מבוא לתיכנות מונחה עצמים – ממשקים

ממשק לעומת מחלקה

מחלקה
מצב
התנהגות עם מימוש

ממשק
התנהגות *בלי מימוש*

ממשק – למה צריך את זה?

– כדי לכתוב אלגוריתם *פעם אחת*,
ולהריץ אותו על *הרבה מחלקות שונות*.

ממשק הוא חוזה בין מחלקה לאלגוריתם:

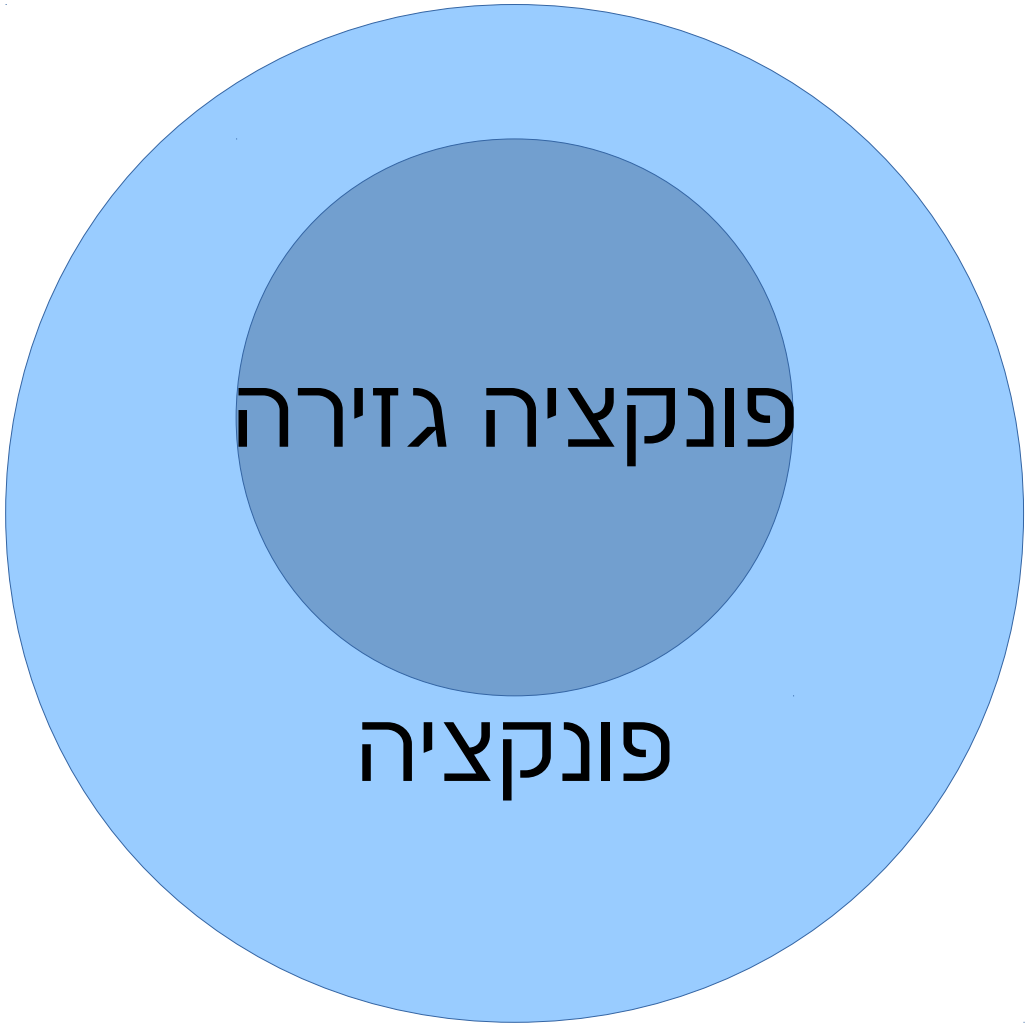
המחלקה מתחייבת:	**האלגוריתם מתחייב:**
לממש את המתודות-בלי-מימוש של הממשק.	לרוץ נכון על העצמים של המחלקה.

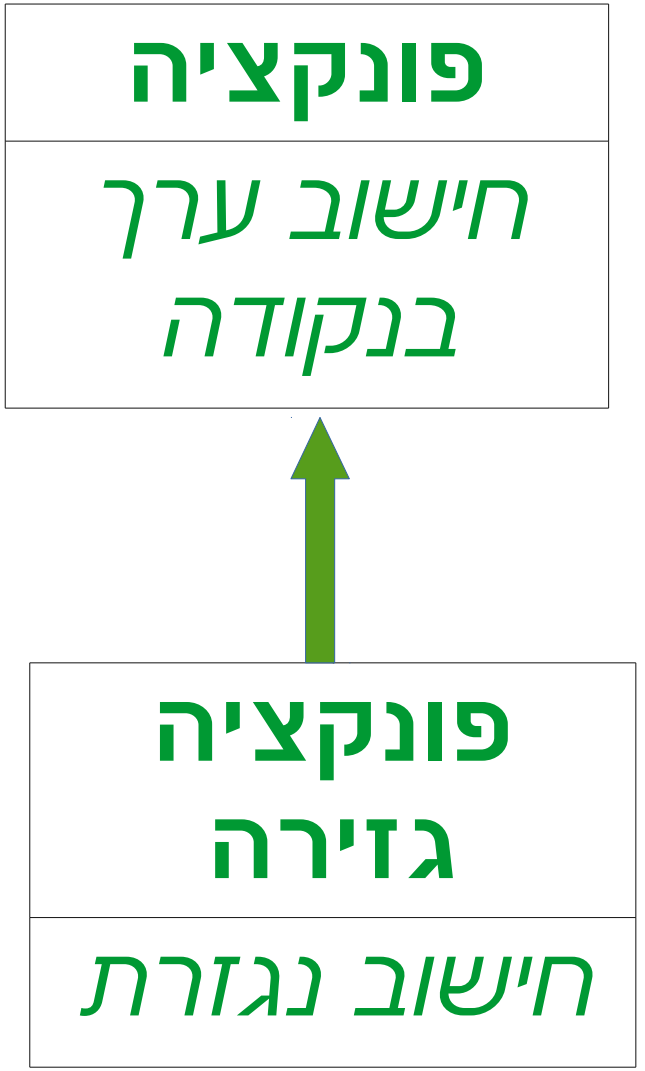
דוגמאות

- א. שרטוט גרף של פונקציה כללית – הממשק Function
- ב. חזרה על פעולה כמה פעמים – הממשק Runnable
- ג. מסננים – הממשק Predicate
  שימוש – סינון במטלה 0.
- ד. משווים – הממשק Comparator
  שימוש – סידור מערכים.
- לעוד ממשקים פונקציונליים (עם מתודה אחת) חפשו:
  Java util function
- **חידה**: אתם רוצים לכתוב פונקציה החוזרת על פעולה עבור כל השלמים בין 0 ל-*n*. *באיזה ממשק תשתמשו?*

הרחבת ממשקים

interface DifferentiableFunction **extends** Function { ... }





הכללת ממשקים

```
interface Predicate {
    bool test(String item);
}
Predicate p = x → x.charAt[0]=='a';
```

```
interface Predicate<T> {
    bool test(T item);
}
Predicate<String> p1 = x → x.charAt[0]=='a';
Predicate<Integer> p2 = x → x>5;
```