structures and Algorithms Made Easy in Java: Data Structure and Algorithmic Puzzles	Amazon	Narasimha Karumanchi
3	מבנה נתונים c#	ויליאם פרג'ון, ד"ר יבגני קנל	
4	מבנה נתונים java	ויליאם פרג'ון, ד"ר יבגני קנל	

**ביבליוגרפיה למרצים ( לינקים, ספרים מקצועיים דיגיטלים, מאמרים )**

מס'	תיאור	אופן הגישה(לינק, פרטי הוצאה)	הערות
1	Data Structures and Algorithms in Java	Amazon	Robert Lafore
2	Data Structures and Algorithms in Java	Amazon	Michael T. Goodrich

**דוגמאות לידע המצופה בסביבת העבודה של בוגר הקורס (ידע אקטיבי המייצר תוצר)**

מס'	ידע ופעולה
1	קבלת קלט נתונים ויצירת אוסף נתונים לינארי לפי דרישה
2	קבלת אוסף נתונים מטיפוס מסויים והעברתו לאוסף נתונים מטיפוס אחר
3	מיון אוסף נתונים נתון
4	הוספה\מחיקה פריט מידע מאוסף נתונים נתון
5	חיפוש ערך באוסף נתונים נתון
6	חישוב סיבוכיות של פעולה נתונה

מס'	ידע ופעולה
7	חשיבה יצירתית ומתן מענה לבעיה אלגוריתמית בשילוב מבנה נתונים
8	השוואה יעילות בין שתי פעולות המבצעות את אותה מטלה
9	התאמת צורת סריקה של עץ בינארי לבעיה נתונה
10	יישום סריקת עץ בינארי לפי רמות עם תור עזר
11	יצירת אוסף לינארי משורשר דו-כיווני מקלט של אוסף נתונים
12	כתיבת קוד לסריקת אוסף לינארי דו-כיווני משמאל לימין
13	כתיבת קוד לסריקת אוסף לינארי דו-כיווני מימין לשמאל
14	הוספת פריט מידע לאוסף לינארי משורשר דו-כיווני
15	מחיקת פריט מידע לאוסף לינארי משורשר דו-כיווני
16	התמרת אלגוריתם רקורסיבי לבעיה לאלגוריתם איטרטיבי תוך שימוש במחסנית עזר
17	יכולת לבחור מבנה נתונים מתאים לבעיה אלגוריתמית
18	יכולת לממש אוסף נתונים במבני נתונים שונים, בחירה מושכלת של מבנה הנתונים באופן שכלל האלגוריתם יהיה יעיל ככל שניתן
19	חיבור מבני נתונים לידע ומיומנויות ויישום בקורסים נוספים בתכנית הלימודים) בסיסי נתונים, מערכות הפעלה, , תכנות embedded בסביבת שפת C)
20	יצירה של מבנה נתונים מורכב: רשימה של מחסניות, תור של עצים בינאריים וכו..

### נספח – מחלקות לתמיכה בסילבוס הקורס ( כל מרצה מוזמן לשנות, לחדש)

מחלקות אלו הן בגדר המלצה, כל מרצה מוזמן לכתוב מחלקות בסיס לייצוג מבני נתונים שונים, יתרה מזו, מומלץ לבנות ולכתוב מחלקות אלו בשיתוף הסטודנטים

### המחלקה הגנרית Node

מחלקה זו מגדירה חוליה גנרית. באמצעות חוליה Node ניתן ליצור אוספים לינאריים, אוסף משורשר, מחסנית, תור

```
public class Node<T>{
    private T value;
    private Node <T> next;
    public Node (T value){
        this.value=value;
        this.next=null;
    }
    public Node (T value, Node <T> next){
        this.value=value;
```

```

        this.next=next;
    }
    public Node <T> getNext(){
        return this.next;
    }
    public void setNext(Node <T> next){
        this.next=next;
    }
    public T getValue(){
        return this.value;
    }
    public void setValue(T value){
        this.value=value;
    }
    public boolean hasNext(){
        return this.next!=null;
    }
    public String toString(){
        return ""+this.value;
    }
}

```

#### ממשק פעולות המחלקה Node

יעילות	שם הפעולה	תיאור הפעולה
O(1)	Node (T x)	הפעולה בונה חוליה. הערך של החוליה הוא x, ההפניה לחוליה הבאה שווה ל- null
O(1)	Node (T x, Node<T> next)	הפעולה בונה חוליה עם ערך של x ערכו של next יכול להיות null
O(1)	T getValue()	הפעולה מחזירה את הערך של החוליה
O(1)	Node<T> getNext()	הפעולה מחזירה את החוליה העוקבת. אם אין חוליה עוקבת, הפעולה תחזיר null
O(1)	boolean hasNext()	האם יש חוליה נוספת
O(1)	void setValue (T x)	הפעולה משנה את ערך החוליה להיות x

יעילות	שם הפעולה	תיאור הפעולה
O(1)	<code>void setNext (Node&lt;T&gt; next)</code>	הפעולה משנה את ההפניה לחוליה הבאה ל- next
O(n T )	<code>String toString()</code>	הפעולה מחזירה מחרוזת המתארת את החוליה

