

## שרשרת גנרית שרשרת דפייט

1. כתוב את המחלקה הגנרית  $\text{Node}\langle T \rangle$  לפי הממשק המובא להלן:

תיאור הפעולה	חתימת הפעולה
<b>בנאים:</b>	
פעולה הבונה חוליה שבערך $x$ שלה יהיה $x$ וב- $\text{next}$ שלה יהיה $\text{null}$ הערך	$\text{Node} (T x)$
פעולה הבונה חוליה שבערך $x$ שלה יהיה $x$ וב- $\text{next}$ שלה יהיה הערך $\text{next}$ (ערך $\text{next}$ המועבר כפרמטר יכול להיות גם $\text{null}$ )	$\text{Node} (T x, \text{Node}\langle T \rangle \text{next})$
<b>שאליות:</b>	
אם $T$ מחלקה עוטפת לטיפוס בסיסי ( $\text{Integer}, \text{Double}, \text{Character}$ ) יוחזר ערך החוליה, ואם הפניה לעצם, תוחזר הפנייה לעצם זה	$T \text{getValue} ()$
מוחזרת הפנייה לחוליה הבאה	$\text{Node}\langle T \rangle \text{getNext} ()$
פעולה המחזירה אמת אם $\text{next}$ מפנה לחוליה נוספת (כלומר אינו $\text{null}$ ) ושקר אחרת	$\text{boolean hasNext} ()$
פעולה המחזירה מחרוזת המתארת את מצב העצם	$\text{String toString} ()$
<b>פקודות:</b>	
הפעולה משנה (מעדכנת) את ערך החוליה ל- $x$	$\text{void setValue} (T x)$
הפעולה משנה את ערכו של $\text{next}$ להיות $\text{next}$ חדש (ערך $\text{next}$ המתקבל כפרמטר יכול להיות גם $\text{null}$ )	$\text{void setNext} (\text{Node}\langle T \rangle \text{next})$

2. כתוב תכנית היוצרת שתי שרשראות באורך 8 חוליות כל אחת: הראשונה מסוג  $\text{Node}\langle \text{Integer} \rangle$  שתכיל מספרים אקראיים בין 5 ו-25. השנייה מסוג  $\text{Node}\langle \text{String} \rangle$  ותכיל שמות של צבעים (אפשר לבחור בכל משפחת שמות אחרת). הצג את השרשראות.

3. נתונה המחלקה Bucket (גרפי). המחלקה נמצאת בתוך `unit4.jar` כדי להשתמש במחלקה יש לכתוב משפט `import`:

```
import unit4.bucketLib.*;
```

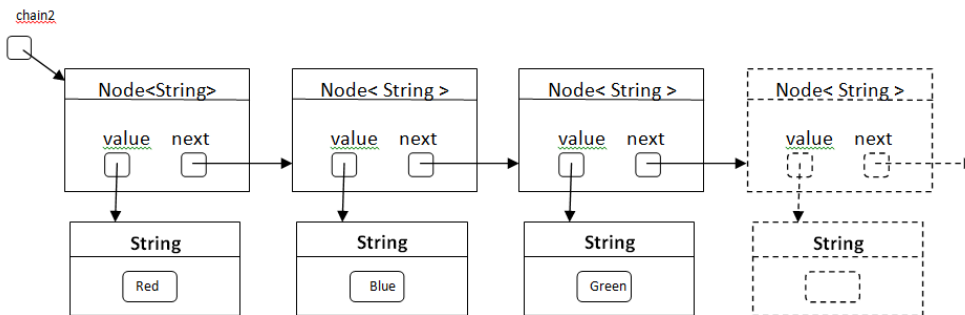
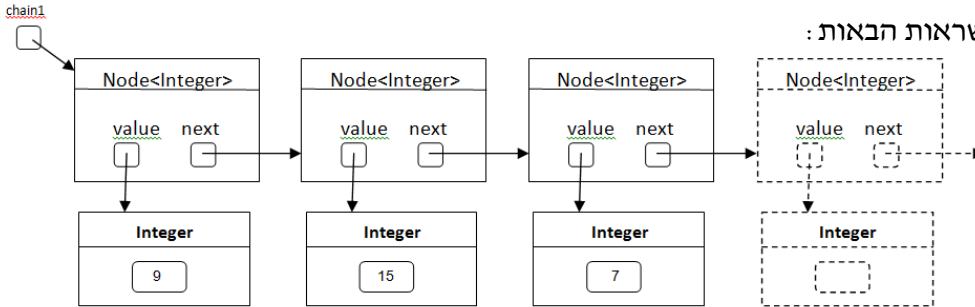
### הנחיות:

כתוב את המילים: `import unit4.` (נקודה אחרי `unit4`) אם לא נפתחת תיבת השלמה אוטומטית, (גם לא אחרי הקשה על צירוף המקשים <רווח> + <ctrl>) יש להתקין את החבילה החיצונית `unit4.jar`. ראה הנחיות התקנה בדף הבא: <http://blog.csit.org.il/UpLoad/FilesUpload/unit4ForEclipse.pdf> (דף ראשי של יסודות בבלוג, התקנת סביבת העבודה לשפת `java`, מדריך לשימוש בספריית העזר `unit4`, ו- שילוב קובץ `jar` באקליפס). תרגיל דוגמה לשימוש במחלקה `Bucket` ראה בדף עבודה 1 בפרק 1 בספר יסודות (הורד גם את הקובץ `Chap1` פרוש ופתח באקליפס).

