

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

### לפתרונות 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. הוצאות מיווצרות על ידי המספרים 1 עד 4.

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

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

שלב זה יבוצע עד שהמשחק יסתתיים.

**המשחק מסתיים** כאשר מוגריל מספר המייצג ערמה שאין בה קלפים.

**ניצחון** הוא מצב שבו סכום המספרים שעלה 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 הערכות הנדרשות במשחק, ו- **Test** המנהלת את המשחק.

כמו כן נכתבת במחלקה **Test** הפעולה `public static boolean game(Card[] cards)`, המתקבלת מערך של 52 קלפים שונים המסודרים באופן אקראי. הפעולה מנהלת את המשחק עד סיוםו.

הפעולה מחזירה `true` אם המשחק הסתיים בניצחון, ואחרת — מחזירה `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. כתוב ב- Java את כותרת המחלקה **Deck** ואת התכונות שלה.
- ב. כתוב ב- Java במחלקה **Deck** את כותרות הפעולות האלה:
- פעולה בונה המגדירה את חמיש הערמות להיות ריקות.
  - פעולה insert המקבלת קלף ומוסיפה אותו לערמה הנוכחי על פי השלב הראשון במשחק.
  - פעולה move המגירה מספר המיצג צורה. אם ערמת הקלפים שצורתה הוגלה היא ריקה, הפעולה תחזיר false. אחרת — הפעולה תעביר לערמה החמישית את הקלף שבראש הערמה שמספרה הוגרל ותחזיר true.
  - פעולה sum המחזיר את הסכום הנוכחי של המספרים שעל הקלפים בערמה החמישית.
- שים לב: בתת-סעיף זה אין צורך למש 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 נוספות, עליך למש אותן. אם אתה משתמש בפעולות נוספות, עליך למש 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, וציור של אחת מארבע צורות. הוצאות מיוצגות על ידי המספרים 1 עד 4.

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

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

שלב זה יבוצע עד שהשחקן יסתום.

**המשחק מסתiem** כאשר מוגרל מספר המייצג ערמה שאין בה קלפים.

**ニיחון** הוא מצב שבו סכום המספרים שעלה הקלפים בערמה החמישית מתחלק ב- 100 ללא שארית.

לאחר שהשחקן מסתiem בודקים אם יש ניחון.

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

כמו כן נכתבת במחלקה **Test** הפעולה public static bool Game(Card[] cards) , המתקבלת מערך של 52 קלפים שונים המסודרים באופן אקראי. הפעולה מנהלת את המשחק עד סיוםו. הפעולהמחזירה true אם המשחק הסתיים בניצחון, ואחתה — מחזירה .false

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