

interface

נניח ואנו מעוניינים ליצור אפליקציה, שמזהה תזוזות ימינה ושמאלה של עצמים בכביש.

הפעולות שהעצמים יכולו **בהכרח** יהיו:

- תזוזה ימינה
- תזוזה שמאלה

דוגמאות לעצמים אפשריים:

- רכב
- אופנוע
- ילד
- מבוגר
- אופניים
- כדור

מה המשותף בין כל העצמים?

- אין מהות משותפת, אלא רק פעולה שכל העצמים יודעים לבצע.
- רכב, אופנוע, אופניים – מסוג: כלי תחבורה
- ילד, מבוגר – מסוג: בנאדם
- כדור – מסוג: משחק

לשם כך נועד `interface`, שתפקידו להגדיר "ממשק" ו"תצורה" עבור מחלקות שונות – שאין בהכרח קשר מהותי ביניהם – אלא כולם ממשים את הפעולות המוגדרות בתוך הממשק.

זו הסיבה ש"ממשק" מכונה בעולם התכנות בשם "חוזה" – כי כל מחלקה שמממשת את הממשק, מתחייבת לתת מימוש לכל הפונקציות המוגדרות בממשק.

ממשק – לא יכול להכיל מימוש של פונקציות, אלא רק הגדרה של פונקציות שימושו במחלקות המממשות.

ההבדלים בין מחלקה אבסטרקטית לממשק

ממשק	מחלקה אבסטרקטית	מילת ההגדרה
<code>interface</code>	<code>abstract class</code>	
יתחיל ב"ו" ואחריה נשרשר את המילה (כאשר גם היא תתחיל באות גדולה) לדוגמה: <code>IMove</code>	יתחיל באות גדולה לדוגמה: <code>Car</code>	מוסכמה לשם שניתן
ממשק הוא הגדרת תצורה של פעולות, כל מחלקה שמכילה בתוכה את כל הפעולות המוגדרות בממשק – תממש את הממשק.	מחלקה ראשונית – המגדירה ישות מסוימת (אבסטרקטית) – שממנה יירשו כל הישויות שהם גם הישות האבסטרקטיות	מהות
ממשק: מגדיר פעולות תזוזה ימינה ותזוזה שמאלה מחלקות מממשות: חילל בשחמט, בנאדם, כדור, חתול, רכב	מחלקה אבסטרקטית: כלי רכב מחלקות יורשות: רכב, אופנוע, משאית	דוגמה
אסור	מותר	הוספת משתנים עם ערך התחלתי
אסור	מותר	הוספת פונקציה עם תוכן פונקציה
אסור – בצורה דיפולטיבית הכל יהיה <code>public</code>	<ul style="list-style-type: none"> מותר מאפיין אבסטרקטי – לא יכול להיות <code>private</code> 	שימוש ב"הרשאות גישה"
אסור	אסור	האם אפשר ליצור מופע
מותר – המשתנה יוכל להצביע לכל מחלקה שמממשת את הממשק	מותר – המשתנה יוכל להצביע לאובייקטים ממחלקות שירשו מהמחלקה האבסטרקטית	האם אפשר ליצור משתנה מהטיפוס הנתון
מותר למחלקה אחת לממש כמה ממשקים	אסור למחלקה אחת לרשת כמה מחלקות	האם אפשר לבצע הורשה מרובה / מימוש מרובה