

דף עבודה – פעולות (Procedures) ו-Notifier

שם התלמיד/ה: _____

תאריך: _____

בכל תרגיל:

1. צרו פרויקט חדש בשם המבוקש.
2. תכננו את המסך ב-**Designer**.
3. כתבו את הלוגיקה ב-**Blocks**.
4. הריצו ובדקו שהכול עובד.

תרגיל 1 – פעולה שמחליפה טקסט

שם הפרויקט: **ProcHello**

1. במסך **Designer**:
 - הוסיפו **Label** עם טקסט התחלתי: "לפני פעולה".
 - הוסיפו **Button** עם טקסט: "הפעל פעולה".
2. במסך **Blocks**:
 - צרו פעולה ללא פרמטרים בשם **sayHello**.
 - בתוך הפעולה – שנו את הטקסט של ה-**Label** ל: "שלום מהפעולה!".
 - באירוע **when Button.Click** – קראו לפעולה **call sayHello**.

שאלת בדיקה:

מה יופיע ב-**Label** אחרי לחיצה על הכפתור?

תשובה: _____

תרגיל 2 – פעולה שמגרילה מספר ומשנה גודל טקסט

(דומה לתרגיל הכיתה במצגת, בגרסה פשוטה)

מצגת 6 - פעולות

שם הפרויקט: **RandomSizeProc**

1. **Designer**:
 - **Label** עם טקסט: "Procedure".
 - **Button** עם טקסט: "Click Me".
2. **Blocks**:

- צרו משתנה גלובלי `num` עם ערך התחלתי 0.
 - צרו פעולה ללא פרמטרים `setRandomSize`:
 - הפעולה מגרילה מספר שלם בין 15 ל-40.
 - שומרת את המספר ב-`num`.
 - משנה את `FontSize` של ה-`Label` לערך `num`.
 - באירוע `Button.Click` – קראו לפעולה `setRandomSize`.
-

תרגיל 3 – פעולה עם פרמטר אחד (ברכת שלום)

שם הפרויקט: `HelloNameProc`

1. `Designer`:
 - `Hint` – `TextBox`: "הכנס שם".
 - `Button` – "ברך".
 - `Label` ריק.
 2. `Blocks`:
 - צרו פעולה בשם `sayHelloTo` עם פרמטר אחד (למשל `name`).
 - בתוך הפעולה:
 - הציגו ב-`Label` את הטקסט: "שלום" + הפרמטר `name`.
 - באירוע `Button.Click`:
 - קראו לפעולה `sayHelloTo` ושלחו לה את הטקסט מתוך ה-`TextBox`.
-

תרגיל 4 – פעולה עם שני פרמטרים (מי גדול יותר)

(גרסה קלה של תרגיל "who is bigger" במצגת)

מצגת 6 - פעולות

שם הפרויקט: `WhoIsBiggerProc`

1. `Designer`:
 - שני `TextBox` (מספר ראשון, מספר שני – `NumbersOnly`).
 - `Button` – "Who is bigger".
 - `Label` ריק לתוצאה.
2. `Blocks`:
 - צרו פעולה `checkBigger` עם שני פרמטרים: `x`, `y`.
 - בתוך הפעולה:
 - אם `y > x` – כתבו ב-`Label`: "הראשון גדול יותר".
 - אחרת אם `x > y` – "השני גדול יותר".
 - אחרת – "שניהם שווים".

- באירוע `Button.Click`:
 - קראו לפעולה `checkBigger` ושלחו לה את שני המספרים מה-`TextBox`-ים.
-

תרגיל 5 – פעולה שמחזירה ערך (סכום פשוט)

שם הפרויקט: `SumReturnProc`

1. `Designer`:
 - שני `TextBox` (מספר 1, מספר 2 – `NumbersOnly`).
 - `Button` – "חשב סכום".
 - `Label` לתוצאה.
 2. `Blocks`:
 - צרו פעולה עם החזרת ערך בשם `sumTwoNumbers` עם שני פרמטרים: `a`, `b`.
 - בתוך הפעולה:
 - החזירו (`result`) את `a + b`.
 - באירוע `Button.Click`:
 - קראו לפעולה `sumTwoNumbers` עם שני המספרים מה-`TextBox`-ים.
 - שמרו את התוצאה במשתנה מקומי או ישירות ב-`Label`.
 - הציגו ב-`Label`: "הסכום הוא: " + התוצאה.
-

תרגיל 6 – שימוש ב-`Notifier` (הודעה על מספר גדול/קטן מ-50)

(בהשראת התרגיל במצגת על `Notifier`)

מצגת 6 - פעולות

שם הפרויקט: `NotifierRandom`

1. `Designer`:
 - `Button` – "הגרל מספר".
 - `Label` לתצוגת המספר.
 - רכיב בלתי נראה `Notifier` (מקטגוריית `Notifier`).
2. `Blocks`:
 - בלחיצה על הכפתור:
 - הגרילו מספר בין 1 ל-100 ושמרו אותו במשתנה.
 - הציגו את המספר ב-`Label`.
 - אם המספר גדול מ-50 – הציגו הודעת `Notifier`: "המספר גדול מ-50".
 - אחרת – הודעת `Notifier`: "המספר קטן או שווה ל-50".



תרגיל 7 – תרגיל קל בלי “גלגלי עזר”

שם הפרויקט: (תבחרו שם בעצמכם)

בתרגיל הזה אין הוראות צעד-אחר-צעד. רק דרישות כלליות:

1. השתמשו לפחות **בפעולה אחת עם פרמטר**.
2. השתמשו לפחות פעם אחת ב-**Notifier**.
3. האפליקציה שלכם יכולה להיות כל דבר שתרצו (משחק קטן, בדיקה של ציון, מחשבון, ברכות וכו').

תכננו לבד:

- אילו רכיבים יהיו ב-Designer.
- מה תהיה הפעולה, אילו פרמטרים היא מקבלת ומה היא עושה.
- מתי אתם מזמנים את הפעולה ואיפה משתמשים ב-Notifier.

בסיום, כתבו במילים קצרות מה האפליקציה שלכם עושה:
