

3.

בנתיבי תחבורת ציבורית הוצבו מצלמות תנוצה, המצלמות כל כלי רכב שעובר על פניהן.

נתונה המחלקה **CarInfo** – מידע על כלי רכב שצולם, ולה שלוש תכונות:

- `id` – מספר לוחית זיהוי של כלי הרכב שצולם, מטיפוס מחרוזת.
- `privateCar` – אם כלי הרכב שצולם הוא פרטי, התכונה היא `true`, ואם כלי הרכב הוא ציבורי – `false`.
- `speed` – מהירות הנסיעה של כלי הרכב שצולם, מטיפוס שלם.

הנحوו שקיימות פעולות `get`/`Set` ו- `get`/`Set` לכל אחת מן התכונות במחלקה.

לכל רכב שנושא בנתיב תחבורה ציבורית תירשם עבירת תנוצה אם יתקיים פחות אחד מן המצביעים الآלה:

– כלי הרכב הוא פרטי.

– כלי הרכב נוסע מעל מהירות המותרת.

a. כתבו פעולה פנימית במחלקה **CarInfo** בשם `illegal` בשפת Java או `Illegal` בשפת C#, המכבלת את מהירות

הנסיעה המותרת – `maxSpeed` מטיפוס שלם.

הפעולה תחזיר `true` אם כלי הרכב עבר עבירת תנוצה (כלומר, אם כלי הרכב הוא פרטי ו/או נסע מעל מהירות המותרת),

אחרת הפעולה תחזיר `false`.

נתונה המחלקה **CameraInfo** – מידע של מצלמת תנוצה, ולה שלוש תכונות:

• `city` – קוד העיר שבה המצלמה מוצבת, מטיפוס שלם. הקוד הוא מספר בין 0 ל- 99 (כולל).

יש 100 ערים, וכל עיר יש קוד יחודי.

• `maxSpeed` – מהירות המותרת באזור המצלמה, מטיפוס שלם.

• `cars` – מערך מטיפוס **CarInfo** של כל הרכב שצולמו במכשיר (המערך ללא ערכי `null`).

הנحوו שקיימות פעולות `get`/`Set` ו- `get`/`Set` לכל אחת מן התכונות במחלקה.

b. (1) כתבו פעולה פנימית במחלקה **CameraInfo** בשם `allGood` בשפת Java או `AllGood` בשפת C#, המכבלת את מהירות

המחזירה `true` אם כל כלי הרכב שצולמו במכשיר לא עברו עבירת תנוצה, אחרת היא מחזירה `false`.

אפשר להשתמש בפעולת `scattered` שכתבתם בסעיף א.

(2) כתבו פעולה חיצונית בשם `legalCities` בשפת Java או `LegalCities` בשפת C#.

הפעולה מקבלת מערך מטיפוס **CameraInfo** מצלמות שהוצעו ב- 100 הערים השונות.

הפעולה תחזיר את כמות הערים שלא אוטרה בהן שום עבירת תנוצה.

הערה: תיתכן יותר ממכשיר אחד באותה העיר.

אפשר להשתמש בפעולת `scattered` שכתבתם בסעיף ב(1).

תור, מחסנית, בלי למשוך וחוליה, אך ביןרי משתמשים בפעולות
 נוספת, יש למשוך אותו.