

שים לב: לשאלה זו שני נוסחים: אחד ב־ Java (עמודים 12-13), ואחד ב־ C# (עמודים 14-15). עבוד על פי השפה שלמדת.

לפותרים ב־ Java

המשחק 'שחק נא' הוא משחק לשחקן אחד בשני שלבים. השחקן מקבל 52 קלפים מעורבבים, שונים זה מזה. על כל קלף יש מספר בין 1 ל־13, וציור של אחת מארבע צורות. הצורות מיוצגות על ידי המספרים 1 עד 4.

בשלב הראשון השחקן מחלק את הקלפים לארבע ערמות על פי הצורה של הקלף, כך שבכל ערמה יש קלפים עם אותה צורה. הקלפים מונחים זה על גבי זה.

בשלב השני השחקן מגריל מספר בין 1 ל־4 המייצג צורה של קלף. השחקן פונה לערמה שעל כל הקלפים בה נמצאת הצורה שמיוצגת על ידי המספר שהוגרל. הוא מרים את הקלף שבראש הערמה, ומעביר אותו לערמה חמישית.

שלב זה יתבצע עד שהמשחק יסתיים.

המשחק מסתיים כאשר מוגרל מספר המייצג ערמה שאין בה קלפים.

ניצחון הוא מצב שבו סכום המספרים שעל הקלפים בערמה החמישית מתחלק ב־100 ללא שארית.

לאחר שהמשחק מסתיים בודקים אם יש ניצחון.

לצורך מימוש המשחק הוגדרו המחלקות: **Card** המייצגת קלף אחד, **Deck** המייצגת את חמש הערמות הנדרשות במשחק, ו־ **Test** המנהלת את המשחק.

כמו כן נכתבה במחלקה **Test** הפעולה `public static boolean game(Card[] cards)`, המקבלת מערך של 52 קלפים שונים המסודרים באופן אקראי. הפעולה מנהלת את המשחק עד סיומו. הפעולה מחזירה `true` אם המשחק הסתיים בניצחון, ואחרת – `false`.

לפניך המחלקה **Card** המייצגת קלף שעליו מספר (value) בין 1 ל־13 וצורה (shape) בין 1 ל־4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }

    public int getValue () { return this.value; }
    public int getShape () { return this.shape; }
}
```

א. i כתוב ב־ Java את כותרת המחלקה **Deck** ואת התכונות שלה.

ii כתוב ב־ Java במחלקה **Deck** את כותרות הפעולות האלה:

— פעולה בונה המגדירה את חמש הערמות להיות ריקות.

— פעולה insert המקבלת קלף ומוסיפה אותו לערמה הנכונה על פי השלב הראשון במשחק.

— פעולה move המגרילה מספר המייצג צורה. אם ערמת הקלפים שצורתה

הוגרלה היא ריקה, הפעולה תחזיר false. אחרת — הפעולה תעביר לערמה

החמישית את הקלף שבראש הערמה שמספרה הוגרל ותחזיר true.

— פעולה sum המחזירה את הסכום הנוכחי של המספרים שעל הקלפים

בערמה החמישית.

שים לב: בתת־סעיף זה אין צורך לממש את הפעולות.

ב. ממש ב־ Java את הפעולה הבונה במחלקה **Deck**.

ג. ממש ב־ Java את הפעולה sum במחלקה **Deck**.

ד. ממש ב־ Java את הפעולה game במחלקה **Test**.

אתה יכול להשתמש בפעולות insert ו־ move של המחלקה Deck, בלי לממש אותן. אם

אתה משתמש בפעולות נוספות, עליך לממש אותן.

המשחק 'שחק נא' הוא משחק לשחקן אחד בשני שלבים. השחקן מקבל 52 קלפים מעורבבים, שונים זה מזה. על כל קלף יש מספר בין 1 ל- 13, וציוור של אחת מארבע צורות. הצורות מיוצגות על ידי המספרים 1 עד 4.

בשלב הראשון השחקן מחלק את הקלפים לארבע ערמות על פי הצורה של הקלף, כך שבכל ערמה יש קלפים עם אותה צורה. הקלפים מונחים זה על גבי זה.

בשלב השני השחקן מגריל מספר בין 1 ל- 4 המייצג צורה של קלף. השחקן פונה לערמה שעל כל הקלפים בה נמצאת הצורה שמיוצגת על ידי המספר שהוגרל. הוא מרים את הקלף שבראש הערמה, ומעביר אותו לערמה חמישית.

שלב זה יתבצע עד שהמשחק יסתיים.

המשחק מסתיים כאשר מוגרל מספר המייצג ערמה שאין בה קלפים.

ניצחון הוא מצב שבו סכום המספרים שעל הקלפים בערמה החמישית מתחלק ב- 100 ללא שארית.

לאחר שהמשחק מסתיים בודקים אם יש ניצחון.

לצורך מימוש המשחק הוגדרו המחלקות: **Card** המייצגת קלף אחד, **Deck** המייצגת את חמש הערמות הנדרשות במשחק, ו- **Test** המנהלת את המשחק.

כמו כן נכתבה במחלקה **Test** הפעולה `Game(Card[] cards)`, המקבלת מערך של 52 קלפים שונים המסודרים באופן אקראי. הפעולה מנהלת את המשחק עד סיומו.

הפעולה מחזירה `true` אם המשחק הסתיים בניצחון, ואחרת – `false`.

לפניך המחלקה **Card** המייצגת קלף שעליו מספר (value) בין 1 ל- 13 וצורה (shape) בין 1 ל- 4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }

    public int GetValue () { return this.value; }
    public int GetShape () { return this.shape; }
}
```

- א. i כתוב ב־ C# את כותרת המחלקה **Deck** ואת התכונות שלה.
- ii כתוב ב־ C# במחלקה **Deck** את כותרות הפעולות האלה:
- פעולה בונה המגדירה את חמש הערמות להיות ריקות.
 - פעולה Insert המקבלת קלף ומוסיפה אותו לערמה הנכונה על פי השלב הראשון במשחק.
 - פעולה Move המגרילה מספר המייצג צורה. אם ערמת הקלפים שצורתה הוגרלה היא ריקה, הפעולה תחזיר false. אחרת — הפעולה תעביר לערמה החמישית את הקלף שבראש הערמה שמספרה הוגרל ותחזיר true.
 - פעולה Sum המחזירה את הסכום הנוכחי של המספרים שעל הקלפים בערמה החמישית.
- שים לב: בתת־סעיף זה אין צורך לממש את הפעולות.
- ב. ממש ב־ C# את הפעולה הבונה במחלקה **Deck**.
- ג. ממש ב־ C# את הפעולה Sum במחלקה **Deck**.
- ד. ממש ב־ C# את הפעולה Game במחלקה **Test**.
- אתה יכול להשתמש בפעולות Insert ו־ Move של המחלקה Deck בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.