

## המחלקה קלף



1. כתוב מחלקה בשם **קלף Card** שתגדיר קלף שתכונותיו:

- ערך value : מספר שלם בין 2 ו-14
- הקלפים 2 - 10 ואחר כך נסיך, מלכה, מלך ואס:
- נסיך J יוגדר כ- 11
- מלכה Q תוגדר כ- 12
- מלך K יוגדר כ- 13
- אס A יוגדר כ- 14.
- צורה shape : מחרוזת שערכיה הם:
- עלה Spade (♠)
- לב Heart (♥)
- יהלום Diamond (♦)
- תלתן Club (♣)

פעולות המחלקה: בנאים, פעולות אחזור (Get) כל תכונה, הפעולה ToString (שים לה שלא הגדרנו פעולות עדכון לצורות, כי הקלף לא יכול לשנות את ערכיו).

2. כתוב תכנית שתשחק את המשחק הבא:

א. כתוב פעולה בשם GetCard() שתקלוט את פרטי הקלף ותחזיר עצם מסוג קלף.

ב. כתוב פעולה בשם stronger שתקבל כפרמטר שני קלפים ותחזיר את הקלף ה"מנצח". קלף מנצח (חזק) הוא קלף שערכו גדול יותר. אם לשני הקלפים אותו ערך, ינצח הקלף בעל הצורה החזקה יותר. חוזק הצורות: Spade (♠) < Heart (♥) < Diamond (♦) < Club (♣). (שים לב לסדר המיון לפי הא"ב של חוזק הצורות).

ג. כתוב תכנית שתקלוט את פרטיהם של שני קלפים, תציג אותם ואחר כך תציג את הקלף המנצח בצירוף הודעה מתאימה. חובה להשתמש בפעולות שכתבת בסעיפים א' ו- ב' ובמחלקה Card.

3. כתוב תכנית שתגדיר מערך בגודל 10 של קלפים.

על התכנית למלא את המערך ב-10 קלפים שונים, ולהציג את הקלפים בשורה. לאחר מכן יש להציג בשורה הבאה את הקלף החזק ביותר במערך.

הנחיות:

א. הוסף למחלקה Card פעולה בוליאנית בשם equals שתקבל קלף ותחזיר אמת אם הקלף זהה בערך ובצורה לקלף הנוכחי ושקר אחרת.

ב. ממש בתכנית פעולה בשם **קיים? Exist** שתקבל את מערך הקלפים וקלף נוסף ותחזיר אמת אם הקלף קיים בחפיסה ושקר אחרת. עבור כל קלף חדש שמתווסף למערך, יש לוודא שהקלף לא קיים במערך.

4. חשוב והצע דרך למלא את החפיסה בקלפים אקראיים (כלומר - להגריל ערך וצורה וליצור קלף שיוכנס לחפיסה רק אם הוא לא קיים בה). ממש את הפתרון שהצעת.

ממשק המחלקה קלף:

| כותרת (חתימת) הפעולה                          | תיאור הפעולה  |
|---|---|
| Card ( <b>int</b> value, <b>string</b> shape) | פעולה בונה היוצרת קלף שערכו value וצורתו shape  |
| <b>int</b> GetValue ()                        | פעולה המחזירה את ערך הקלף   |
| <b>string</b> GetShape()                      | פעולה המחזירה את צורת הקלף  |
| <b>string</b> ToString()                      | פעולה המחזירה מחרוזת המתארת את מצב הקלף באופן הבא:<br>ערך מקף צורה.<br>למשל: 3 יהלום יוצג כ- 3-D ונסיד תלתן יוצג כ- J-C |
| <b>bool</b> Equals (Card c)                   | פעולה המקבלת כפרמטר קלף ומחזירה אמת אם הקלף שהתקבל זהה לקלף הנוכחי, ושקר אחרת.  |



עבודה נעימה

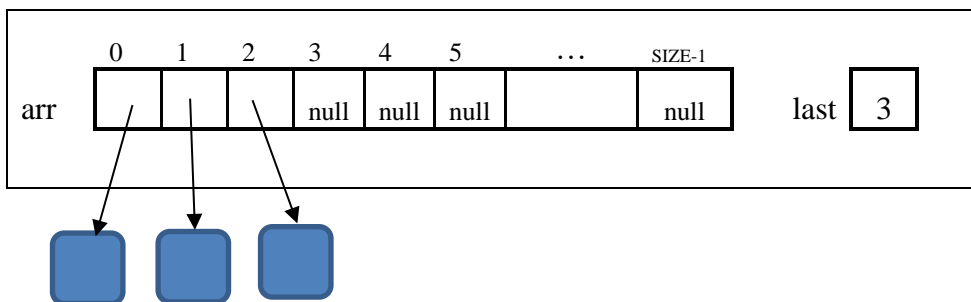
## אוסף המוגדר באמצעות מערך

ראה פרק 6 בספר עיצוב תכנה הפניית ועצמים מורכבים, סעיף ו' עצם מורכב המגדיר עצם (עמ' 161 בספר) "רשימה שמית כיתתית".