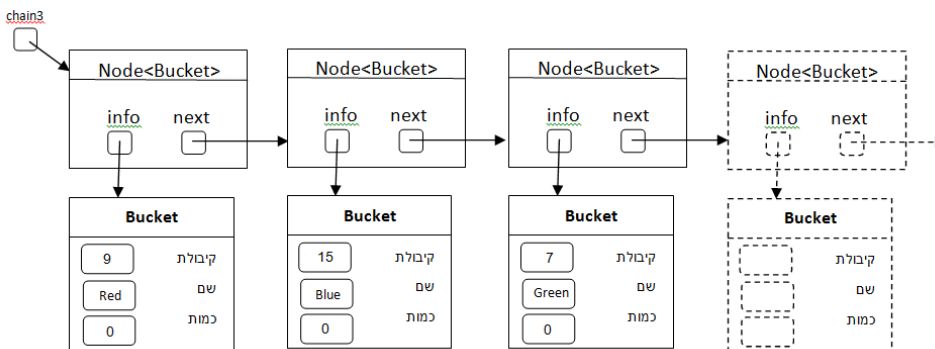
למחלקה Bucket יש פעולה בונה המקבלת כפרמטר מספר המציין את קיבולת הדלי ומחרוזת המציינת את שם הדלי. דלי שנוצר בפעולה new יופיע במסך הגרפי.

כתוב פעולה בשם getChain שתקבל כפרמטר שתי שרשראות של מספרים ושל מחרוזות. הפעולה תיצור ותחזיר שרשרת של דליים Chain<Bucket> שכל חוליה בה תכיל הפנייה לדלי שהקיבולת שלו רשומה בחוליה במקום המתאים בשרשרת המספרים ושמו רשום בשרשרת המחרוזות באותו מקום.

לדוגמה: עבור השרשראות הבאות:



תקבל שרשרת הדליים הבאה:



4. כתוב הוראה בתכנית שתיצור שרשרת של דליים באמצעות הפעולה שכתבת בסעיף 3.



5. הרץ את התכנית. התוצאה: יפתח חלון גרפי דוגמת החלון שלהלן, שבו מצוירים הדליים שבשרשרת.

להלן ממשק חלקי של המחלקה Bucket :

שם הפעולה	תיאור הפעולה
Bucket (capacity, "name")	פעולה בונה היוצרת דלי ריק בעל קיבולת capacity (מספר שלם) ושם הרשום בין סימני הגרשיים.
void empty ()	פעולה המרוקנת את הדלי הנוכחי.
void fill (x)	פעולה הממלאת את הדלי הנוכחי בכמות מים השווה ל- x ליטרים. אם כמות המים היא מעבר לקיבולת הדלי, הדלי יתמלא ויתרת המים תשפך החוצה. הנחה: x הוא מספר לא שלילי, לא בהכרח שלם.
int getCapacity()	
double getCurrentAmount ()	פעולה המחזירה מספר השווה לכמות המים שבדלי הנוכחי.
void pourInto (otherBucket)	פעולה המעבירה את כמות המים המרבית האפשרית מהדלי הנוכחי לדלי ששמו otherBucket.

6. כתוב פעולה המקבלת את שרשרת הדליים וממלאה כל דלי בכמות מים אקראית שבין 0 וקיבולת הדלי: האלגוריתם:

**מילוי-הדליים** (שרשרת דליים chain)

(1) הצב הפנייה על החוליה הראשונה בשרשרת  $pos \leftarrow$

(2) כל עוד  $(pos \neq null)$

// Bucket b  $\leftarrow pos.getValue()$  (2.1) b הינו הפנייה מסוג דלי

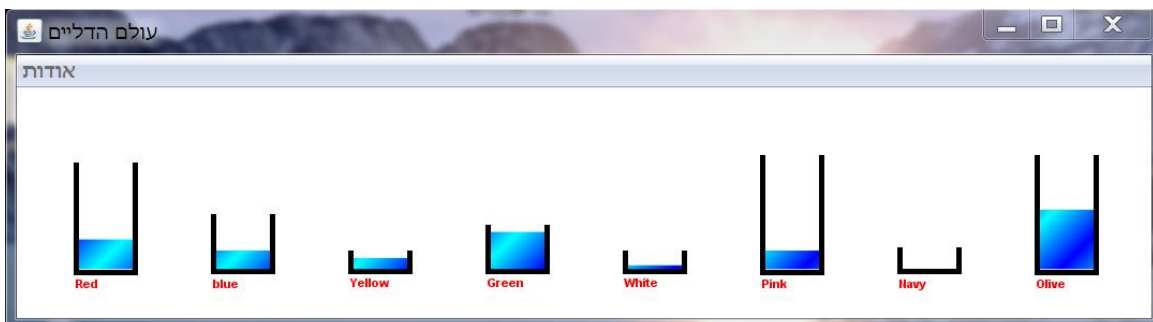
(2.2) הגרל מספר שלם בין 0 וקיבולת הדלי  $x \leftarrow$

(3.3) מלא את הדלי בכמות של x ליטרים

(3.4) עבור לדלי הבא

זמן את הפעולה מהתכנית, הרץ וצפה בדליים המתמלאים.

7. כתוב פעולה המקבלת כפרמטר את שרשרת הדליים, מחשבת ומחזירה את סך כמות המים שבכל הדליים. זמן את הפעולה מהתכנית והדפס את הכמות שהתקבלה.



צבוקה נעימה