

2. נתונה המחלקה **Cube** - קובייה, ולה שתי תכונות:

- size – גודל הקובייה בסמ"ר, מטיפוס מספר שלם.
 - type – סוג הקובייה, מטיפוס תו.
- הניחו שיש פעולות get/Get לתכונות המחלקה.

נתונה המחלקה **Box** - קופסה, ולה שתי תכונות:

- cubes - מערך בגודל 3 מטיפוס **Cube**, בו שומרים את הקוביות. ניתן להכניס למערך קוביות או שכולן מאותו סוג או שכולן מאותו גודל.
 - count – כמות הקוביות במערך.
- לפניכם פעולות המחלקה **Box**. ניתן להשתמש בפעולות ללא צורך במימוש:

הפעולה	הסבר
<pre>public Box(){ cubes= new Cube[3]; count = 0; }</pre>	פעולה בונה.
<pre>Java: public boolean add(Cube c) C#: public bool Add(Cube c)</pre>	הפעולה מקבלת עצם מסוג Cube . אם ניתן להכניס את העצם למערך (בהתאם לדרישות לעיל), הפעולה מכניסה אותה למקום הראשון שפנוי במערך (ומעדכנת את ערך המשתנה count) ומחזירה true . אחרת (אם המערך מלא או שהעצם אינו עונה על הדרישות לעיל), הפעולה מחזירה false ללא שינוי ערכי המערך.

א. קופסה נקראת "מושלמת" אם יש בה 3 קוביות מאותו הסוג וגם מאותו הגודל. ממשו את הפעולה הפנימית שלפניכם במחלקה **Box**:

```
Java: public boolean isPerfect ()
```

```
C#: public bool IsPerfect ()
```

הפעולה מחזירה **true** אם הקופסה מושלמת. אחרת, מחזירה **false**.

ב. במפעל צעצועים אורזים קוביות לקופסאות לפי סדר ההגעה שלהן.

(1) ממשו את הפעולה החיצונית שלפניכם:

```
Java: public static Box packBox (Queue<Cube> q)
```

```
C#: public static Box PackBox (Queue<Cube> q)
```

הפעולה מקבלת תור של קוביות ומחזירה עצם מסוג **Box**. הפעולה תכניס קוביות מהתור (לפי סדר הופעתן) לקופסה, כל עוד פעולת ההכנסה אפשרית. אם לא ניתן להכניס את הקובייה הבאה בתור, הפעולה תעצור ותחזיר את הקופסה שנוצרה.

הערה: בסיום הפעולה סדר הקוביות בתור המקורי נשמר, ללא הקוביות שהוכנסו לקופסה.

(שימו לב: המשך השאלה בעמוד הבא.)