אחת הדרכים להגדיר אוסף של עצמים היא באמצעות מערך.

### תכונות האוסף:

- מערך של עצמים: מערך שכל איבר בו יהיה מטיפוס הפנייה לעצם.
- קבוע מספרי שלם הקובע את גודל המערך. `public static const int SIZE = 100;` הקבוע מוגדר כתכונה של המחלקה (static) ולא של העצם, והרשאת הגישה שלו היא בדרך כלל ציבורית. משמעות ה- static: גם אם נגדיר הרבה אוספים מטיפוס המחלקה, עדיין יהיה קבוע אחד ויחיד משותף לכולם. משמעות ה- public: הקבוע מוכר בכל הפרויקט וניתן לגשת אליו מכל מחלקה (ע"י ציון שם המחלקה שבה הוא מוגדר).
- העצמים מוחזקים במערך כשהם מצופפים ולכן נגדיר מספר שלם last שתפקידו כפול: מצד אחד הוא מציין את מספר העצמים הקיימים בפועל בחפיסה ומצד שני את מקומו של התא הפנוי הראשון במערך. כך, במערך שבו 3 עצמים יהיו בתאים 0, 1 ו- 2 הפניות לעצמים וערכו של last יהיה 3.
- תכונות נוספות במידת הצורך.



### הפעולות באוסף

- פעולה בונה:** יוצרת מערך בגודל הנתון, ומאתחלת את מספר העצמים בו ל-0.
- האם ריק?** פעולה בוליאנית המחזירה אמת אם אין עצמים באוסף ושקר אחרת.
- הוספה:** הפעולה מקבלת כפרמטר עצם ומוסיפה אותו למערך. הפעולה מעדכנת את מספר העצמים במערך.
- איתור:** פעולה המתארת עצם לפי אחת מתכונותיו ומחזירה הפנייה אליו. אם העצם לא קיים יוחזר null.
- מחיקה:** פעולה המתארת עצם לפי אחת מתכונותיו, מוחקת אותו מהמערך ומחזירה הפנייה אליו. הפעולה מעדכנת את ערכו של last. בסיום הפעולה יישאר המערך מצופף.
- ToString:** הפעולה מחזירה מחרוזת המתארת את מצב העצמים במערך.

פעולות נוספות במידת הצורך

## חפיסת קלפים

חפיסת קלפים Deck הינה מחלקה המגדירה אוסף של קלפים.

למחלקה 3 תכונות:

- מערך של קלפים.
- גודל המערך SIZE שערכו 52
- מספר הקלפים הקיימים בפועל בחפיסה

`private Card [] arr`

`public static const int SIZE = 52`

`private int last`

**שים לב!** אתחול המערך וערך last ייעשה בפעולה הבונה (ולא בשלב הגדרת התכונות)

1. ממש את המחלקה חפיסת קלפים Deck לפי הממשק שלהלן:

| הפעולה                      | תיאור הפעולה   |
|-----------------------------|--|
| Deck()                      | פעולה בונה המחזירה חפיסת קלפים ריקה  |
| bool IsEmpty()              | פעולה המחזירה אמת אם החפיסה ריקה ושקר אחרת.  |
| void Add (Card c)           | פעולה המוסיפה את הקלף c לחפיסת הקלפים  |
| Card GetCard()              | פעולה המוציאה ומחזירה את הקלף שבראש החפיסה.<br>הנחה: החפיסה לא ריקה.                                 |
| Card GetCard (int value)    | פעולה המוציאה ומחזירה את הקלף הראשון שערכו value מהחפיסה.<br>אם אין קלף שערכו value יוחזר הערך null  |
| Card GetCard (string shape) | פעולה המוציאה ומחזירה את הקלף הראשון שצבעו color מהחפיסה.<br>אם אין קלף שצורתו shape יוחזר הערך null |
| string ToString()           | פעולה המחזירה מחרוזת המתארת את הקלפים שבחפיסה  |

2. כתוב תכנית שתיצור חפיסת קלפים (אוסף) ותמלא את החפיסה ב- 10 - 15 קלפים שונים.

על התכנית לנסות ליצור 2 סדרות קלפים, האחת: 3-4 קלפים בעלי אותו ערך

והשנייה: לפחות 3 קלפים היוצרים רצף עולה מאותה צורה.

א. הצג את הקלפים בחפיסה.

ב. הצג את הסדרות שיצרת. אם לא ניתן ליצור את אחת הסדרות, יש להציג הודעה מתאימה.



צבוקה נעימה