### המחלקה BinNode

מחלקה זו מגדירה חוליה גנרית עם שתי הפניות. באמצעות חוליה BinNode, ניתן ליצור אוסף לינארי דו כיווני, עץ בינארי.

```
public class BinNode<T>{
    private T value;
    private BinNode <T> left;
    private BinNode <T> right;
    public BinNode (T value){
        this.value=value;
        this.left=null;
        this.right=null;
    }
    public BinNode (BinNode <T> left,T value, BinNode <T> right){
        this.value=value;
        this.left=left;
        this.right=right;
    }
    public BinNode <T> getLeft(){
        return this.left;
    }
    public BinNode <T> getRight(){
        return this.right;
    }
    public void setLeft(BinNode <T> left){
        this.left=left;
    }
    public void setRight(BinNode <T> right){
        this.right=right;
    }
    public T getValue(){
        return this.value;
    }
    public void setValue(T value){
        this.value=value;
    }
    public boolean hasLeft(){
        return this.left!=null;
    }
    public boolean hasRight(){
        return this.right!=null;
    }
    public String toString(){
        return " "+this.value;
    }
}
```

}

ממשק פעולות המחלקה BinNode

ממשק מחלקה Binnode

סיבוכיות	חתימת הפעולה	תיאור הפעולה
O(1)	BinNode (T x)	בונה חוליה בינארית משים: את ערך החוליה להיות - x ואת שתי הפניות שלה ל- null
O(1)	BinNode ( BinNode<T>left, T x, Bin Node<T> right )	בונה חוליה בינארית משים: את ערך החוליה להיות - x, ואת שתי ההפניות שלה לחוליות הבינאריות left ו- right ערכן של כל אחת משתי ההפניות הללו יכול להיות null
O(1)	T getValue()	החזר את ערך החוליה
O(1)	BinNode<T> getLeft()	החזר את החוליה השמאלית, או null
O(1)	BinNode<T> getRight()	החזר את החוליה הימנית, או null
O(1)	boolean hasLeft()	האם יש חוליה משמאל?
O(1)	boolean hasRight()	האם יש חוליה מימין?
O(1)	void setValue(T x)	שנה את ערך החוליה ל-x
O(1)	void setLeft(BinNode<T>left)	השם את ההפניה לחוליה השמאלית ל- left ערכו של left יכול להיות null
O(1)	void setRight(BinNode<T>right)	השם את ההפניה לחוליה הימנית ל- right ערכו של right יכול להיות null

### מחסנית גרית ממומשת באמצעות החוליה Node

```

public class Stack <T>{
private Node <T> first;
public Stack() {
    this.first= null;
}
public boolean isEmpty() {
    return this.first==null;
}
public void push (T x) {
    this.first=new Node <T>(x, this.first);
}
public T pop() {
    T x = this.first.getValue();
    this.first=this.first.getNext();
    return x;
}

```



```

}
public T top() {
    return this.first.getValue();
}
public String toString(){
    String str="[";
    Node <T> pos =this.first;
    while (pos.hasNext()) {
        str=str+pos.getValue()+",";
        pos=pos.getNext();
    }
    str = str +pos.getValue()+"]";
    return str;
}
}

```

מימוש מחסנית גנרית ממומשת באמצעות מערך ( מגבלה על אורך המחסנית)

```

public class Stack<T>{
    public static final int STACK_SIZE = 100;
    private T[] data;
    private int top;
    public Stack (){
        this.top=-1;
    }
    public boolean isEmpty(){
        return this.top== -1;
    }
    public void push (T x) {
        this.top++;
        this.data[top]= x;
    }
    public T pop (){
        T x = this.data[top];
        this.top--;
        return x;
    }
    public String toString(){
        String str="[";
        int k =this.top;
        while (k>=0) {
            str=str+this.data[k]+",";
            k--;
        }
        str = str +"]";
        return str;
    }
}

```

תור גנרי ממומש באמצעות החוליה Node

```
class Queue<T>{
    private Node<T> first;
    private Node<T> last;
    public Queue(){
        this.first = null;
        this.last = null;
    }
    public void insert(T x){
        Node<T> temp = new Node<T>(x);
        if (first == null)
            first = temp;
        else
            last.setNext(temp);
        last = temp;
    }
    public T remove(){
        T x = first.getValue();
        first = first.getNext();
        if (first == null)
            last = null;
        return x;
    }
    public T head(){
        return first.getValue();
    }
    public boolean isEmpty(){
        return first == null;
    }
    public String toString(){
        String s = "[";
        Node<T> p = this.first;
        while (p.hasNext()){
            s = s + p.getValue().toString() + ",";
            p = p.getNext();
        }
        s = s + p.getValue().toString() + "]";
        return s;
    }
}
```

תור גנרי ממומש באמצעות מערך

```
public class Queue <T> {
    public static final int DEFAULT_SIZE=100;
    private T data[];
    private int index;
    public Queue(){
```

```
data= (T[]) new Object [DEFAULT_SIZE];
this.index=0;
}
public boolean isEmpty(){
    return this.index==0;
}
public void insert (T x){
    this.data[index]=x;
    this.index++;
}
    public T remove() {
        T x=this.data[0];
        for(int i=0; i<this.index-1; i++){
            data[i]=data[i+1];
        }
        this.index--;
        return x;
    }
    public T head(){
        return this.data[0];
    }
    public String toString() {
        String s="[";
        for(int i=0; i<this.index; i++)
            if(i<index-1) s=s+data[i]+",";
            else s=s+data[i];
        s=s+"]";
        return s;
    }
}
```

## 203 - בסיסי נתונים sql

### מספר הקורס: 203

### קורסי קדם חובה :

1. רצוי ידע בסיסי בתכנות.

### היקף שעות : 80

**מטרת הקורס :** חשיפה והכרות עם תיאוריה, מבנה ומושגים בסיסיים של בסיס נתונים רלציוניים לצורך טיפול בנתונים רבים. התנסות ותרגול של החומר התאורטי עם שאילתות SQL לצורך הכנסה, אחזור, מחיקה ועדכון נתונים.

### תקציר הקורס ( 3-5 שורות)